Ilyoung Chong
Kenji Kawahara (Eds.)

Information Networking

Advances in Data Communications and Wireless Networks

LNCS 3961

International Conference, ICOIN 2006
Sendai, Japan, January 2006
Revised Selected Papers

Lecture Notes in Computer Science 3961

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Ilyoung Chong Kenji Kawahara (Eds.)

Information Networking

Advances in Data Communications and Wireless Networks

International Conference, ICOIN 2006
Sendai, Japan, January 16-19, 2006
Revised Selected Papers

Springer

Volume Editors

Ilyoung Chong
Hankuk University of Foreign Studies
Seoul, Korea
E-mail: iychong@hufs.ac.kr

Kenji Kawahara
Kyushu Institute of Technology
Kawazu 680-4, Japan
E-mail: kawahara@cse.kyutech.ac.jp

Library of Congress Control Number: 2006935653

CR Subject Classification (1998): C.2, H.4, H.3, D.2.12, D.4, H.5

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-48563-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-48563-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11919568 06/3142 5 4 3 2 1 0

Preface

This volume, LNCS 3961, contains the papers selected from those presented at the International Conference on Information Networking 2006 (ICOIN 2006), held in Sendai, Japan. ICOIN 2006 constituted the 20th Anniversary of ICOIN.

This year's conference program mainly focused on the field of ubiquitous and overlay networks, and on technology for ad hoc and sensor networks, mobile networks, transport networks, QoS and resource management, network security, peer-to-peer and overlay networks, resource management, and their applications.

In response to the call for papers, 468 papers were submitted by authors from 23 different countries from Europe, the Middle East, and the Americas. Each paper was evaluated by two or three internationally known experts to assure the excellence of the papers presented at ICOIN 2006. To keep within the conference topics, some excellent papers had to be rejected to our regret. After extensive reviews, 141 papers were chosen for presentation in 25 technical sessions. Furthermore, another review of these papers was performed during presentation, and finally 98 papers were selected for printing in LNCS 3961. We expect this will add to the excellence of ICOIN 2006.

The papers in LNCS 3961 are categorized into 8 sections: Mobile and Ubiquitous Networking, Ad Hoc and Sensor Networks, Advanced Networking, QoS and Resource Management, Network and Transport Protocols, Network Security, Applications and Services, and Peer-to-Peer and Overlay Networks, ranging from information networking to applications in next generation networks.

With our great pleasure, we would like to express our thanks to all authors including those whose papers were not included in the program. We would also like to thank all the members of the Technical Committee, who helped with their expertise, dedication and time, to put together the outstanding technical like to say a word of program. We would also thanks to the Organizing Committee Co-chairs of ICOIN 2006, Hideki Sunahara and Jin Pyo Hong for organizing the ICOIN 2006 conference successfully.

We are confident that this book will prove rewarding for all those working in the area of information networking.

September 2006

Ilyoung Chong

Kenji Kawahara

Table of Contents

Mobile and Ubiquitous Networkings

Ad Hoc Sensor Networks

Advanced Networking

QoS and Resource Management

Network and Transport Protocols

Network Security

Applications and Services

Peer-to-Peer and Overlay Networks

Mobile and Ubiquitous Networkings

νLIN6: An Efficient Network Mobility Protocol in IPv6

Ayumi Banno and Fumio Teraoka

3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
Graduate School of Science and Technology, Keio University
{banno, tera}@tera.ics.keio.ac.jp

Abstract. NEMO Basic Support Protocol has several problems such as pinball routing, large header overhead due to multiple passes of tunneling, and a single point of failure. This paper proposes a protocol called νLIN6 which supports both network mobility and host mobility in IPv6. In νLIN6, packet relay is required only once regardless of the nested level in network mobility, while optimal routing is always provided in host mobility. A fixed-sized extension header is used in network mobility while there is no header overhead in host mobility. νLIN6 is more tolerant of network failure and mobility agent failure than NEMO Basic Support Protocol. It also allows ordinary IPv6 nodes to communicate with nodes in the mobile network.

1 Introduction

NEMO Basic Support Protocol (NBSP) [1] provides a communication mechanism transparent to movement of networks. Since it is based on Mobile IPv6 (MIP6) [2], it has several problems such as pinball routing, large header overhead due to multiple layers of tunneling, and a single point of failure. Although there are several proposals [3,4,5,6] to solve these problems, there is no fundamental solution because they are based on MIP6.

νLIN6 is based on a node mobility protocol called *LIN6* [7]. The fundamental concept of LIN6 is separation of the node identifier and the node locator. There are several proposals based on similar concepts such as HIP [8] and SHIM6 [9]. However, none of them support network mobility.

In host mobility, LIN6 provides optimal routing without header overhead. Since the Mapping Agents, the mobility agents in LIN6, can be connected to arbitrary locations in the Internet, LIN6 is tolerant to network failure and crash of any mobility agent. LIN6-NEMO [10] also provided optimal routing without header overhead, via a network mobility protocol based on LIN6. However, it generated a lot of signaling packets when a nested mobile network changed its point of attachment to the Internet.

This paper proposes νLIN6, an enhanced version of LIN6-NEMO. It supports host mobility in optimal routing without header overhead. In network mobility, it requires packet relay only once by the Mapping Agent and uses a fixed-sized extension header regardless of the nested level of the mobile network.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 3–10, 2006.

2 Separation of Node Identifier and Its Locator

νLIN6 is based on the concept of separation of the node identifier and its locator. νLIN6 uses the *LIN6 address* in the network layer. The LIN6 address is an IPv6 address which consists of the current network prefix and the node identifier called the *LIN6 ID*. They are 64 bits in length. On the other hand, the transport and upper layers use the *LIN6 generalized ID*, which consists of a constant called the *LIN6 prefix* and the LIN6 ID. Since the LIN6 generalized ID remains unchanged regardless of the node location, node movement is transparent to the transport and upper layers.

The relation between the LIN6 ID and the current locator is called the *mapping*. The mapping is maintained by the *Mapping Agent*. In the sending host, the LIN6 generalized ID of the destination host is passed from the transport layer to the network layer, and then it is converted to the LIN6 address by overwriting the LIN6 prefix with the current locator. The current locator of the destination host can be obtained by accessing the Mapping Agent. In the receiving host, the source and destination LIN6 addresses are converted to LIN6 generalized IDs by overwriting the locator part with the LIN6 prefix, and then they are passed to the transport layer.

As described above, the network prefix part of the LIN6 address can arbitrarily be modified in the network layer because it is overwritten with the LIN6 prefix before it is passed to the transport layer. νLIN6 makes use of this feature to achieve low routing and header overheads.

3 Components of νLIN6

νLIN6 consists of the following components: the mobile router (MR), the visiting mobile node (VMN), the correspondent node (CN), the access router (AR), and the Mapping Agent (MA). A VMN and a MR have their own *home network*. The home network of a VMN or a MR is the network to which the VMN or the MR is usually connected, i.e., the network prefix of the VMN or the MR is aggregatable to the network prefix of the home network. The MA connected to the home network of a MR is called the *Primary Mapping Agent (P-MA)*. The MA connected to the subnets other than the home network is called the *Secondary Mapping Agent (S-MA)*.

In νLIN6, the following records are registered with the DNS.

- AAAA record: $\mathrm{FQDN}_{VMN,MR} \Rightarrow$ network prefix of $\mathrm{P\text{-}MA}_{VMN,MR}$+$\mathrm{ID}_{VMN,MR}$
- PTR record: $\mathrm{ID}_{VMN,MR} \Rightarrow \mathrm{FQDN}_{VMN,MR}$
- TXT record: $\mathrm{ID}_{VMN,MR} \Rightarrow$ IPv6 addresses of $\mathrm{MAs}_{VMN,MR}$

$\mathrm{FQDN}_{VMN,MR}$ is the fully qualified domain name (FQDN) of a VMN or a MR. $\mathrm{ID}_{VMN,MR}$ is the LIN6 ID of a VMN or a MR. $\mathrm{MAs}_{VMN,MR}$ are the MAs of a VMN or a MR. Since a VMN or a MR can have more than one MA (one P-MA and several S-MAs), the TXT record of $\mathrm{ID}_{VMN,MR}$ may have several IPv6 addresses. By accessing the DNS, the CN which implements νLIN6 can obtain

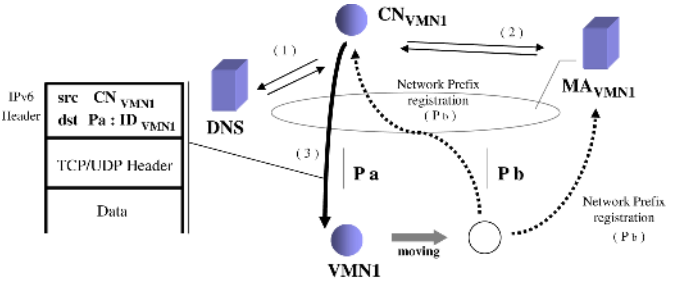


Fig. 1. Communication procedure between LIN6 nodes in νLIN6

the network prefixes of the MAs of the target VMN. If the CN is an ordinary IPv6 node, the packet sent to a VMN reaches the P-MA of the VMN, and the P-MA relays the packet to the VMN by tunneling.

4 Host Mobility Support

νLIN6 supports host mobility in optimal routing without header overhead. In Fig. 1, the visiting mobile node *VMN1* registered its current network prefix with its Mapping Agent MA_{VMN1}. MA_{VMN1} holds the mapping $ID_{VMN1} \Rightarrow P_a$. ID_{VMN1} is the LIN6 ID of VMN1 and P_a is the current network prefix of VMN1.

The communication procedures from CN to VMN1 are as follows. CN inquires of the DNS the IP address of MA_{VMN1} that manages the mapping of VMN1 when CN begins communication to VMN1, and then CN inquires of MA_{VMN1} the mapping of VMN1 (Fig. 1-(1)(2)). CN acquires the network prefix (P_a) as the current network prefix of VMN1. CN generates the LIN6 address from the acquired network prefix (P_a) and the LIN6 ID (ID_{VMN1}) of VMN1, makes it the destination address, and transmits the packet. The packet reaches VMN1 (Fig. 1-(3)). If the CN is an ordinary IPv6 node, the packet sent by CN reaches MA_{VMN1}, and then MA_{VMN1} relays the packet to VMN1 by tunneling.

When VMN1 moves, it registers the current network prefix (P_b) with MA_{VMN1} and CN if an IPsec SA is established between VMN1 and CN. If an IPsec SA is not established, VMN1 notifies CN of its movement, and then CN inquires of MA_{VMN1} the new mapping of VMN1. In Fig. 1, VMN1 registers the network prefix P_b with MA_{VMN1} and CN_{VMN1} because it is assumed that an IPsec SA is established between VMN1 and CN_{VMN1}.

5 Network Mobility Support

5.1 Mapping Registration Procedure

The mapping registration procedure is depicted in Fig. 2. Table 1 shows the mapping table after the registration. If a MR is away from the home network,

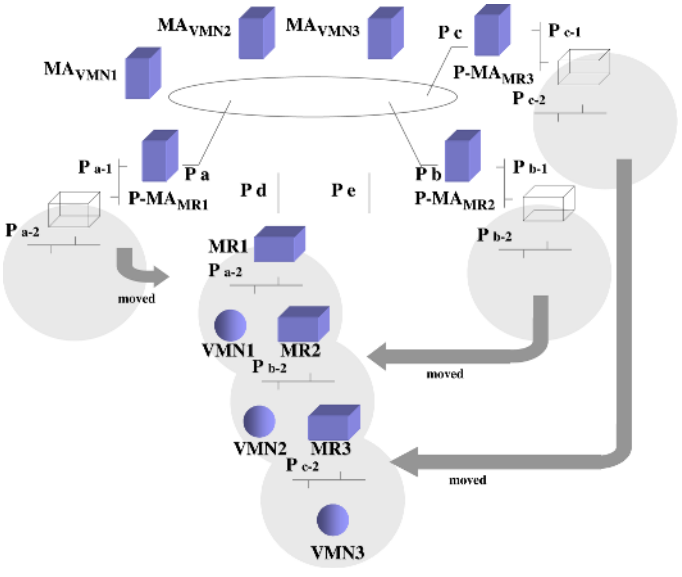


Fig. 2. Mapping registration with the MA in νLIN6

the on-flink flag is set in its mapping table entry. MA_{VMN1}–MA_{VMN3} and P-MA_{MR1}–P-MA_{MR3} of Fig. 2 are the MAs of VMN1–VMN3 and MR1–MR3, respectively. P_a–P_e are the network prefixes allocated to each link. P-MA_{MR1} and P-MA_{MR3} are the P-MAs of MR1 and MR3, respectively. In Fig. 2, MR1 moved from P_{a-1} to P_d, MR2 moved from P_{b-1} to P_{a-2} and MR3 moved from P_{c-1} to P_{b-2}, and then they form a nested mobile network (*NEMO*). P_{a-1} is the home network of MR1. P_{a-2} is called the *Mobile Network Prefix (MNP)* of MR1. The MNP is the IPv6 prefix delegated to the MR. It is advertised in the NEMO. Similarly, P_{b-1} is the home network of MR2 and P_{b-2} is the MNP of MR2. P_{c-1} is the home network of MR3 and P_{c-2} is the MNP of MR3.

In case of a nested NEMO, the outermost MR is called the *root-MR* and the network prefix of the root-MR is called the *root-MRLoc*. If a MR is the root-MR of a nested NEMO, the MR registers its network prefix as the root-MRLoc with its MAs. In Fig. 2, since MR1 is the root-MR, it registers P_d (root-MRLoc) with P-MA_{MR1}. A MR inside a nested NEMO is called a *sub-MR*. The sub-MR registers the root-MRLoc of the nested NEMO to which it belongs with its P-MA. In Fig. 2, when MR2 moves to P_{a-2}, it receives the modified Router Advertisement message from MR1 which includes MR1's LIN6 ID, and then detects that it resides inside a NEMO. MR2 registers the mapping of ID_{MR2} and P_{a-2} with MR1, and then receives the acknowledgment from MR1 which includes the root-MRLoc. Since MR2 is a sub-MR in the nested NEMO, it registers P_d (root-MRLoc) with P-MA_{MR2}. In Fig. 2, when MR3 moves to P_{b-2}, it receives the modified Router Advertisement message from MR2 which includes MR2's LIN6 ID, and then detects that it resides inside a NEMO. MR2

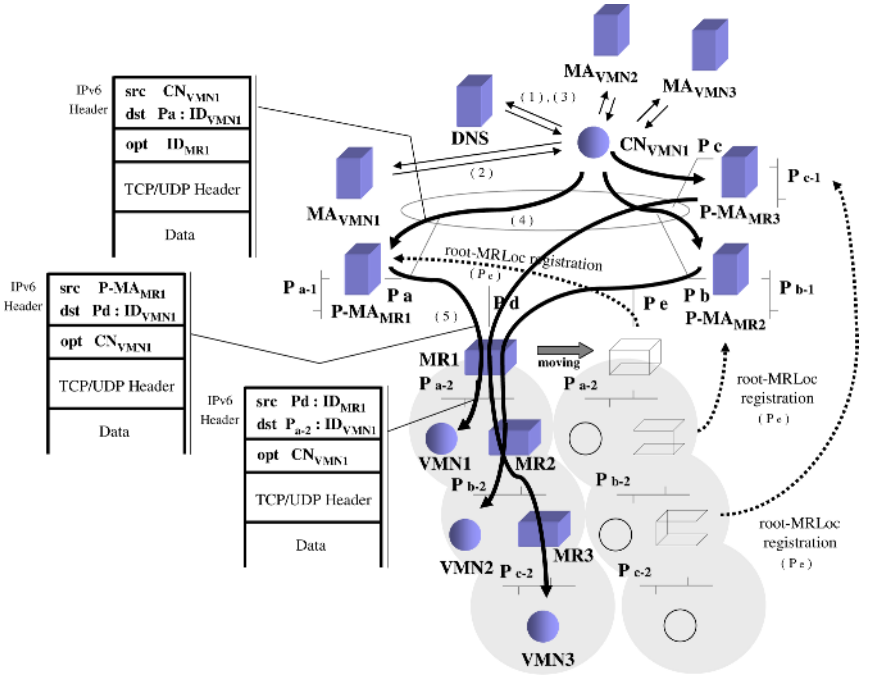


Fig. 3. Communication procedure between LIN6 nodes in νLIN6

registers the mapping of ID_{MR3} and P_{b-2} with MR2, and then receives the acknowledgment from MR2 which includes the root-MRLoc. MR2 relays that registration message to MR1. Since MR3 is a sub-MR in the nested NEMO, it registers P_d (root-MRLoc) with P-MA_{MR3}.

If the VMN is connected to a NEMO, it registers the LIN6 ID of the MR to which it is connected. The MR advertises its LIN6 ID to VMNs via the modified Router Advertisement message. In Fig. 2, since VMN1 is connected to a NEMO link provided by MR1, it registers the LIN6 ID of MR1 (ID_{MR1}) with MA_{VMN1}. Similarly, since VMN2 is connected to the nested NEMO provided by MR2, it registers the LIN6 ID of MR2 (ID_{MR2}) with MA_{VMN2}, and since VMN3 is connected to the nested NEMO provided by MR3, it registers the LIN6 ID of MR3 (ID_{MR3}) with MA_{VMN3}.

5.2 Communication Procedures

Figure 3 depicts the communication procedure. Table 1 shows the related mapping table entries created by the registration procedures of the MRs and the VMNs.

The communication procedure from CN to VMN1 is as follows. CN queries the DNS for the AAAA record of VMN1 when CN begins communication with

Table 1. Mapping Table in each Mapping Agent in Fig. 2. and 3. after registration

	LIN6 ID	network prefix or LIN6 ID	on-flink flag
MA_{VMN1}	ID_{VMN1}	ID_{MR1}	-
MA_{VMN2}	ID_{VMN2}	ID_{MR2}	-
MA_{VMN3}	ID_{VMN3}	ID_{MR3}	-
MA_{MR1}	ID_{MR1}	P_d(root-MRLoc)	1
MA_{MR2}	ID_{MR2}	P_d(root-MRLoc)	1
MA_{MR3}	ID_{MR3}	P_d(root-MRLoc)	1

VMN1, and then CN acquires the network prefix of MA_{VMN1} and LIN6 ID of VMN1 (ID_{VMN1}) (Fig. 3-(1)). CN queries MA_{VMN1} for the mapping of VMN1 and CN acquires LIN6 ID of MR1 (ID_{MR1}) (Fig. 3-(2)). The ID_{MR1} acquired at this time is the LIN6 ID of the MR to which VMN1 is connected. CN inquires of the DNS the TXT record of MR1 and CN acquires the IP address of MA_{MR1} (Fig. 3-(3)). CN generates the LIN6 address from the acquired network prefix of MA_{MR1} (P_a) and the LIN6 ID of VMN1 (ID_{VMN1}), and then makes it the destination address. It also sets the LIN6 ID of the MR (ID_{MR1}) in the *νLIN6 option* and transmits the packet. The νLIN6 option is included in the Destination Options Header of IPv6. The size of the Destination Options Header containing the νLIN6 option is 24 bytes. The packet reaches P-MA_{MR1} (Fig. 3-(4)). P-MA_{MR1} rewrites the network prefix of the destination address of the received packet with the root-MRLoc (P_d). It rewrites the source address of the received packet with its address to avoid ingress filtering, also rewrites the νLIN6 option with the address of CN, and then relays the packet. The packet reaches MR1. MR1 also rewrites the network prefix of the destination address of the received packet with the network prefix P_{a-2}. It rewrites the source address with its address, and then forwards the packet (Fig. 3-(5)). Upon receiving the packet, VMN1 rewrites the source address with the address contained in the νLIN6 option (CN).

In the communication from CN to VMN2, a similar procedure is executed. Because VMN2 registers the LIN6 ID of MR2 (ID_{MR2}) with MA_{VMN2}, P-MA_{MR2} relays the packet to MR1. In the communication from CN to VMN3, a similar procedure is executed. Because VMN3 registers the LIN6 ID of MR3 (ID_{MR3}) with MA_{VMN3}, P-MA_{MR3} relays the packet to MR1. In the communication from VMN1, VMN2, or MR3 to CN, the source address is the current locator and the LIN6 ID of VMN1, VMN2 or VMN3. Upon forwarding this packet, MR1 overwrites the network prefix of the source address with the root-MRLoc (P_d) to avoid ingress filtering. This packet reaches CN on the optimal route. Thus, the communication paths between LIN6 nodes become asymmetric in νLIN6.

5.3 Handover Procedures

In Fig. 3, when the nested NEMO moves to a new link (P_e), the root-MRLoc of MR1 changes from P_d to P_e. MR1 registers the new root-MRLoc (P_e) with

P-MA$_{MR1}$. It also announces the new root-MRLoc in the NEMO via a *root-MRLoc Update* message. Upon receiving the root-MRLoc Update message, MR2 registers the new root-MRLoc with P-MA$_{MR2}$ and sends the root-MRLoc Update message in the NEMO. Similarly, upon receiving the root-MRLoc Update message, MR3 registers the new root-MRLoc with P-MA$_{MR3}$. Thus, when a nested NEMO moves, the number of signaling messages sent to the Internet is equal to the number of MRs in the NEMO regardless of the number of nodes in the NEMO.

6 Considerations

This section compares νLIN6 with NBSP and ONEMO [5] from the viewpoint of qualitative performance. Table 2 shows the results of the comparison.

Packet Routing: In NBSP, pinball routing occurs in packet delivery in which the packet is relayed by two or more Home Agents (HAs). The more the nested level of the NEMO increases, the more serious pinball routing becomes. In ONEMO, the Correspondent Router (CR) always relays the packet to the node in the NEMO. In νLIN6, the MA of the root-MR always relays the packet to the node in the NEMO. νLIN6 and ONEMO need to relay the packet only once regardless of the nested level of the NEMO.

Header Overhead: In NBSP, a 40 bytes IPv6 header is added to the packet by tunneling. If the nested level is n, the header overhead becomes $40n$ bytes. In ONEMO, tunneling is used once regardless of the nested level of the NEMO and a 40 bytes IPv6 header is added to the packet. Tunneling might cause packet fragmentation on the communication path from the CN to the NEMO. As a result, TCP performance might go down. In νLIN6, tunneling is not used; the MA rewrites the network prefix of the destination address of the relayed packet in communication between LIN6 nodes. νLIN6 has no header overhead in node mobility, while it uses the extension header containing the νLIN6 option in network mobility. The extension header is 24 bytes in size and it is attached to the packet at the source node.

Fault Tolerance: In NBSP, all packets are relayed by the HA. Since the HA must be connected to the home network of the VMN or the MR, if the HA crashes or failure occurs in the home network, communication to/from the NEMO becomes unavailable. In ONEMO, all packets are relayed by the CR. Even if the CR crashes, communication to the NEMO is still available by using another CR. But all packets concentrate at the CR. In νLIN6, an arbitrary number of MAs can be connected to arbitrary locations in the Internet. Even if a MA crashes or failure occurs in the link to which the MA is connected, communication to the NEMO is still available by using another MA. In addition, νLIN6 can decentralize the traffic to the MAs by distributing the S-MAs in the Internet.

Table 2. Comparison between NBSP, ONEMO and νLIN6

	NBSP	ONEMO	νLIN6
Routing to NEMO	pinball routing	relay only once	relay only once
Routing to VMN	relay by HA or optimal	relay by HA or optimal	always optimal
Signaling messages	1	No. of MRs in NEMO + 6	No. of MRs in NEMO
Header overhead	40byte×No. of HAs passed	40bytes	24bytes
Tunneling	nested level	once	none
Fault Tolerance	HA is single point of failure	CR is bottleneck	MAs can be distributed

7 Conclusion

This paper has proposed a new network mobility protocol νLIN6. νLIN6 is superior to NBSP in terms of the packet routing, header overhead, and fault tolerance. We plan to implement νLIN6 on NetBSD and to test it in a testbed.

References

1. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network Mobility (NEMO) Basic Support Protocol. (2005) RFC 3963.
2. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. (2004) RFC 3775.
3. Takagi, Y., Ohnishi, H., Sakitani, K., Baba, K., Shimojo, S.: Route Optimization Methods for Network Mobility with Mobile IPv6. IEICE Transactions on Communication **E87-B**(3) (2004) 480–489
4. Thubert, P., Molteni, M.: IPv6 Reverse Routing Header and Its Application to Mobile Networks. (2004) Internet Draft, work in progress.
5. Watari, M., Wakikawa, R., Thierry, E., Murai, J.: Optimal Path Establishment for Nested Mobile Networks. In: Proceedings of VTC2005-Fall. (2005)
6. Ng, C., Zhao, F., Watari, M., Thubert, P.: Network Mobility Route Optimization Solution Space Analysis. (2006) Internet Draft, work in progress.
7. Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H., Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks. IEICE Transactions on Communication **E84-B**(8) (2001)
8. Moskowitz, R., Nikander, P., Nikander, P.: Host Identity Protocol. (2006) Internet Draft, work in progress.
9. Nordmark, E., Bagnulo, M.: Level 3 multihoming shim protocol. (2006) Internet Draft, work in progress.
10. Oiwa, T., Kunishi, M., Ishiyama, M., Kohno, M., Teraoka, F.: A Network Mobility Protocol Based on LIN6. In: Proceedings of VTC2003-Fall. (2003)

Applying NEMO to a Mountain Rescue Domain

Ben McCarthy, Christopher Edwards, and Martin Dunmore

Computing Department, Infolab 21, Lancaster University,
Lancaster, LA1 4WA, UK
{b.mccarthy, ce, m.dunmore}@comp.lancs.ac.uk

Abstract. In an effort to provide a solution to the problem of internetworking mountain rescue workers without the use of a fixed infrastructure, our group has explored the use of a network model based on the concept of Network Mobility and the use of the NEMO Basic Support Protocol. In this paper we consider the feasibility of this approach through the design, configuration and testing of a working example of this scenario using an implementation of the NEMO Basic Support Protocol made available to us by Cisco Systems (running on their 3200 Series Mobile Access Router platform). We provide the results from performance testing carried out over our testbed that highlights the real impact of the scalability problems of the NEMO Basic Support Protocol in a scenario which results in a nested NEMO topology. In addition we present results that illustrate the performance improvements achievable in our scenario through using the NEMO Route Optimisation (RO) solution, Reverse Routing Header (RRH).

1 Introduction

The purpose of our work is to explore the feasibility of providing data networking between mountain rescue workers without the requirement of any fixed hardware infrastructure. With the recent increases in cheap, high throughput wireless technologies (802.11 a/b/g [1]) and the purported range capabilities of the next generation of standards (i.e. 802.16 d/e [2][3]), the ability to provide such a solution at the hardware level is becoming increasingly feasible. However it is when we consider the routing architecture of such a model that the biggest elements of research arise.

The network model used throughout this paper is based on the typical structure and movement of a mountain rescue team (and was produced through consultation with the Cockermouth rescue team). The team as a whole splits up into numerous smaller search parties when performing a search and rescue mission; search parties move independently of each other, but members within each search party move together and remain clustered together throughout the mission. Using this model we explore the feasibility of using a NEMO-based mobile networking solution to support this real life mobility scenario. In our proposed solution, each search party forms an individual mobile network. To facilitate this, one team member from each search party is expected to carry a mobile router which provides connectivity to all the other wireless devices carried by members of their individual search party and at the same time attempts to maintain connectivity (via a long range wireless link) with the other search parties (or with the rescue team's All Terrain Vehicles positioned at the bottom of the search area).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 11–20, 2006.

From a broad perspective, the NEMO Basic Support Protocol (NEMO BS) [4] can support all of the functionality that the mountain rescue team would require, but the technique it employs introduces many overheads to scenarios with complex mobility models and could prove to be too inefficient to provide even a working solution. To contrast this we explore the efficiency improvements facilitated by using the Reverse Routing Header (RRH) technique [5] and analyse how well we believe these protocols could cope with the demands of a working solution. In order to carry out some lab based performance testing we created a testbed consisting of five Cisco 3200 Mobile Access Routers [6], running a pre-release IOS version which supports both the NEMO Basic Support Protocol and the RRH extension.

2 Mobility Model Used to Perform Testing

Here we define the mobility model that the NEMO Basic Support Protocol and the RRH extension were tested against in order to highlight their performance as the network topology altered. Supporting the Mountain Rescue Network using NEMO introduces an interesting problem when we begin to consider the location of each of the mobile network's Home Agents (HAs). For the purpose of this paper we chose to locate each of the "Base" HAs on the HQ network and each of the "Team" HAs on their respective "Base" Mobile Networks. We chose this configuration because in the mountain rescue scenario, search parties (and hence the networks they form) are more mobile than the vehicles that they originate their search and rescue missions from. By locating the HAs of the search party networks at their respective vehicle networks, we expect the network would better cope with the localised mobility of the search party teams and thus be better suited to support the network in its most typical state.

In all of our tests the Mountain Rescue Network begins in the state shown in figure 1, prior to the movement shown as stages 1 & 2. This is what we term as the network's "Home Configuration" due to the location of the Mobile Network HAs (i.e. Red and Green Base's HAs are located on the HQ MR and the Red and Green Team HAs are located on their respective Base MRs.) From this "Home Configuration" starting point, we then proceed to implement the mobility model as shown in the following series of stages:

Stage 1 and 2: Team Mobility. Team mobility refers to the roaming of a team from one "Base" Mobile Network to another. In physical terms this represents the movement of a team during a search, conceivably moving from one all terrain vehicle's 802.16 hotspot to another vehicle's 802.16 hotspot. This scenario is highly plausible and would be most likely to occur when a victim has been found and all of the seperate teams converge at the point the victim has been located. In Figure 1 this type of mobility occurs in Stage 1 when the Green Team move away from the Green Base and connects to the Red Base Mobile Network. Then in Stage 2 we consider when the Green Team continues to move and remains connected via the Red Team's network. The path that a flow of packets between the Green Team and the Red Team will take while the network is at Stage 2 illustrates well how suboptimal a route can become when using the NEMO Basic Support Protocol. Packets generated by a node on the Green Team network that

are destined for a node on the Red Team network will flow through the Red Team network toward the Green Team's HA via a bi-directional tunnel before being delivered back up to the Red Team network once the packets are decapsulated by the HA.

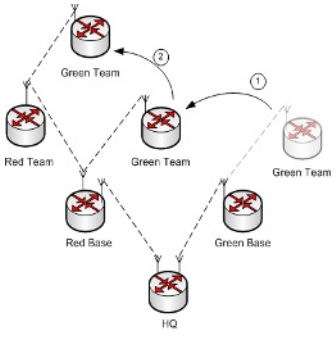


Fig. 1. Team Mobility

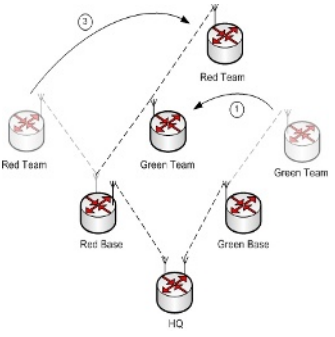


Fig. 2. Nested Team Mobility

Stage 3: Nested Team Mobility. Figure 2 illustrates the scenario where after the roaming of the Green Team highlighted in Stage 1, the Red Team roams and remains connected via the Green Team network. This sequence of mobility will produce a nested NEMO structure, this is because the Red Team is roaming via the Green Team's network which is also away from home. This means that (for NEMO BS) when the Red Team sets up its bi-directional tunnel with its HA it will now go through the Green Teams bi-directional tunnel and hence all traffic flowing to and from the Red Team network will travel via the respective HAs on both the Red Base network and the Green Base network.

Stage 4: Home Agent Mobility. The least complex scenario that involves HA mobility can be seen to occur when one of the "Base" Mobile Networks moves together with all of its teams. This scenario is illustrated in Figure 3 and would conceivably occur when one of the teams (in this case the Green team) are instructed to search an entirely different geographical area, which they must travel to in their all terrain vehicle and is located in an area where they can only connect back to the HQ via one of the other Mobile Networks. In this scenario, the route suboptimality exists when packets are directed toward a Mobile Network's Home Network as the Home Network itself is now roaming and packets must be tunnelled to its new location.

Stage 5: Home Agent Mobility with Continued Team Mobility. The next stage of the mobility model continues on from the scenario outlined in figure 3. The "Base" Mobile Networks are comparatively less mobile than the "Team" Mobile Networks and therefore once in the topology illustrated, we could expect the "Team" Mobile Networks to begin to roam and form topologies that would produce route suboptimality to the HA and then further route suboptimality to the Mobile Network that packets are destined for. In Stage 5, the Green and the Red Team switch positions topologically because of their movements, this means that both teams are away from home as shown in Figure 4. Testing the network in this configuration with NEMO BS will impose a high degree of

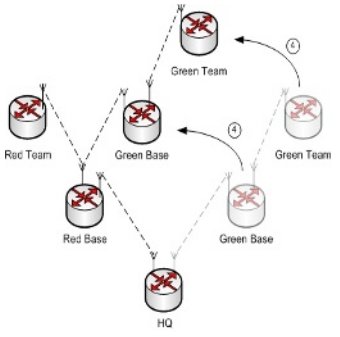


Fig. 3. HA Mobility

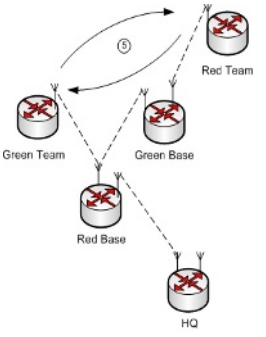


Fig. 4. HA Mobility with Team Mobility

routing suboptimality as packets travelling between the two nodes will go via the HAs of both the Green and Red Team and the HA of the Green Base.

Stage 6: Inverted Home Agent with Team Mobility. This performance test (illustrated in Figure 5) represents the scenario whereby a mountain rescue team's HA (the "Base" Mobile Network) must connect to the Mountain Rescue Network via one of its "Team" Mobile Networks, thus placing the HA further up the network topology than one of its Mobile Networks registered with it. Whilst this type of topology may not occur frequently, it is entirely possible that it could, for instance if two search teams began their missions at different altitudes on a particular mountain face where the "Base" Mobile Network could not negotiate a connection back to the HQ network.

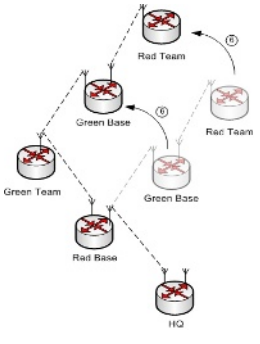


Fig. 5. Inverted HA with Team Mobility

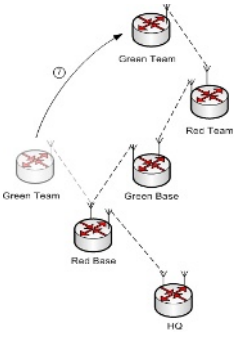


Fig. 6. Worst Case Scenario

Stage 7: Worst Case Scenario. Mobility stage 7 represents an example of one of the worst case scenarios (that doesn't incorporate an inverted HA as in the previous mobility stage) that can be replicated using our testbed. This scenario occurs when a single connectivity chain exists whereby each Mobile Network has only one Ingress and one Egress connection each. In addition, both rescue team Mobile Networks are configured to be roaming and there is also HA mobility.

3 Mountain Rescue Network Testbed Setup

All performance tests presented in this paper have been carried out using the Mountain Rescue Network Testbed within the Lancaster University Computing Department [7][8]. This testbed comprises of 5 Cisco 3200 Mobile Access Routers (MARs), each running a pre-release IOS version that supports both the NEMO Basic Support Protocol and proprietary NEMO extensions such as RRH. We configured and tested this testbed using both Wired and Wireless links, in this paper we present the findings using wired links as they more accurately display the performance implications imposed by the mobility protocols because they reduce the number of unknown factors (i.e. interference) that could affect our testing results. All the tests performed over the Mountain Rescue Network were between two 1Ghz, 256Mb RAM PCs (one connected to the Red Team Mobile Network and one connected to the Green Team Mobile Network). The two PCs were the only nodes connected to the testbed, which itself is a stand alone network with no external connectivity.

3.1 Specification of Performance Tests

For each different configuration of the testbed generated by the different stages of the mobility model (specified in section 2), we measured TCP and UDP throughput and Latency.

TCP Testing: Using IPerf, we calculated the bandwidth available by generating a TCP flow of IPv6 packets with a 1000 byte payload and a TCP window size of 200000 bytes. We created these flows for ten minute durations and ran the test three times separately to ensure that the test results were not affected by some unknown sporadic event.

UDP Testing: Again using IPerf, we generated UDP flows of IPv6 packets with a 1000 byte payload at different bandwidth rates. Using a Binary search method, we generated short lived (30 second) flows at different bandwidths until we identified the maximum UDP flow bandwidth supportable without experiencing any loss at each stage in the mobility model.

Latency Testing: Finally, the latency tests were performed by running a Ping utility between the two end nodes for ten minutes and recording the average Round Trip Time experienced.

For the TCP bandwidth testing, the window size used was chosen to produce the best possible throughput over the wired testbed and for both the UDP and TCP testing, the payload size was chosen to be 1000 bytes to prevent the additional headers created by the MR-HA bi-directional tunnels from causing packet fragmentation (analysis of the further inefficiencies imposed on our network by fragmentation was not considered to be within the scope of this paper).

4 Performance Testing Results

The results discussed throughout this section are summarised in Figure 7.

4.1 Home Configuration Performance Results

Before trying to determine the performance degradation that mobility imposes on our testbed, it is important to understand the network's capabilities in its Home Configuration state. In the Home Configuration, each of the MRs are connected to their respective HAs and therefore are not roaming. Throughput and latency in this configuration should therefore be the maximal values attained over this testbed. These results were attained with the testbed configured in both NEMO BS mode and with RRH enabled, in both modes the testbed provided exactly the same performance.

4.2 NEMO Basic Support Testing

In this section we present the results achieved when testing the NEMO BS implementation against the mobility model we devised. For each stage in the mobility model we also highlight the Hop count of the end-to-end path to illustrate the suboptimality of the route imposed in that scenario:

[Mobility Stage 1 (Hop Count = 7)]. This mobility stage (illustrated in Figure 1) highlights the performance degradation experienced when a single mobile network in the path is roaming and therefore 1 layer of tunelling and 1 level of indirection is experienced. This simple network configuration difference imposes an increase in latency of over 30 % and an overall reduction in throughput of nearly 40 %.

[Mobility Stage 2 (Hop Count = 8)]. Mobility stage 2 (also illustrated in Figure 1) is very similar to mobility stage 1 as it consists again of a single mobile network in the path that is roaming. What it highlights is the performance degradation experienced along a path in the Mountain Rescue Network Testbed when a single additional hop is introduced. The results show that each additional hop introduces a further 1 millisecond delay to the latency experienced and a small reduction in UDP throughput. The biggest impact was on the TCP flow which experienced over a 1Mbps reduction in overall throughput.

[Mobility Stage 3 (Hop Count = 12)]. Mobility stage 3 introduces a simple nested NEMO configuration that causes packets being sent along the path between the two end nodes to travel via two HAs and to be encapsulated in two different MR-HA tunnels. This mobility scenario provides a perfect example of the massive inefficiencies that can be experienced when utilising NEMO BS. Whilst the Green and Red Team networks are physically only 1 hop away from one another, because of the tunnelling required by NEMO BS, the end-to-end path ends up being 12 hops long! This fact is clearly illustrated in the results, with both throughput and latency degrading by 50 % more than in stage 2.

[Mobility Stage 4 (Hop Count = 6)]. This mobility stage introduces a mobile HA to the network configuration. Theoretically this configuration should result in an end-to-end performance that is better than that achieved over mobility stage 1, because there is one fewer Hop to be traversed on the path. The results reflect this, with a throughput improvement of around 10% on that achievable in stage 1. This shows that a mobile HA performs in the expected manner and does not introduce any unforeseen overhead.

[Mobility Stage 5 (Hop Count = 12)]. Mobility Stage 5 continues with the theme of a mobile HA, but complicates the network configuration further by also roaming both of the mobile networks that the communicating nodes are connected to (Red Team and Green Team). In this configuration, packets traversing the end-to-end path traverse 12 Hops and are encapsulated and decapsulated 3 times (with a 2 layer tunnel experienced between the HQ and the roaming Green Base). If we compare the results for this stage with those of mobility stage 3 (which also has 12 hops but encapsulation is only performed twice) we can see that the necessity to perform encapsulation and decapsulation a third time introduces additional adverse affects to the overall throughput achievable and latency times recorded.

[Mobility Stage 6 (Hop Count = N/A)]. NEMO Basic Support does not support this configuration because of the inverted HA (the HA is higher in the topology than the Mobile Network that is registered to it.) NEMO BS's inability to support the scenario is a known problem and is documented in section 2.7 (under the heading "Stalemate with a Home Agent Nested in a Mobile Network") in the NEMO Working Group (WG) RO Problem statement draft [9]. The problem exists because the Mobile Network cannot find its HA in the Internet and thus cannot establish its MR-HA tunnel.

[Mobility Stage 7 (Hop Count = 17)]. This mobility model represents one of the worst case scenarios configurable using our testbed. It clearly illustrates how complex and inefficient the end-to-end path can become when using NEMO BS to support our mountain rescue scenario. Although the same number of MRs are used as in the Home Configuration, the increased route complexity generated by the mobility model imposed has reduced the overall throughput by almost 85 % and created a four fold increase in the latency experienced.

4.3 Reverse Routing Header Testing

After performing the NEMO BS performance tests we then reconfigured each of the MRs in the testbed to enable RRH support and carried out the same tests again.

[Mobility Stages 1, 2 and 4]. Mobility stages 1, 2 and 4 only impose a network configuration that generates a single MR-HA tunnel. This means that the RRH technique should not introduce any improvements to these configurations as it is only intended to optimise delivery to nested NEMO. The results presented in Figure 7 show that whilst the UDP throughput and latency times aren't affected by any additional overhead that the RRH protocol may impose, TCP flows consistently experienced an unexplained 3% decrease in throughput.

[Mobility Stage 3 (Hop Count = 9)]. As Mobility stage 3 introduces a simple nested NEMO scenario, the performance results achieved across our testbed using RRH should theoretically improve on those achieved when using NEMO BS. The results in Figure 7 are evidence of these performance improvements, with total throughput experiencing an increase of approximately 28% (26% for UDP & 30% for TCP). This is a strong example of the efficiency gains that can be made in even the most simple nested NEMO scenarios, by removing the overhead introduced by Pinball routing.

[Mobility Stage 5 (Hop Count = 10)]. Mobility stage 5 again contains an example of a nested NEMO structure but also introduces HA mobility to the configuration as

well. The test results for this configuration show a much smaller improvement over NEMO BS than those experienced with mobility stage 3. In this scenario, RRH can only make minor optimisations to the route. This occurs because the path between the HQ and the roaming Green Base cannot be optimised (it is already the direct route) and also the Green Base cannot optimise its route to the Green Team network because all of its traffic must first travel via the HQ (because it is roaming). The only optimisation that is performed successfully is between the Red Team and its HA on the Red Base network.

[Mobility Stage 6 (Hop Count = 15)]. Mobility stage 6 was not supported by NEMO BS because the Green Team HA is located higher in the mobile network topology than the Green Team itself. However, with RRH this configuration is supported and therefore it isn't possible to draw any other comparison with NEMO BS other than an atomic one (i.e. RRH supports this scenario, NEMO BS does not). RRH is able to support this scenario because in trying to remove the nested tunnelling, it reveals the path between the Green Base and the HQ to the Green Base HA.

[Mobility Stage 7 (Hop Count = 12)]. In Mobility stage 7, again only the Red Team HA (based on the Red Base network) is able to optimise the route to its mobile network effectively, but in this scenario the optimisation reduces the number of Hops along the end-to-end path by 5. This significant optimisation of the end-to-end path becomes evident when we analyse the performance results for this mobility stage. With a decrease in latency times of 40 % and an increase in throughput of over 70 %, this scenario dramatically illustrates the benefits that the RRH technique can provide when our mobility model grows increasingly complex.

Mob Stage:	BS UDP	BS TCP	BS Latency	RRH UDP	RRH TCP	RRH Latency
Home Conf:	23.3	17.4	3.97	23.3	17.4	3.97
Stage 1:	13	10.6	5.86	13.4	10.2	5.82
Stage 2:	12.8	9.44	6.42	12.8	9.24	6.57
Stage 3:	7.3	4.58	9.83	9.8	7.59	6.93
Stage 4:	13.3	11.7	4.94	13.6	11.3	5.02
Stage 5:	4.6	3.12	10.44	5.5	4.65	9.20
Stage 6:	N/A	N/A	N/A	5.26	4.19	11.48
Stage 7:	4.2	2.63	12.21	7.6	4.51	7.40

Fig. 7. NEMO Basic Support & RRH Testing Results

5 Analysis and Future Direction

In developing the Mountain Rescue Testbed and carrying out the performance tests described, we feel we have developed a thorough understanding of the implications of applying the NEMO Basic Support protocol to our mountain rescue scenario. The use of mobile networks in this scenario can introduce some extremely desirable functionality however the purpose of this paper was to determine the cost implications of attaining that functionality. Figures 8 & 9 illustrate this cost by comparing our test results for each mobility stage against the performance of the network in its Home Configuration. In a

non optimised state, we found that the NEMO Basic Support Protocol unacceptably consumed network resources as the mobile network configurations experienced in the mountain rescue scenario became more and more complex. What is also evident from our test results is that the RRH protocol successfully improved this degradation in performance whenever a nested NEMO structure emerged. However, we also observed that the degree to which RRH improved the performance of the mountain rescue network was extremely dependent on the specific network configuration used and in many cases it only resulted in a minor improvement. Therefore, whilst we recognise the benefits that the RRH technique can provide and conclude that it represents an all round better solution to our identified domain than the NEMO Basic Support Protocol, we still feel that it introduces an unacceptable level of inefficiency and would be unsuitable for a full scale deployment of our scenario.

With the knowledge gained from this study, our next step will be to explore the feasibility of using existing MANET protocols to efficiently support Mobile Networks. MANET protocols are typically designed to adapt quickly and efficiently to complex models of mobility. However, these protocols (such as AODV [10] and OSLR [11]) have in the past been approached from the perspective of individual nodes forming Mobile Ad Hoc Networks with one another as opposed to Mobile Networks of nodes forming Ad Hoc Networks with other Mobile Networks of nodes. Part of our future work will be to carry out further research into this area and confront the associated challenges that arise. Another important consideration is that a perfect solution to this scenario may be one which incorporates the ability of MANET routing protocols to adapt quickly and efficiently to constantly changing network topologies with NEMO's ability to support networks of mobility unaware nodes that can be contacted at any time by Correspondent Nodes external to the Mobile Network infrastructure. Therefore our future work will also explore the feasibility of a MANEMO (MANET+NEMO) approach.

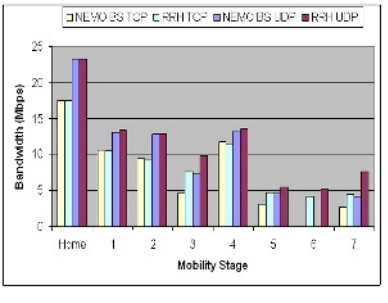


Fig. 8. UDP & TCP Results Graph

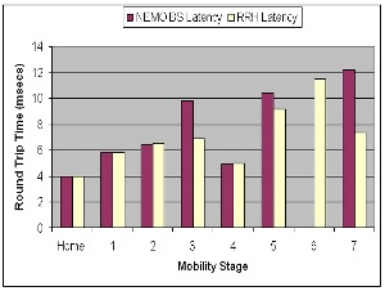


Fig. 9. Latency Results Graph

6 Conclusion

In this paper we have shown how the NEMO Basic Support Protocol could be used to produce a working solution to the mountain rescue scenario that would allow us to capitalise on the benefits mentioned above. However, although a basic working solution

could be produced, what we have also illustrated is the infeasibility of this approach due to the inefficiencies it imposes. Pursuing this solution using NEMO BS would result in much of the available network resources being wasted. In comparison to NEMO BS, we also considered the NEMO Route Optimisation technique RRH's ability to improve the overall performance we could expect to achieve from our testbed but whilst still supporting mobile networks. The results conclusively proved that the RRH technique could provide significant performance improvements in many of the mobility scenarios we would expect to occur in our mountain rescue scenario. Even with these performance improvements however, we still feel that it would be inadequate to develop a solution around the NEMO + RRH technique to support real-time applications. In most of our mobility scenarios, even with RRH, it can still be expected that the end-to-end path setup is over twice the length of the best possible route. This can be attributed to the fact that the NEMO model was fundamentally not designed to support the types of mobility we identify in the mountain rescue scenario, but most importantly, does not assume we have total control over the Mobile Network structure as a whole. With the Mountain Rescue Network we do control all of the networks and therefore we should capitalise on this advantage.

Acknowledgements

The authors wish to thank Cisco Systems for their support of the Mobile Networks project at Lancaster University under which this work was completed.

References

1. IEEE 802.11 Working Group Homepage. http://grouper.ieee.org/groups/802/11/.
2. IEEE 802.16 Task Group e Homepage. http://www.ieee802.org/16/tge/.
3. Wimax/802.16 Technological Overview. http://www.wimaxforum.org/technology.
4. P. Thubert et al. "NEMO Basic Support Protocol". IETF RFC 2693, January 2005.
5. M. Molteni P. Thubert. "IPv6 Reverse Routing Header and its application to Mobile Networks". IETF Draft (Work In Progress), June 2004.
6. Cisco 3200 Series Mobile Access Router Homepage. http://www.cisco.com/go/3200.
7. Lancaster University Computing Department Website. http://www.comp.lancs.ac.uk/.
8. Lancaster University Network Mobility Website. http://www.network-mobility.org/.
9. C. Ng, P. Thubert, M. Watari, and F. Zhao. "Network Mobility Route Optimization Problem Statement". NEMO Working Group IETF Draft (Work In Progress), October 2005.
10. S. Das C. Perkins, E. Belding-Royer. "Ad hoc On-Demand Distance Vector (AODV) Routing". IETF Request For Comments 3561, July 2003.
11. P. Jacquet T. Clausen. "Optimized Link State Routing Protocol (OLSR)". IETF Request For Comments 3626, October 2003.

Route Enhancement Scheme Using HMIP in Heterogeneous Wireless Data Networks

Jaeho Lee and Jaiyong Lee

Dept. of Electrical and Electronic Engineering,
Yonsei University, Seoul, Republic of Korea
ljh@nca.or.kr, jyl@yonsei.ac.kr

Abstract. With building heterogeneous wireless data networks becoming more popular - IEEE 802.11, 802.15, 802.16 and 3G - how to realize seamless mobility is fast becoming a serious issue. Therefore several proposals related to the media independent handoff are suggested in standard bodies including 3GPP, IETF and IEEE 802.21. In this paper, we will present ways to address inefficiency in the routing problem found after adopting the IP-based integrated architecture especially in both MS and CoN moving environment. Also we will propose a scheme of route enhancement and better data transfer rate by using HMIP(Hierarchical Mobile IP) in heterogeneous wireless data networks.

1 Introduction

The need for creating an ubiquitous computing environment has resulted in the demand for various wireless data networks such as IEEE 802.11 (WLAN), 802.15 (WPAN), 802.16 (WiBro, WiMAX), 3G, etc. These wireless technologies have different properties in terms of service coverage, bandwidth and cost. It is expected that assorted wireless technologies will be used depending on what is required. Especially, IEEE 802.16 (WiBro, WiMAX) and HSDPA (High Speed Download Packet Access) will become drivers accelerating broadband services in a wireless environment from the middle of 2006. Therefore, the seamless mobility function between heterogeneous wireless data networks will be one of the most significant issues. In order to provide the mobility function between 3G and WLAN, 3GPP introduces not only the interworking architecture but also a PDG (Packet Data Gateway) which establishes a tunnel from the user mobile station to the core network.[1,3] However, PDG poses a risk of causing inefficiency in routing, therefore the route enhancement scheme using HMIP (Hierarchical Mobile IP) is proposed in this paper.

2 Interworking Architecture and PDG

One of the most important requirements for the ALL-IP-based next generation network is the performance of mobility. 3GPP suggests interworking scenarios of 3G and WLAN. These scenarios are organized in two ways. First, a loosely

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 21–30, 2006.

coupled interworking model. Second, a tightly coupled interworking model. In a loosely coupled model, billing, authentication and basic mobility services are provided, but it is hard to support real time applications such as VoIP and streaming services during the handoff between heterogeneous wireless networks. However, in a tightly coupled model, the seamless mobility service can be provided as well as the authentication and the integrated billing[1]. In order to support the seamless mobility function between different datalink networks, the vertical handoff is required. The vertical handoff can provide seamless connectivity between PAN, LAN, MAN and WAN. Usually horizontal handoff means the flat mobility model between homogeneous datalink networks. On the other hand, the vertical handoff means the perpendicular mobility model between heterogeneous datalink layers within the same terminal and a multi-interface environment. IEEE 802.21 MIH deals with vertical handoff requirements. The logical diagram of the interworking function using PDG is shown in Figure 1[3].

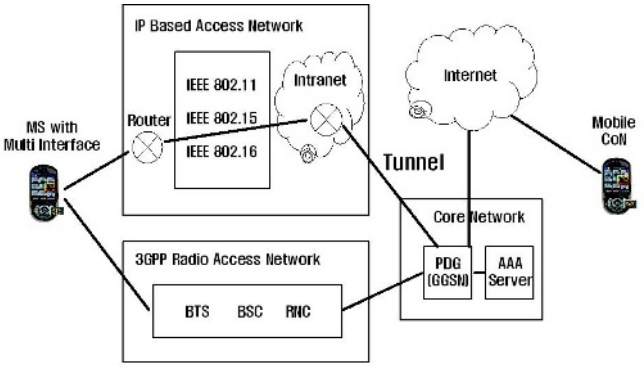


Fig. 1. Logical Structure for Interworking

MS (Mobile Station) has multi-stack network interfaces which can provide multi-network connectivity. If MS is within the coverage range of 3G, MS could take on a connectivity with 3G RAN(Radio Access Network) and then it would receive an IP address from PDG in the 3GPP CN (Core Network). On the other hand, if MS is within the coverage range of IEEE 802.x networks - such as IEEE 802.11, 802.15 and 802.16 - MS has to take on a connectivity with IP-based 802.x networks first. This means that all the application service packets between MS and CoN must go through IP-based access networks. PDG plays an important role in ensuring that the tunnel acts with an interworking function. For example, let's assume that MS is in the 802.x networks and without PDG, packets are forwarded through Internet backbone network instead of the 3G core network. Then it is hard to support authentication and billing services because packets don't visit 3GPP core network where AAA (Authentication Authorization Accounting) and billing servers are located. So PDG is an essential device for providing the interworking function in heterogeneous data networks.

3 Problem Statement

According to a series of related research[1][3], MS selects the best connection services among 3G, 802.11 (WLAN), 802.16 (WiBro, WiMAX), etc, depending on what is required. Packets between MS and CoN must go through PDG, so the closest PDG from MS usually makes the tunnel between MS and PDG. The tunneling mechanism can maintain application session continuity by providing network mobility for MS, but it has a potential problem of inefficient traffic routing such as the triangular problem in mobile IP shown in Figure 2. If CoN is stationary and traffic routing between PDG and CoN is optimized, traffic routs between MS and CoN will be optimized regardless of MS mobility. But in the case of mobile CoN, the best route can change according to MS and CoN mobility. For example, if MS and CoN are moving in the same direction simultaneously, the best route between MS and CoN can be established via PDG3. But if there is no mobility function on PDGs, the tunnel between MS and PDG cannot change. So as shown in Figure 2, traffic routing is established via PDG1 with MS and CoN being far away from PDG1. This kind of inefficiency in routing could cause additional traffic delays and overall decline in performance. Unnecessary backbone network overload could also occur in the PDG1 area.

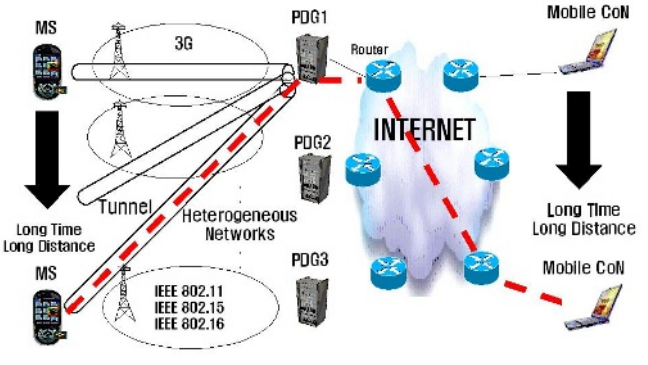


Fig. 2. Inefficient Route between MS and CoN

4 Proposed Scheme

In the 2-tier IP structure which is adopted by the legacy architecture, an application binds with Home Address. CoA (Care of Address) is used when MS is in the coverage range of IP-based access network, such as WLAN or WiBro (WiMAX). In the 3G access network, only HA is necessary because 3G RAN (Radio Access Network) doesn't require an IP address to make a local connection. In this paper, the 3-tier IP structure is proposed as shown in Figure 3[2][6]. One more IP address layer is used for the PDG mobility. It has a corresponding

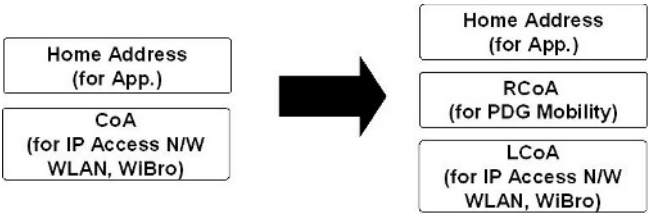


Fig. 3. 3-Tier HMIP Structure

structure of HMIP (Hierarchical Mobile IP address), but HMIP was designed to reduce signals while realizing local mobility. On the other hand, this 3-tier IP structure is designed for enhancing route between PDGs. The detailed scheme is shown in Figure 3.

First, HA is the address for supporting application continuity. This HA does not change until the current network is disconnected. Second, RCoA (Regional Care of Address) is the address for providing the PDG mobility. In this case, TEP (Tunnel End Point) between MS and PDG can change according to which direction MS and CoN move in. Third, LCoA (Local Care of Address) is the address for IP-based access network connectivity, such as WLAN or WiBro (WiMAX), because if MS is in the area of IP-based network, an IP address is required between MS and AR (Access Router).

If MS receives a Home Address in the 3G network – it only uses the Home Address and RCoA provided by PDG1 – and then MS doesn't have to use LCoA. However, if MS sets up LCoA on condition that it moves to the IP-based access network, the Home Address and RCoA are retained until it moves within the intra-PDG domain. Inter-PDG handoff is applied when MS moves to another PDG domain. Then MS changes its RCoA to a new RCoA which is provided by the new PDG. In the end, MS doesn't need to change it's Home Address in order to maintain its application session. In this scheme, PDG plays a role of MAP (Mobile Anchor Point) in HMIP. Figure 4 shows how IP addresses (RCoA, LCoA) change according to MS movement. Here is how it works. First, MS establishes the Home Address and RCoA, letting IP_{PDG11} be the IP address assigned by PDG1. Then, both Home address and RCoA become IP_{PDG11} because MS is in the coverage range of 3G RAN, so PDG1 assigns the IP address to MS. When MS moves to another IEEE 802.x IP-based access network, intra-PDG handoff, only the LCoA address will change. After moving to another PDG domain network, inter-PDG handoff, MS updates its RCoA from the nearest router and LCoA from PDG2 respectively. PDGs play a role of MAP in HMIP. PDG and AR maintain IP subnet blocks and assign them to MS.

- IP_{PDG1} IP Subnet block used in PDG1
- IP_{PDG11} IP Address assigned by PDG1, used in MS
- IP_{AR1} IP Subnet block used in AR1
- IP_{AR11} IP Address assigned by AR1, used in MS
- IP_{PDG2} IP Subnet block used in PDG2

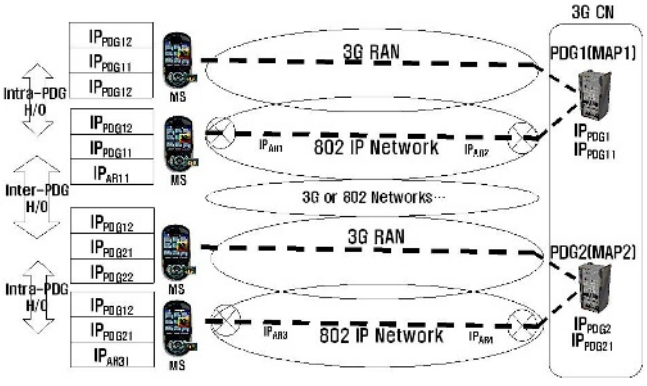


Fig. 4. IP Structure Status

- IP_{PDG21} IP Address assigned by PDG2, used in MS
- IP_{AR3} IP Subnet block used in AR3
- IP_{AR31} IP Address assigned by AR3, used in MS

In this proposed scheme, different kinds of tunneling mechanisms are used in packet forwarding from MS to CoN. For example as shown in Figure 5, generated packets from CoN are delivered to PDG with the encapsulated frame from a point where RCoA is the destination address of the packet. PDG converts the destination address from RCoA to LCoA in accordance with the address mapping table in PDG. PDG forwards the packet from PDG to MS conforming to the change in the destination address. As a result, the packet from CoN will be delivered to CoN's encapsulated destination address while maintaining session continuity.

5 Analysis

5.1 Analysis of PDG Boundary Crossing Rate

The handoff ratio between PDGs can be modelled with regard to network coverage, the speed of MS and the amount of MS, etc. Therefore mobility can be explained by the crossing-rate analysis model[4]. In order to get a crossing rate between PDGs, we needed to calculate the number of cell rings assuming it is in a cellular structure. A cell has a hexagonal shape and the length of a side is 'a'. Let R be the distance from the center, we can have the expression of number of cells as (1).

$$N(R) = \sum_{r=1}^{R} 6r + 1 = 6\frac{R(R+1)}{2} + 1 = 3R(R+1) + 1 \tag{1}$$

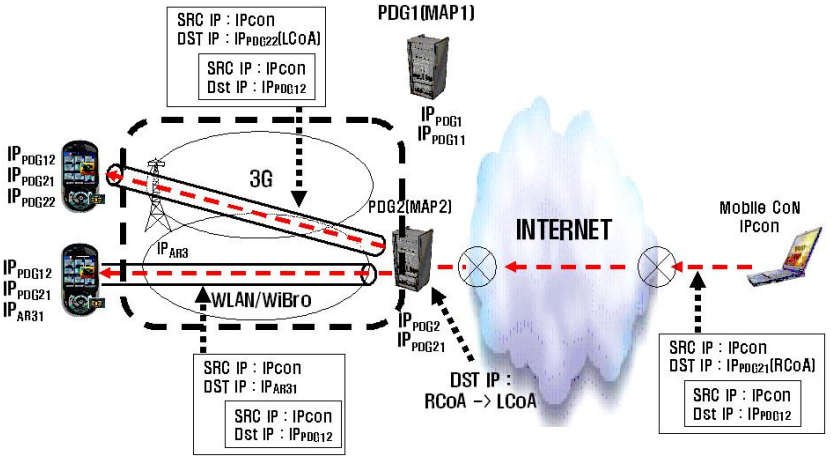


Fig. 5. Tunneling Mechanism

We apply the fluid flow model which can show the mobility rate of in a steady state in a moving environment and (2) is the basic expression of the fluid flow model. The crossing rate of the fluid flow model is proportional to the number and velocity of nodes as well as the length of the cell boundary. Each n, v, l and A indicate the number of nodes, velocity, the length of boundary and the cell area. However, the crossing rate is inverse proportional to the cell coverage range A. The intra-PDG handoff is executed within the same PDG and the inter-PDG handoff is performed between different PDGs. Therefore the total crossing rate is the sum of the internal crossing rate and the boundary crossing rate as (3). The entire area of R can be depicted as (4) and it is derived from (1). We can also have the boundary length of a cell as (5).

$$C = \frac{\rho v l}{\pi} = \frac{nvl}{\pi A} \quad where \quad \rho = \frac{n}{A} \tag{2}$$

$$C_{internal} = C_{total} - C_{boundary} \tag{3}$$

$$A(R) = N(R) \times A = N(R) \times \frac{3\sqrt{3}}{2}a^2 = \{3R(R+1)+1\} \times \frac{3\sqrt{3}}{2}a^2 \tag{4}$$

$$L(R) = 6a(2R+1) \tag{5}$$

Therefore, the total crossing rate which contains inter and intra handoff of PDG can be expressed as (6) with the combination of (1),(2) and (4). Here, the boundary length of cells which are composed of the number of N(R) can be suggested as (7).

$$C_{total}(R) = \frac{nvN(R)}{\pi A} = \frac{4nv(2R+1)(3R^2+3R+1)}{\pi\sqrt{3}a^2} \tag{6}$$

$$C_{boundary}(R) = \frac{nvL(R)}{\pi A(R)} = \frac{4nv(2R+1)}{\pi\sqrt{3}(3R^2+3R+1)a} \tag{7}$$

$C_{internal}$ means that MS moves within intra-PDG coverage range. On the other hand, $C_{boundary}$ means that MS moves between inter-PDG coverage range. As a result, $C_{boundary}$ (7) is used to obtain the crossing rate between PDGs. The Crossing rate is altered according to R, number of nodes and velocity. Figure 6 shows the condition of the crossing rate when the number of nodes range from 20 to 100 and velocity is from 10Km to 90Km for each other where the side length of the cell is 0.5Km. According to Figure 6, inter-PDG handoff can occur very frequently in proportion to the number of nodes and velocity. Figure 6 also shows the relationship of the crossing rate with R. If the R is decreased, the coverage of PDG will be small. As a result, the crossing and handoff rate will be high. On the other hand, if the R is increased, the coverage of PDG will be large, therefore the crossing and handoff rate will be low. However, the low crossing and handoff rate means that the possibility of inefficient routing path could be higher because of the wide coverage range.

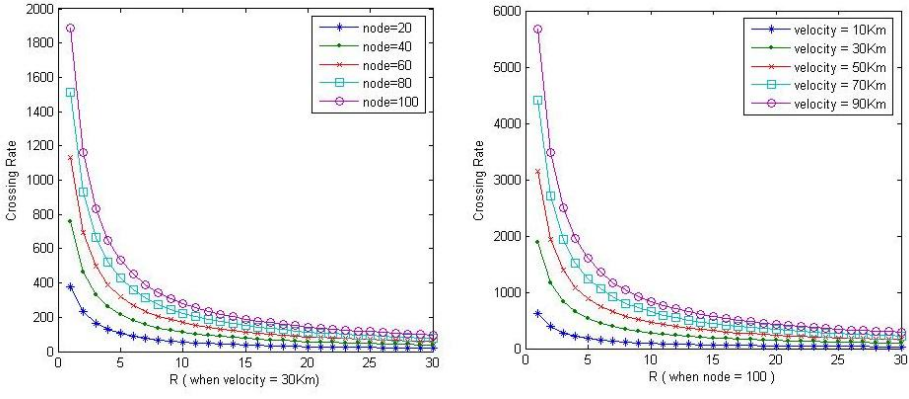


Fig. 6. Crossing Rate with Number of Nodes and Velocity

5.2 Performance Analysis

In the 2-tier structure, with the movement of MS and CoN, the end-to-end transmission time can be delayed. This delay time lowers throughput between MS and CoN. Let λ denote the required bandwidth of the application and let ω denote the additional delay time according to the movement of MS or CoN. Let Dt_{PDG} be the handoff delay time by the reason of changing the tunneling end-point as well as HMIP with Fast handoff.

- λ : Required application bandwidth
- ω : Additional delay time
- Dt_{PDG} : Handoff delay time between PDGs
- $t1$: Time to transfer 1 bit
- $t2$: Time to transfer 1 bit with additional delay time ω
- Th : Throughput, bit per seconds

$$t1 = \frac{1}{\lambda} \tag{8}$$

$$t2 = (\frac{1}{\lambda} + \omega) \tag{9}$$

$$Th = \frac{\lambda}{1 + \lambda\omega} \tag{10}$$

$$\sum(\sum_{\omega}^{inter-PDG-handoff} \frac{\lambda}{1 + \lambda\omega} + Dt_{PDG}) \tag{11}$$

The next generation mobile application requires the bandwidth such as 64Kbps (VoIP), 300Kbs (Mobile Video Phone), 500Kbps (VoD), 1000Kbps (HQ VoD). If the additional delay time occurs due to the movement of MS or CoN, the end-to-end transmission time would be delayed. Therefore according to (10) the total throughput between MS and CoN will be lowered shown in Figure 7. Especially, if the delay time per bit is constantly increased, QoS of the application could encounter the serious condition.

3-tier IP structure can prevent the throughput declining caused by the reestablishment of the tunnel between MS and PDG, the entire route can be enhanced. Therefore the delay time between MS and CoN can be minimized, and the overall throughput becomes the normal condition. However Dt_{PDG} should be considered to be the additional handoff delay time between PDGs. (11) is the expression of the cumulative data transferred in 3-tier IP structure environment.

Figure 8 shows the throughput comparison of 2-tier IP structure and 3-tier IP structure. These four graphs show the result of modeling assumptions within some conditions. First, MS and CoN are moving at the same time. Second, data is transferred in two hours(7200 seconds) with 300 Kbps. Third, intra-PDG handoff occurs in every minutes and this intra-handoff causes the additional delay time 0.0001 sec (= 0.1 ms) in the data transmission between MS and CoN. Forth, inter-PDG handoff occurs every ten minutes with 0.1 sec delay time (= 100ms) only in right upper and right bottom graphs. Therefore we can get the result that the proposed scheme shows the better performance in throughput and delay time. With the movement of MS and CoN, the left upper graph indicates constantly declining performance in the legacy environment. On the other hand, the right upper graph displays a relatively steady performance in the proposed environment. When the inter-PDG handoff takes place, the throughput steadily declines every 600 seconds before recovering to its original level. Inter-PDG handoff overhead can be accommodated because of the

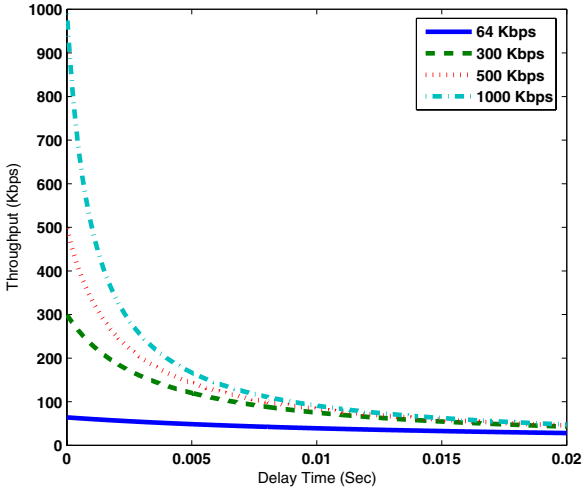


Fig. 7. Additional Delay Time and Throughput

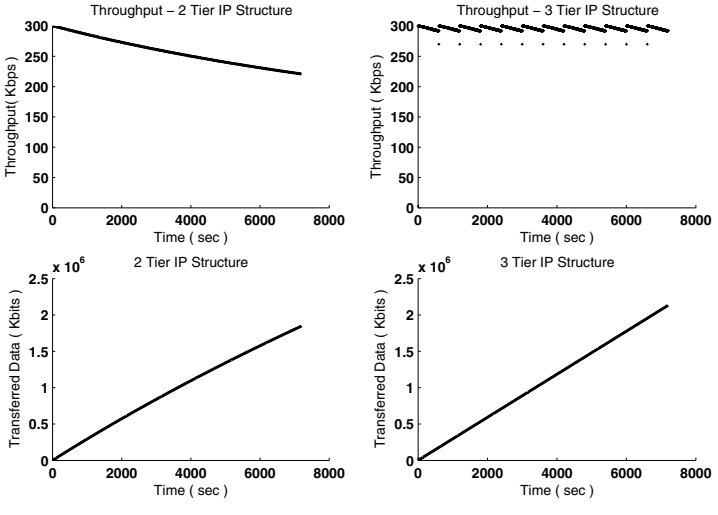


Fig. 8. Performance Comparison of Legacy and Proposed Scheme

enhancement of overall throughput. The left and right bottom graphs shows the cumulative data transfer with the time sequence. In the optimal condition of 300 Kbps, 2,160 Mbits can be transmitted in two hours. In 2-tier IP structure, 1,847 Mbits are transmitted and in 3-tier IP structure 2,131 Mbits are transmitted.

6 Conclusion

In a heterogeneous wireless data network environment, seamless mobility is the essential function of supporting quality of service during the handoff. PDG also becomes the important device in providing mobility. In this paper, we propose not only the potential problem of inefficient routing path in PDG-based interworking environment, but also provide a solution in enhancing the routing path through analysis and simulation. Applying the HMIP scheme to PDG entails additional costs and time because of increasing message exchanges. However, the overall cost for transferring mobile multimedia data will be reduced by the improved routing path. In order to design enhanced heterogeneous networks, more specific message exchange procedures, such as IEEE 802.21 MIH (Media Independent Handoff), need to be introduced in future work.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)(IITA-2005-C1090-0502-0012).

References

1. 3GPP TS 23.234 v6.0.0 : 3GPP system to Wireless Local Area Network(WLAN) interworking. Mar. (2004)
2. H. Soliman, K. El-Malki : Hierarchical Mobile IPv6 mobility management(HMIPv6). draft-ietf-mobileip-hmipv6-08.txt
3. Alan Carlton, Marian Rudolf,Juan-Carlos Zuniga, Ulises Olvera : MEDIA INDEPENDENT HANDOVER Functions and Services Specification-Initial Proposal for 802.21. IEEE 802.21 MIH, Mar. (2003)
4. Sangheon Pack, Yanghee Choi : A Study on Performance of Hierarchical Mobile IPv6 in IP-Based Cellular Networks. IEICE Trans. Commun. Vol.E87-B, No.3 Mar.(2004)
5. Kyung-ah Kim, Jong-deok Kim, Chong-kwon Kim, Jae-yoon Park : An enhanced Handoff Mechanism for Cellular IP. Lecture Notes in Computer Science(LNCS), Springer, vol.3090. pp 164-173. Aug. (2004)
6. H.Soliman : Mobile IPv6-Mobility in a Wireless Internet. Addison Wesley.(2004)

Performance Evaluation of TCP Variants to Downward Vertical Handoff

Woojin Seok[1], Yoonjoo Kwon[1], Okhwan Byeon[1], and Sang-Ha Kim[2]

[1] Supercomputing Center, Korea Institute of Science and Technology Information, Taejon, Korea
{wjseok, yulli, ohbyeon}@kisti.re.kr
[2] Department of Computer Engineering, Chungnam National University, Taejon, Korea
shkim@cnu.ac.kr

Abstract. The behavior of a regular TCP is not good to adjust quickly to the new higher available bandwidth of a newly arrived network after a downward vertical handoff. In this paper, we compare three TCPs by mathematical and simulation analysis to the downward vertical handoff. With mathematical analysis, we compare the link utilization of NewReno TCP, Scalable TCP, and BIC TCP. Among them, the performance of BIC TCP shows best, and it gets better than the other two as RTT increases. With simulation analysis, we compare the throughput of NewReno TCP and BIC TCP. BIC TCP has higher throughput than NewReno TCP, and the performance gain by BIC TCP gets larger with longer RTT. BIC TCP shows the best performance to the downward vertical handoff among the three TCPs. That is because BIC TCP can increase the congestion window size exponentially to find a new available bandwidth with a connection sustained.

1 Introduction

TCP is a dominant protocol of transportation layer. Its performance depends on the network conditions such as available bandwidth, latency, and so on. TCP of a mobile node moving over wireless networks is additionally affected by lossy link channel, handoff, and so on [1],[2],[3].

When a mobile node moves over heterogenous networks, it meets a new type of handoff between a WLAN(Wireless LAN) and a 3G cellular network, and we call it vertical handoff. Especially, we call it a downward vertical handoff when the mobile node moves from a cellular network to a WLAN, and an upward vertical handoff, vice versa. For this network environment, the conditions that affect TCP performance are dramatic changes of bandwidth, latency, and buffer size between a sender and a receiver [4],[5],[6].

Dramatic changes of latency and bandwidth due to the vertical handoff produce a bunch of packet reordering, link underutilization, packet overflow, spurious timeout and so on [7],[8]. NewReno TCP shows good performance on the vertical handoff due to its great feature about the multiple packet losses that can be produced from lots of packet reordering [7].

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 31–41, 2006.

Even though NewReno TCP has a good performance among famous TCP variants, it still suffers from the link underutilization when it moves from a 3G cellular network to a WLAN network, and generates a downward vertical handoff. Then its CWND(congestion window size), already converged to the available bandwidth of a 3G cellular network, should increase to the new available bandwidth of a WLAN that is relatively very high. But the congestion avoidance mode of NewReno TCP can advance just one CWND after one RTT(Round Trip Time), and it probably takes long time to tolerate.

In this paper, we compare and evaluate the performances of NewReno TCP, BIC TCP, Scalable TCP by both mathematical and simulation analysis to downward vertical handoff.

2 Related Works

In early stage, the authors in [9] compared TCP variants and they summarized the performances of the TCP variants. The comparison and evaluation of the performances of TCP variants were focused on homogeneous networks with consistent end-to-end states. The end-to-end path of an established connection was not changed, so the available bandwidth and latency between the two ends does not change much while the connection was sustained. The only possibilities, that the available bandwidth and latency can change, are caused by the cross-traffic.

In the heterogeneous networking stage, the bandwidth and the latency of end-to-end path will be dramatically changed by a vertical handoff. In [10], they indicated the worst performance of TCP on vertical handoff. They said that the vertical handoff made packet reordering and TCP reacted with spurious packet retransmissions. Duplicate packets being sent unnecessarily reduced the bandwidth efficiency of the wireless channel.

In [7], they examined the behavior of TCP variants when mobile nodes performed a handover from a high delay radio network to a radio network with less delay. They said a high degree of packet reordering could be observed to the situations that a mobile node moved from low quality link to high quality link. NewReno TCP that provided means to deal with multiple packet losses in a transmission window showed better handover behavior.

In [8], they identified TCP and HTTP performance problems during vertical handovers using Mobile IPv6. They indicated problems during handover from slow network like GPRS to fast network like wireless LAN, and they were link underutilization and packet reordering.

[4] and [5] suggested a new TCP scheme to get over the way that it increases one CWND for one RTT on a downward vertical handoff. They proposed to use exponentially increasing way after a vertical handoff to quickly adjust to the new network bandwidth. Their proposed TCP needed the vertical handoff triggering from layer 2 and made a sender or a receiver know the event of the vertical handoff by using TCP option field. Their idea had good performance, but it had to deploy their TCP on both sides of sender and receiver, and handoff information had to be passed from layer 2 to layer 4 directly.

3 Motivations

When a mobile node moves from a 3G cellular network to a WLAN, it is highly probable that the current CWND was converged to the available bandwidth of the 3G cellular network. So the time that the CWND will get to the new available bandwidth of WLAN probably takes very long time because, in the congestion avoidance mode, the CWND increases just one for one RTT.

The similar problem was issued in the research area of fast and long distance network. Inefficient way of increasing CWND makes TCP take long latency due to the high link capacity and long RTT. Some of TCP variants were proposed to overcome this problem. Scalable TCP revises the agility of AIMD(Additive Increase Multiplicative Decrease) property against taking very long time to reach the available bandwidth if TCP is in the stage of congestion avoidance mode. The CWND deflates by 1/8 of the previous CWND, and increases 0.01 for each ACK segment later on. The behavior of Scalable TCP doubles up the current CWND during around 70 RTTs [11].

BIC TCP searches the new available bandwidth with the combined way of linear and binary search after 3 duplicate packets. If the window grows past the last available bandwidth, the new available bandwidth must be larger than the last one. Then BIC TCP enters a new phase called max probing, and the growth function during max probing is exponential [12].

In this paper, we will show and compare the performance of NewReno TCP, Scalable TCP, and BIC TCP with mathematical analysis to evaluate the link utilization on a downward vertical handoff, and with simulation analysis to evaluate the throughput after the handoff.

4 Performance Evaluations

The assumption that TCP is in congestion avoidance mode when the downward handoff happens is applied to both analyses. The assumption that there is no segment loss is applied to the mathematical analysis, and the assumption that there may be segment losses is applied to the simulation analysis.

4.1 Mathematical Analysis of Link Utilization

Link utilization, U, is the time spent in transmitting segments(T_f) over the total time(T_t) including T_f and the time to spent in waiting ACK segment(T_s).

$$U = \frac{T_f}{T_t} = \frac{T_f}{T_f + T_s}.$$

O is the size of an object to transmit, and R is the transmission rate, then T_f is $\frac{O}{R}$ for all three TCPs. The unique features of different TCPs affect T_s, and those will be covered in the next sub-section.

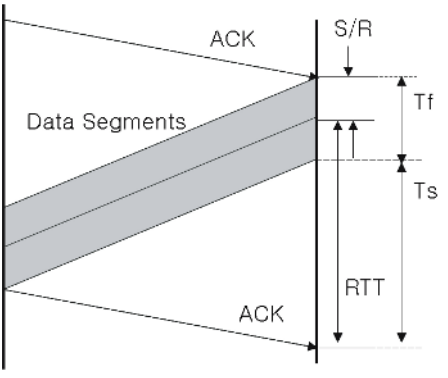


Fig. 1. The Flow of Segments

The Comparisons of Three TCPs. T_s of NewReno TCP is as follows, C is CWND, the half of the previous value of CWND, and P is the min value of {Q, K-1} [1].

$$T_s = \sum_{j=1}^{p} (RTT + \frac{S}{R} - (C + j - 1) \times \frac{S}{R}).$$

K is the number of window that covers the object, and Q is the number of times the server would be stall if the object contained an infinite number of segments. Then, K and Q can be calculated as follows.

$$K = min\{j | C + (C + 1) + (C + 2) + ... + (C + j - 1) \geq \frac{O}{S}\}$$

$$= \frac{(1 - 2C + \sqrt{4 \times C^2 - 4 \times C + 1 + 8 \times \frac{O}{S}})}{2}.$$

$$Q = max\{j | RTT + \frac{S}{R} - (C + j - 1) \times \frac{S}{R} \geq 0\}$$

$$= 2 - C + RTT \times \frac{R}{S}.$$

T_s of Scalable TCP is as follows, and the way to solve is almost same with that of NewReno TCP. The constant, 70, means that Scalable TCP doubles CWND after 70 RTTs.

$$T_s = \sum_{j=1}^{p} (RTT + \frac{S}{R} - 2^{(\lceil \frac{j}{70} \rceil - 1)} \times \frac{C \times S}{R}).$$

[1] We used the methodology of the analysis from [13].

With the same way done for NewReno TCP, K and Q can be calculated as follows. We assume that 35 times of transmission with the same CWND would be done rather than 70 times before the final transmission. That is, the last CWND update is done and TCP uses it 35 times rather than 70 times. 35 means the average in the sense of the probability of uniform distribution.

$$K \approx min\{j = 70 \times j' - 35 | 70 \times C \sum_{l=1}^{j'} 2^{(l-1)} \geq \frac{O}{S}\}$$

$$= 70 \times log \frac{\frac{O}{S} + 70 \times C}{70 \times C} - 35.$$

$$Q \approx max\{j = 70 \times j' - 35 | RTT + \frac{S}{R} - C \times 2^{j'-1} \times \frac{S}{R} \geq 0\}$$

$$= 70 \times (1 + log \frac{RTT + \frac{S}{R}}{C \times \frac{S}{R}}) - 35.$$

T_s of BIC TCP is the sum of T_{phase1} and T_{phase2}. T_{phase1} is the accumulated waiting time in increasing CWND to the last highest CWND immediately after 3 duplicate ACKs on the downward vertical handoff, then T_{phase2} is the accumulated waiting time in increasing CWND exponentially after the CWND gets higher than the last highest CWND. Let f mean the function of CWND, and k is the subscript to represent the order of stalls during T_{phase1} and C is the last highest CWND, then T_{phase1} is as follows.

$$f_1 = 0.875 \times C.$$

$$f_j = f_{j-1} + \frac{C - f_{j-1}}{2},$$

until

$$C - f_{j-1} > S_{min},$$

where

$$S_{min} = 0.01.$$

Then,

$$T_{phase1} = \sum_j (RTT + \frac{S}{R} - f_j \times \frac{S}{R}).$$

We assume the object is large enough to send during T_{phase1}, then, the quantity of the object left to send in T_{phase2} is b, and S is the segment size in byte.

$$b = O - S \times (f_0 + f_1 + \cdots + f_{j-1})$$

$$= O - S \times \sum_{i=1}^{j} f_{i-1}.$$

During T_{phase2}, CWND increases exponentially. The function of CWND can be represented by g_j as follows where P is the min value of {Q, K-1}.

$$g_j = C + 2^{j-1}.$$

$$T_{phase2} = \sum_{j=1}^{p}(RTT + \frac{S}{R} - g_j \times \frac{S}{R}).$$

$$K = min\{j|\sum_{l=1}^{j} g_l \geq \frac{b}{S}\}.$$

$$Q = max\{j|RTT + \frac{S}{R} - g_j \times \frac{S}{R}\}.$$

Results and Discussions. From the results of the previous sub-section, we can calculate the link utilization. We assume that a mobile node moves from a 3G cellular network to a WLAN. The mobile node sends one big file, 4 Mbytes, and it has 3 Mbytes to send after the handoff. The average transmission rate of the WLAN is 2 Mbps, and the RTT varies from 100ms to 500ms. The number of CWND immediate after the handoff can be calculated as $\frac{144Kbps}{(536bytes \times 8)}$ if we assume the bandwidth of a 3G cellular network is 144Kbps and MSS(Maximum Segments Size) is 536 bytes. The time spent in transmitting the segments, T_f, is 12.0 seconds because 3Mbytes data with 2Mbps transmission rate produces 12.0 seconds.

Table 1. Link Utilizations of Three TCPs

RTT	NewReno TCP				Scalable TCP				BIC TCP			
	U	Ts	K	Q	U	Ts	K	Q	U	Ts	K	Q
100ms	0.919	1.05	90.7	31.8	0.831	2.43	97.7	83.80	0.967	0.40	13	5
200ms	0.647	6.52	90.7	78.4	0.506	11.71	97.7	152.7	0.860	1.95	13	7
300ms	0.439	15.29	90.7	125.1	0.360	21.31	97.7	193.3	0.768	3.61	13	8
400ms	0.331	24.19	90.7	171.7	0.279	30.91	97.7	222.2	0.698	5.19	13	9
500ms	0.266	33.09	90.7	218.4	0.228	40.51	97.7	244.6	0.628	7.09	13	9

In the 5 RTT cases, NewReno TCP shows better performance than Scalable TCP, but worse than BIC TCP. BIC TCP shows best performance among them, and gets better as RTT gets larger. Scalable TCP have very small coefficient, 0.01, to double CWND because it is devised to improve TCP for the high speed networks with long delay.

BIC TCP has very great feature to downward vertical handoff situation. The fact that BIC TCP does not have the congestion avoidance mode used in NewReno TCP works very well with the downward vertical handoff. Even though it is devised to increase the performance over high speed networking

with long distance, it also can give great performance to the downward vertical handoff because it can get to any available bandwidth very quickly with exponentially. From this feature, the performance of BIC TCP is the best among them. As RTT increase, NewReno TCP suffers from the large latency to increase one CWND, but BIC TCP relatively suffers less from the longer RTT because it can double CWND for one RTT.

See the Q factor in Table.1, only BIC TCP has less Q factor than K factor for all RTT cases. As we explained before, the Q factor means the number of stall of transmission by the feature of increase of CWND on the assumption that an object size is infinite, and the K factor means the number of stall by the limited size of an object. That means that the performance of NewReno TCP and Scalable TCP will get worse if the file size gets larger. BIC TCP, however, has less Q factor than K factor to every RTT case in this example. So, BIC TCP will show better performance than the others as the object size gets larger.

4.2 Simulation Analysis of End-to-End Performance

Simulation Scenario and Parameters. In this simulation, we have two nodes over heterogeneous networks where a WLAN is overlaid with a 3G cellular network. One node sends segments with TCP connection by FTP(File Transfer Protocol) program, and the other node receives or discards the segments with depending on the buffer conditions. One of the nodes moves from a 3G cellular network to a WLAN in the middle of sending segments with FTP.

We assume the available bandwidths between the two nodes are 144Kbps for the 3G cellular network and 2Mbps for the WLAN as we used in the mathematical analysis in the previous section, and 300ms end-to-end latency for the 3G cellular and two latencies of 100ms and 200ms for the WLAN. We also assume that the segment size is 536 bytes and we have an infinite size of object to transfer.

Results and Discussions. In Fig.2, the RTT of WLAN is 100ms, and the time to get to the new available bandwidth is about 216 second and about 227 second for BIC TCP and NewReno TCP, respectively. After the handoff, while NewReno TCP increases CWND linearly, BIC TCP increases CWND with binary search and exponential. The way for BIC TCP to find the new available bandwidth is 40% faster than the NewReno TCP; 16 second vs. 27 second. From this reason, the quantity of segments sent by BIC TCP after the handoff gets larger during the duration T1 as shown in Fig.3.

When the handoff occurs, the difference of the quantities of the segments successfully transmitted is 950. BIC TCP sent 950 more segments than NewReno TCP at this point. By the end of the duration T1, the difference is 2100, and it means that BIC TCP shows better performance to the downward vertical handoff. Beyond the duration T1, the performance on the two TCPs does not much depend on their behaviors to the handoff, so that is not considered in this paper.

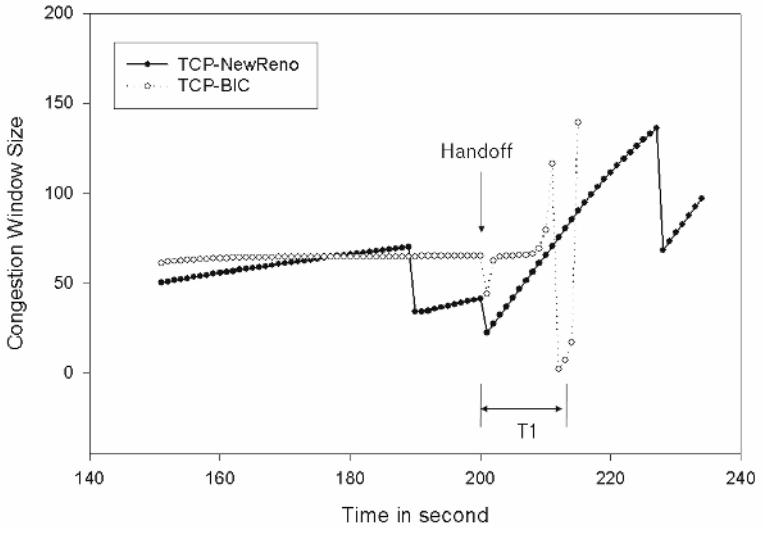


Fig. 2. CWND vs. Time : 100ms RTT case

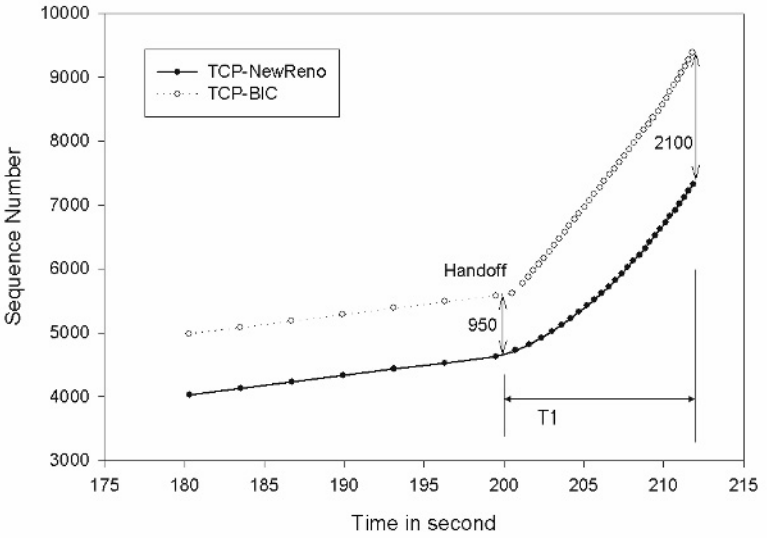


Fig. 3. Sequence Number vs. Time : 100ms RTT case

In Fig.4 and Fig.5, the RTT of the WLAN is 200 ms that is only different factor from the previous simulation. The times to get to the new available bandwidth is about 227 second and 289 second for BIC TCP and NewReno TCP, respectively. BIC TCP is 70% faster than NewReno TCP; 27 second vs. 89 second. The difference of the quantities of segments sent gets larger from 960 to 4600 during

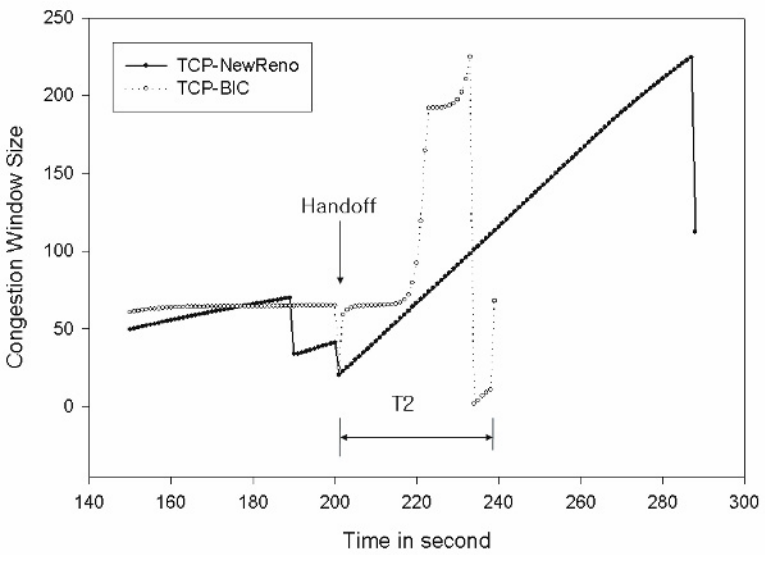


Fig. 4. CWND vs. Time : 200ms RTT case

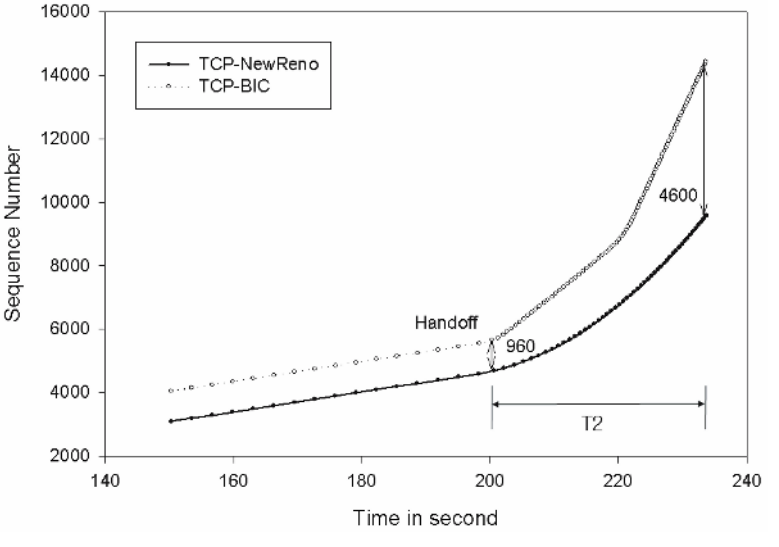


Fig. 5. Sequence Number vs. Time : 200ms RTT case

the duration T2. When RTT is 200 ms, the performance gain by BIC TCP is larger than when RTT is 100 ms. In the longer RTT, the end-to-end pipe that TCP can fill with segments gets larger. So aggressive way to fill the pipe takes shorter time on longer RTT than it does on shorter RTT.

5 Conclusions

In the heterogeneous network, the vertical handoff makes the performance of TCP degraded due to the sudden change of path characteristics. Especially, the handoff from a 3G cellular network to a WLAN makes regular TCP like NewReno TCP takes long time to get to the new available bandwidth of the WLAN because the congestion avoidance mode takes one RTT to increase one CWND.

With mathematical analysis, we compared the link utilization of NewReno TCP, Scalable TCP, and BIC TCP. Among them, BIC TCP showed the best performance, while Scalable TCP showed the worst. The performance of BIC TCP got better than the other two as RTT increased.

With simulation analysis, we compared the throughput of NewReno TCP and BIC TCP in a fixed duration. BIC TCP can send more segments than NewReno TCP can, for the duration immediately after handoff. With longer RTT, the performance gains by BIC TCP got better.

From these results, we can figure out that BIC TCP has great behavior to the downward handoff. That is because BIC TCP increases CWND exponentially to find a new available bandwidth in the middle of a connection.

References

1. Balakrishnan,H., Seshan, S., Amir, E., Katz, R.H.: Improving TCP/IP Performance over Wireless Networks. ACM Conference on Mobile Computing and Networking, Novermber (1995)
2. Ludwig, R., Katz, R.H.: The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions. ACM Computer Communications Review, Vol. 30., January (2000)
3. Goff, T., Moronski, J., Phatak, D.S., Gupta, V.: Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. IEEE Infocom, March (2000)
4. Kim, S.E., John, A.C.: TCP for Seamless Vertical Handoff in Hybrid Mobile Data Networks. IEEE Globecom, December (2003)
5. Kim, S.E., John A.C.: Interworking Between WLANs and 3G Networks : TCP Challenges. IEEE WCNC, March (2004)
6. Gurtov, A., Korhonen, J.: Effect of vertical handovers on performance of TCP-friendly rate control. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 8, July (2004)
7. Hansmann, W., Frank, M., Wolf, M.: Performance Analysis of TCP Handover in a Wireless/Mobile Multi-Radio Environment. IEEE LCN, November (2002)
8. Chakravorty, R., Vidales, P., Subramanian, K., Pratt, I., Crowcroft, J.: Practical Experiences with Wireless Networks Integration using Mobile IPv6. ACM Conference on Mobile Computing and Networking, September (2003)
9. Fall, K., Floyd, S.: Simulation-based comparisons of Tahoe, Reno and SACK TCP, ACM SIGCOMM Computer Communication Review, Vol. 26, July (1996)
10. Schwade, T., Schuler, J.: Investigations on TCP Behavior during Handoff. ITG Workshop, July (2001)

11. Kelly, T.: Scalable TCP: improving performance in highspeed wide area networks. ACM SIGCOMM Computer Communication Review, Vol. 33, April (2003)
12. Xu, L., Harfoush, K., Rhee, I.: Binary Increase Congestion Control for Fast Long-Distance Networks. IEEE INFOCOM, March (2004)
13. Kurose, J.F., Ross, K.W.: Computer Networking. Addison Wesley, (2001) 253-260

Resource Reservation for Multi Classes and Regions over OFDM-Based Multi-cell Environments

Sungjin Lee and Sanghoon Lee*

Wireless Network Lab., Center for IT of Yonsei University, Seoul, Korea, 120-749

Abstract. For OFDMA (Orthogonal Frequency Division Multiple Access)-based broadband systems, a frequency reuse factor of 1 has been highly desirable for more improved channel throughput. However, the forward link capacity is rapidly decreased at the cell boundary region due to the increase in the ICI (InterCell Interference). This paper presents a QoS (Quality of Service) maintenance technique by measuring a SIR-based channel capacity and by performing a radio resource reservation at the initial service setup time. In order to prove the effectiveness of the proposed algorithm, two difference radio resource management schemes are compared associated with multiple classes.

1 Introduction

Great interest in recent years has been devoted to OFDM (Orthogonal Frequency Division Multiplexing)-based mobile communications due to its high spectral efficiency. In order to improve channel throughput and facilitate network deployment, a frequency reuse factor of 1 has been highly considered to be employed. In particular, the forward link capacity of an OFDMA (Orthogonal Frequency Division Multiple Access) or MC-CDMA (Multi-Carrier Code Division Multiple Access) system is rapidly decreased at the cell boundary region due to the increase in the ICI (InterCell Interference). Therefore, depending on the location of each MS (Mobile Station), the radio resource needed to maintain a QoS (Quality of Service) is quite different. For example, if all the MSs are located in the cell boundary, only a few MSs can be supported. On the other hand, if all of them are near the central region of the cell, more users can be supported. However, most of previous researches have been done for a given channel capacity as a function of a variety of system parameters such as the equivalent number of users[3][4], mobility[1], SIR(signal-to-interference ratio)[4][5], resource availability [2] and so on.

In addition, the radio resource depends on the traffic dynamics according to the service class. If the class is a real-time VBR (Variable Bit Rate) video

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment). (IITA-2005-C1090-0502-0012).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 42–51, 2006.

service, it requires more flexible radio resources than other classes, such as a CBR (Constant Bit Rate) voice service. For a given variety of service classes, there would be a way to achieve high channel throughput by sharing a limited resource among different classes. Thus, the amount of radio resources in reserve relies on the following factors : the location and movement of MSs, the service classes and the radio resource management. For the service setup request of an MS, a radio resource reservation is preceded in advance. In such a case, the number of accommodated users may be different according to the efficiency of the radio resource reservation scheme.

To increase the number of accommodated users, efficient radio resource allocation algorithms have been developed. Usually, the number of users is utilized as a major criterion for the call acceptance decision [8][9][10][4]. Among those algorithms, the NB (No Boundary) channel allocation has demonstrated a high channel utilization due to the elegant resource management [8][9]. In [8][9], they proved that the NB scheme is more efficient than the FB(Fixed Boundary) scheme through the simulation.

This paper presents a resource reservation technique for maintaining a QoS negotiated at the service setup time over a channel holding time. In order to absorb the radio resource dynamics according to the location of MSs for the OFDM-based broadband network, a SIR-based capacity measurement is performed based on the partitioned region. In order to provide an efficient radio management, the FB and NB schemes are compared associated with multi-class services. For the performance measurement, four resource reservation schemes are employed: NCFB (Non-divided Cell using Fixed-Boundary), DCFB (Divided Cell using Fixed-Boundary), NCNB (Non-divided Cell using No-Boundary) and DCNB (Divided Cell using No-Boundary).

2 Problem Statement

Without loss of generality, a single class is only considered to state the problem. Fig. 1 depicts the key idea of the proposed algorithm using a simple comparison. In the figure, N_{max} represents the maximum number of admissible users using the average link capacity in (a) named $Case\ I$, or using the region-based link capacity corresponding to the partitioned regions in (b) and (c) named $Case\ II$. In

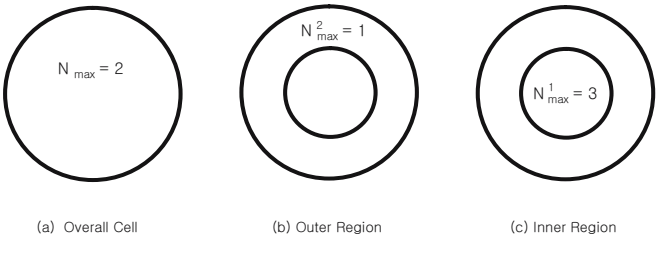


Fig. 1. The maximum number of admissible users per each region

the example, $N_{max} = 2$ for *Case I* in Fig. 1 (a). Fig. 1 (b) depicts *Case II* where all the users reside in the outer region. Due to the ICI effect, more capacity is required so that N_{max} is decreased compared to *Case I*. Thus, N^2_{max} is assumed to be 1. This value is increased as the users move to the inner region. In Fig. 1 (c), $N^1_{max} = 3$ when all the users are in the inner region and no one is in the outer region. In general, let (a, b) represent a state where a and b are the number of users in the inner region and the outer region, respectively. Since $N_{max} = 2$ for *Case I*, the available states are (0,0), (1,0), (0,1), (1,1), (2,0) and (0,2). For *Case II*, the available states are (0,0), (1,0), (0,1), (2,0) and (3,0). The states of (1,1) and (0,2) in *Case I* are not allowable and become error states because the required capacity exceeds the maximum capacity obtained in *Case II*. These states are termed as *error states*. On the other hand, the state (3,0) in *Case II* is an allowable state which is not included in *Case I*. Thus, (3,0) is termed as a *loading state* accrued from the benefit of the region-based link capacity scheme, which does not exist in *Case I*. Thus, more cell partitioning is preformed, more precise state decision can be made.

3 Target System Environment

In order to measure the SIR-based channel capacity, a general MC-CDMA transmitter is employed as shown in Fig.2. In the system, parameters are defined as the

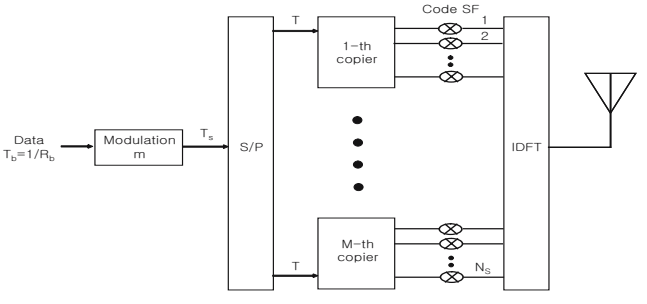


Fig. 2. A general MC-CDMA transmitter

following : BW_T is Total bandwidth of the system(Hz), BW_S is Bandwidth of a subcarrier(Hz), BW_S is Bandwidth of a subcarrier(Hz), SF is Spreading factor, N_{cp} is Number of copiers, N_{sc} is Total number of subcarriers (N_{sc}=N_{cp}·SF), m is Modulation index (ex. QPSK=2, BPSK=1), T_s is Modulated symbol duration, T is MC-CDMA symbol duration and R_b is User bit rate.

Based on the parameters above and T can be expressed by

$$T = \frac{1}{BW_S} = N_{cp}T_s = \frac{m}{R_b} \tag{1}$$

Utilizing the property of OFDM, BW_T is given by $BW_T = \frac{N_{sc}+1}{2} \cdot BW_S$. From T and BW_T, the bit rate is derived as $R_b = \frac{2N_{cp} \cdot BW_T}{N_{sc}+1} \cdot m$. If $N_{sc} \cdot \mathrm{SF} \gg 1$, R_b can be approximated as $R_b = \frac{2BW_T}{SF} \cdot m$

Let an MS be located in an "x" position of the i^{th} BS (i.e., the home BS). $(E_b/N_0)_{i,x}$ is then

$$\left(\frac{E_b}{N_0}\right)_{i,x} = \frac{L_{(i,x:i)} \cdot S \cdot \alpha_{i,k} \cdot \frac{BW_T}{R_b}}{\sum_{j=1}^{N_{oc}} S \cdot L_{(i,x:j)} + S(1-\alpha_{i,k}) \cdot L_{(i,x:i)}} \tag{2}$$

where $L_{(i,x:j)}$ is the distance between x in the i^{th} cell and the j^{th} BS, S is the total power of each BS, $\alpha_{i,k}$ is the normalized power portion of the k^{th} user in the i^{th} cell, BW_T is the total transmission bandwidth and R_b is the data rate. Denote $\hat{I}_{oc(i,x)} = \sum_{j=1}^{N_{oc}} L_{(i,x:j)}/L_{(i,x:i)}$, $(E_b/N_0)_{i,x}$ in (2) is represented by

$$\left(\frac{E_b}{N_0}\right)_{i,x} = \frac{\alpha_{i,k} \cdot \frac{BW_T}{R_b}}{\hat{I}_{oc(i,x)} + (1-\alpha_{i,k})}. \tag{3}$$

The outage probability at the x position in the i^{th} cell is defined by $P_{i,x}^{out} = P\left[\left(\frac{E_b}{N_0}\right)_{i,x} < \gamma\right]$ where γ is a target threshold(E_b/N_0). Then,

$$P_{i,x}^{out} = P\left[\hat{I}_{oc(i,x)} > \alpha_{i,k} \cdot \frac{BW_T}{R_b \gamma} - (1-\alpha_{i,k})\right]. \tag{4}$$

4 SIR-Based Cell Partitioning for Multi-classes

4.1 Mobility Model

Fig. 3 shows an example of region partitioning. The shape of each region has a hexagonal form and the radius from the central point is denoted as r_k for the k^{th} region. The radius difference $r_k - r_{k-1}$ is assumed to be a constant for all $2 \leq k \leq K$.

Let a cell be divided into K regions and are then numbered from the inner region. Fig. 3 shows an example of region partitioning. Here, it is assumed that this system supports J classes. The area of the k^{th} region is given by

$$A_k = (r_k^2 - r_{k-1}^2) \quad \text{and} \quad r_0 = 0. \tag{5}$$

The new call arrival rate the j^{th} class in the k^{th} region is represented by

$$\lambda_{n,j}^k = \Lambda_{a,j} \cdot A_k \tag{6}$$

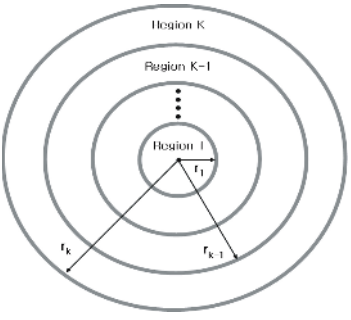


Fig. 3. Cell partitioning into the K regions

where A_k is the area of the k^{th} region in (5) and $\Lambda_{a,j}$ is the new call arrival rate per unit area the j^{th} class. Let $\lambda_{h,j}^k$ be the handoff rate of the j^{th} class in the k^{th} region and be represented by

$$\lambda_{h,j}^k = \frac{1}{\pi} \cdot v_j \cdot \rho_j^k \cdot l_k, \quad l_k = 2\pi \cdot r_k \tag{7}$$

where v_j is the velocity of the MS, ρ_j^k is the traffic density and l_k is the boundary length of the k^{th} region [7]. The density of traffic ρ_j^k can be given by

$$\rho_j^k = \frac{1}{\mu_j^k}(\lambda_{nc,j}^k + \lambda_{hc,j}^k) \cdot \frac{1}{A_k} \tag{8}$$

where $\mu_j^k, \lambda_{nc,j}^k$ and $\lambda_{hc,j}^k$ are the service rate, the successful new call arrival rate and the successful handoff attempt rate of the j^{th} class in the k^{th} region, respectively. In a non-prioritized scheme, the rates can be expressed by

$$\lambda_{nc,j}^k = \lambda_{n,j}^k(1 - P_B) \tag{9}$$

$$\lambda_{hc,j}^k = \lambda_{h,j}^k(1 - P_B) \tag{10}$$

where P_B is the blocking probability, and

$$\frac{1}{\mu_j^k} = \frac{1}{(\mu_{c,j} + \mu_{cell,j}^k)} \tag{11}$$

where $\mu_{c,j}$ and $\mu_{cell,j}^k$ are the channel holding rate and the cell residual rate of the j^{th} class in the k^{th} region. Let T_j^k be the average passing time of the j^{th} class over the k^{th} region under the assumption of that v_j is a constant and unidirectional. The cell residual rate of the j^{th} class in the m^{th} region is then approximated by

$$\frac{1}{\mu_{cell,j}^k} \simeq \frac{1}{\mu_{cell,j}} \cdot \frac{T_j^m}{\sum_{k=1}^K T_j^k}. \tag{12}$$

$\lambda_{h,j}^{k}$ in (7) can be also expressed as the sum of two components:

$$\lambda_{h,j}^{k} = \lambda_{h,j}^{k:k+1} + \lambda_{h,j}^{k:k-1}. \tag{13}$$

In (13), $\lambda_{h,j}^{k:k+1}$ is the handoff attempt rate of the j^{th} class between the k^{th} region and the $(k+1)^{th}$ region, and $\lambda_{h,j}^{k:k-1}$ is also the handoff attempt rate of the j^{th} class between the k^{th} region and the $(k-1)^{th}$ region. Then,

$$\lambda_{h,j}^{k:k+1} = \frac{1}{\pi}\rho_j^k \cdot v_j \cdot l_k \tag{14}$$

$$\lambda_{h,j}^{k:k-1} = \frac{1}{\pi}\rho_j^k \cdot v_j \cdot l_{k-1}. \tag{15}$$

However, (13) is the general form of the handoff rate. At the 1^{th} and the K^{th} regions,

$$\lambda_{h,j}^{1} = \lambda_{h,j}^{1:2} \tag{16}$$

$$\lambda_{h,j}^{K} = \lambda_{h,j}^{intercell} + \lambda_{h,j}^{K:K-1} \tag{17}$$

where $\lambda_{h,j}^{intercell}$ is the handoff attempt rate over the inter-cells.

5 Resource Allocation Schemes

5.1 The Comparison Between the NB and FB Schemes

Fig. 4 (a) shows available states for the NB(No Boundary) scheme with 2 regions and 2 classes. Let $N_{max,j}^{k}$ be the maximum allowable users for the k^{th} region and the j^{th} class. For example, $N_{max,1}^{2}$ ($N_{max,2}^{2}$) means that the number of the maximum admissible users for the $1^{st}(2^{nd})$ class at the 2^{nd} region is 1(2). Let (a, b) be the state where a is the number of users for the 1^{st} class and b is the number of users for the 2^{nd} class. The available states then becomes (0,1), (0,0), (1,0) and (2,0) for the 2^{nd} region. Similarly, the available states in the 1^{st} region can be expressed as shown in Fig. 4 (a). It can be seen that there is no strict

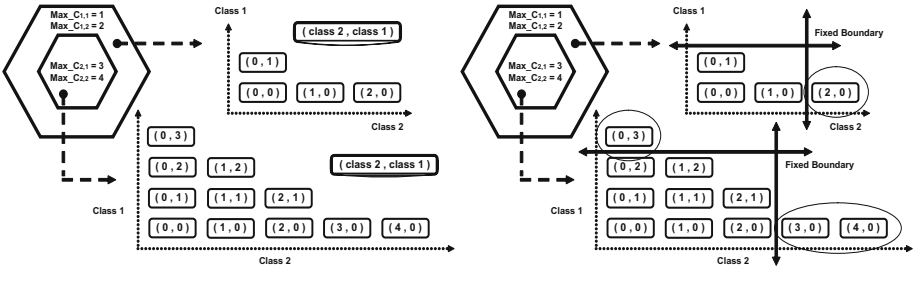


Fig. 4. An example of available states in the NB and FB schemes

resource boundary in the NB scheme. The resource can be shared between the two classes depending on the number of requests.

On the other hand, the available resources in the FB(Fixed Boundary) scheme can be shown in Fig. 4 (b) with the same capacity limits as Fig. 4 (a). In the example, the strict sharing boundary is determined at (1,0) and (0,1) at the outer region, and (2,0) and (0,2) at the inner region. Thus, (2,0) at the outer region is not allowable, and (0,3), (3,0) and (4,0) at the inner region are not allowable due to the capacity boundary violation. There is a tradeoff between the resource management complexity and efficiency. In terms of fairness, the FB scheme is more reliable.

5.2 No Boundary Resource Management Scheme

The steady state probability for the K hexagonal regions and the J classes are derived by using the mobility model defined in the previous section. The state probability for the k^{th} region and the j^{th} class can be written by

$$P^k_{n^k_j,j} = \prod_{q=1}^{n^k_j} \left(\frac{\lambda^k_{nc,j} + \lambda^k_{hc,j}}{q \cdot \mu^k_j}\right) \cdot P^k_{0,j} \tag{18}$$

where n^k_j is the number of the j^{th} class users for the k^{th} region, and $P^k_{0,j}$ is the steady state probability for zero user. Let $N^k_{max,j}$ be the maximum admissible number of users under the NB condition. Using $\sum_{n^k_j=1}^{N^k_{max,j}} P^k_{n^k_j,j} = 1$, (18) can be expressed by

$$P^k_{0,j} = \frac{1}{\sum_{n^k_j=1}^{N^k_{max,j}} \prod_{q=1}^{n^k_j} \left(\frac{\lambda^k_{nc,j} + \lambda^k_{hc,j}}{q \cdot \mu^k_j}\right)}. \tag{19}$$

To represent the number of users at each state, let n^1_1 and n^1_2 (n^2_1 and n^2_2) be the number of the 1^{st} class and the 2^{nd} class users in the 1^{st} region (2^{nd} region). Assume that the number of users at each state is independently distributed. The state probability is then denoted by $P(n^1_1, n^1_2, n^2_1, n^2_2) = P(n^1_1) \cdot P(n^1_2) \cdot P(n^2_1) \cdot P(n^2_2)$. Therefore, the steady state probability over the cell is given by

$$P(a\ state\ over\ K\ regions\ for\ J\ classes) = \prod_{j=1}^{J} \prod_{k=1}^{K} P(n^k_j). \tag{20}$$

The blocking probability for the NB scheme can be then obtained by

$$P^{NB}_B = \sum_{state\ \in U} P(a\ state\ over\ K\ regions\ for\ J\ classes) \tag{21}$$

where U is the blocked state set which consists of states exceeding the capacity constraint by (22)

$$\sum_{k=1}^{K} \sum_{j=1}^{J} (n^k_j \cdot C^k_j) > C_{total} \tag{22}$$

where C_j^k is the capacity for the k^{th} region and the j^{th} class user, and C_{total} is the total capacity.

5.3 Fixed Boundary Resource Management Scheme

In the FB scheme, a fixed boundary exists for each region and each class by

$$FB\ Condition\ :\ \ n_j^k \le N_{fix,j}^k \quad , \quad 0 \le N_{fix,j}^k \le N_{max,j}^k \tag{23}$$

where $N_{fix,j}^k$ is the maximum admissible number of users in the k^{th} region for the FB condition, respectively.

The blocking probability for the FB scheme can be denoted by

$$P_B^{FB} = \sum_{state\ \in U} P(a\ state\ over\ K\ regions\ for\ J\ classes) \tag{24}$$

$$such\ that\ :\ \sum_{k=1}^{K}\sum_{j=1}^{J}(n_j^k \cdot C_j^k) > C_{total} \tag{25}$$

where U is the blocked state set of the NB scheme with $n_j^k \le N_{fix,j}^k$, $0 \le N_{fix,j}^k \le N_{max,j}^k$

6 Simulation Result

For the simulation, the following parameters are used. The modulation method is QPSK (m=2), BW_T/R_b is 128, the target E_b/N_0 is 8 dB, the target outage probability is 0.2, the cell radius is 500 m, and the velocity is $20km/h$. To measure the performance gain accrued from the SIR-based capacity analysis, the number of partitioning regions $K = 4$ and the number of classes $J = 1$ are used. For $Case1$, the total number of no blocking states is 495. On the other hand, for $Case2$, the total number of no blocking states is 340.

Since the channel capacity is rapidly dropped at the outer region of each cell, the number of no blocking states is rapidly decreased at $Case2$ due to its accuracy of the resource estimation. In Table 1, it is shown that the number of common, error and loading states are respectively 165, 330 and 175, respectively. It can be observed that a lot of error states in $Case1$ and loading states in $Case2$ occur due to the non-uniform channel capacity according to the region. Thus, it turns out that the proposed scheme demonstrates a better performance as the non-uniformity of the radio capacity is increased according to the region.

Fig. 5 shows the blocking probabilities for $Case1$ and $Case2$ when $K = 2$ and $J = 1$. Fig. 5 (a) shows the blocking probability measured with $\mu_{c,j}$=8 for all j and Fig. 5 (b) shows the blocking probability measured with $\Lambda_{a,j}$=10 for all j. $Case2$ shows a better performance than $Case1$ as the new call arrival rate is increased and the channel holding rate is decreased. When $\Lambda_{a,j}$ becomes larger, more accurate resource management method is needed to handle more MSs. In

Table 1. Comparison between Divided Cell and Non-divided Cell

	$Case1$	$Case2$
Common State	165	165
Error State	330	x
Loading State	x	175

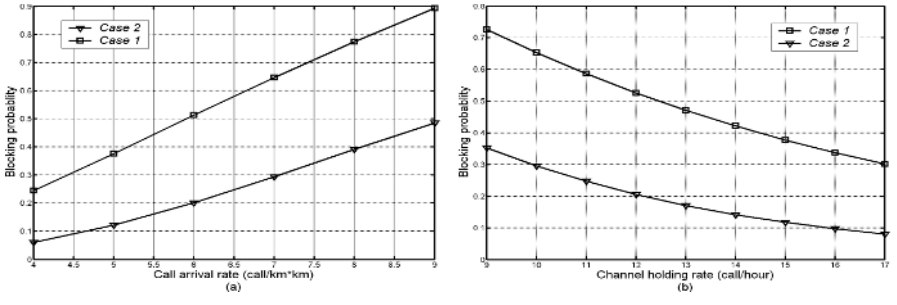


Fig. 5. The blocking probability according to the new call arrival rate in (a) and the channel holding rate in (b)

addition, as the channel holding rate is increased, the crossover rate over the regions is decreased, which leads to decrease the blocking probability.

The resource reservation scheme accommodates multi-class services using two difference resource allocation schemes, i.e., the NB and FB schemes. Here, J is set to 2 and the blocking probability is measured. Fig. 5 (a) shows the blocking probability according to the new cell arrival rate $\Lambda_{a,j}$ when $\mu_{c,j}$=30. It is shown that the difference of the blocking probability becomes larger as $\mu_{c,j}$ is increased. Here, the statistical behavior of the traffic for each class is assumed to be the same, so the performance gain by the NB is not large. If the input traffic patterns are more heterogeneous, more additional performance gain should be obtained.

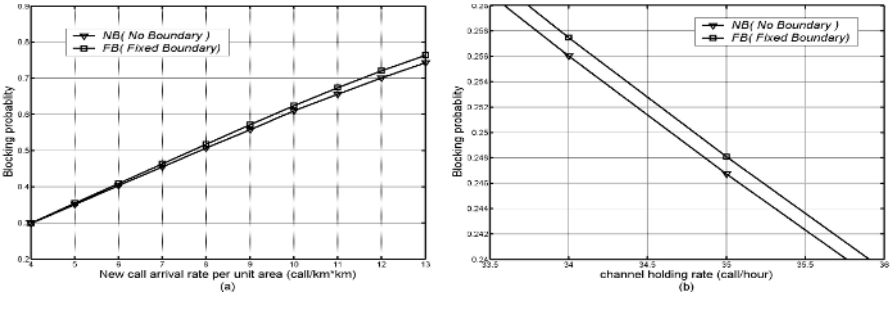


Fig. 6. Comparison between the NB and FB shcemes

In Fig. 6 (b), the blocking probability according to the channel holding rate is demonstrated with $\Lambda_{a,j}= 5$. It is shown that the blocking probability of the NB scheme is lower than that of the FB scheme along the channel holding rate.

7 Conclusion

When a frequency reuse factor of 1 is employed to improve channel throughput for OFDM-based broadband networks, a severe ICI can occur at the cell boundary, which results in a nonuniform channel capacity according to the locaiton. Thus, it is necessary to develop several strategies needed to overcome such problems caused by the ICI. This paper presents a resource reservation scheme for maintaining a QoS over the channel holding time duration. The location-based channel resource is measured in advance and reserved an additional resource needed to absorb the radio resource dynamics. To prove the effectiveness of the algorithm, a closed form of blocking probability according to two resource management schemes associated with multi-class services is derived. In the simulation results, it is proved that the proposed scheme demonstrates a better performance as the non-uniformity of the radio capacity is increased according to the region.

References

1. D. K. Kim and D. K. Sung, "Traffic management in a multicode CDMA system supporting soft handoffs," *Vehicular Technology, IEEE Trans.*, vol. 51,no 1, pp. 52-62, Jan., 2002.
2. Evans, J.S., Everitt, D, "Effective bandwidth-based admission control for multiservice CDMA cellular networks," *IEEE Trans.*, vol. 48,no 1, pp. 36-46, Jan., 1999.
3. Anding Zhu, Jiandong Hu, "Adaptive call admission control for multi-class CDMA cellular systems," *Communications Conference, IEEE*, vol. 1, pp. 533-536, Oct.,1999
4. Ishikawa, Y., Umeda, N, "Capacity design and performance of call admission control in cellular CDMA systems," *Selected Areas in Communications, IEEE Journal*, vol. 15,no 8, pp. 1627 - 1635, Oct.,1997
5. Dongmei Zhao, Xuemin Shen, Mark, J.W, "Efficient call admission control for heterogeneous services in wireless mobile ATM networks," *Communications Magazine, IEEE*, vol. 38,no 10, pp. 72-78, Oct.,2000
6. Hyukmin Son and Sanghoon Lee, "Forward-link capacity analysis for MC-CDMA," *IEICE Trans. On Comm. Letter*, Oct.,2005
7. K. S. Meiler-Hellstern and E. Alonso, "The use of SS7 and GSM to support high density personal communications," *in Proc, ICC*, pp. 1698-1702, Oct.,1992
8. J. H. Wen and J. W. Wang, "A non-collision PRMA protocol for integrated voice and data wireless networks," *Universal Personal Comm. 1995 Fourth IEEE International Conference*, pp. 462-466, Nov.,1995
9. Michael Cheung and Jon W. Mark, "Resource Allocation for Handling Two QoS Classes at a Generic Radio Cell," *GLOBECOM 01. IEEE*, pp. 2617-2621, Nov.,2001
10. R.F. Chang and S.W. Wang, "QOS-Based Call Admission for Intergrated Voice and DATA in CDMA Systems," *VTC IEEE*, pp. 623-627,1996

Efficient Wireless Resource Management Scheme Using Differential Received Signal Strength Indicator in Soft Handoff

YoungHwan Kwon[1], Seong Gon Choi[2,⋆], Jun Kyun Choi[1], Jeong Yun Kim[3], and Jin Ho Hahm[3]

[1] Information and Communications University (ICU)
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
{yhkwon, jkchoi}@icu.ac.kr
[2] ChungBuk National University (CBNU)
12, Gaeshin-dong, Heungduk-gu, Chungbuk, Korea
sgchoi@cbnu.ac.kr
[3] Electronics and Telecommunications Research Institute (ETRI)
12, Gajeong-Dong, Yuseong-gu, Daejeon, Korea
{jykim, jhhahm}@etri.re.kr

Abstract. This paper proposes an efficient wireless resource management scheme using a Differential Received Signal Strength Indicator (DRSSI) in soft handoff. DRSSI is the differential value of RSSIs, and can be used to monitor the movement of a Mobile Node (MN). For example, the DRSSI has a bigger value or changes its sign (+, -) if the MN changes its movement speed or direction. A base station can determine the priority of a MN and can acknowledge the movement direction of a MN with these features. By using the priority and movement direction of a MN, we can reduce the handoff blocking probability by reducing the unnecessary usage of wireless resource, which can be analyzed by numerical analysis.

1 Introduction

Developments in wireless communications have resulted in an increase in the number of users and the need for more resource capacity. As such, there have been many studies on ways to handle the wireless resource efficiently in order to improve the wireless system. The most general mechanism to increase the capacity of a wireless resource is through the use of cellular architecture. This mechanism uses the wireless resource repeatedly in each different cell. However, the disadvantage is that when a Mobile Node (MN) moves in the cellular environment, it is possible that it can move out of cell range and go into another cell [4], [5].

Thus, a handoff mechanism is needed to support connection between cells. This handoff mechanism significantly affects the performance of mobile service in handoff blocking. A soft handoff mechanism is proposed to provide an efficient

⋆ Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 52–61, 2006.

handoff. It maintains a connection continuously in handoff by receiving signals from cells in the overlapped region between two cells [2].

This paper proposes to manage the wireless resource of a cell efficiently in soft handoff when the resource of the Base Station (BS) in a cell is not enough to support a new MN. In the mobile environment, a handoff request is blocked when there is no available bandwidth in a cell. However, if the BS could allocate the resource of a slow MN to a fast MN during the time the slow MN stays in the overlapped region of soft handoff, the BS could deal with the new handoff request using limited resource.

There are times when a MN changes its movement direction and the BS does not need to assign a wireless resource continuously. If the BS was able to release the resource when a MN changes its movement direction to other cells, these two mechanisms could reduce the usage of unnecessary resource.

To support these ideas, we use the Differential Received Signal Strength (DRSSI) of a MN. DRSSI is the differential value between RSSIs and has two properties. DRSSI has a different variation depending on the speed of a MN. If a MN moves faster, it has more variation. We use DRSSI to classify the priority of a MN with variation of DRSSI. A fast MN is a high priority user and a slow MN is a low priority user. A high priority user could request the reserved resource of a low priority user in the overlapped region.

The DRSSI then changes its sign (+, -) when a MN changes its movement direction. If the BS could monitor the movement of a MN in the overlapped region of soft handoff, the BS could acknowledge its movement and release the resource of MN instantly. We use queuing analysis to evaluate the efficiency of the proposed algorithm through retrieving handoff blocking probability. From this modeling and numerical analysis, we show that our algorithm has better handoff blocking probability.

The remainder of the paper is organized as follows. In Section 2, we explain various soft handoff mechanisms. In Section 3, we propose our efficient wireless resource management scheme using DRSSI. We then analyze numerically the handoff blocking probability of the proposed algorithm in Section 4, and draw our conclusions in Section 5.

2 Soft Handoff Mechanism

The handoff procedure is the most important in mobile communications because it provides mobility for a MN and has a serious affect on service quality. In this section, we describe handoff classification method and various soft handoff mechanisms using various parameters to decide handoff.

Handoff mechanisms are divided into hard handoff mechanism and soft handoff mechanism according to connection maintenance in the handoff. A hard handoff mechanism establishes a new connection with a new cell after disconnecting with a previous cell. If a new connection is not established with a new cell instantly after the previous connection is disconnected, data is lost and suffers a big delay due to temporary loss of connection [1][2].

Soft handoff can receive signals concurrently from a previous cell and a new cell and can select the better signal of cell. Therefore, it has low data loss probability because it always maintains connection. However, this mechanism is not efficient in the use of wireless resource because it maintains more than one connection.

To overcome this problem, there have been many researches on an efficient handoff algorithm to manage several signals received by a MN [2]. Several parameters are used to manage wireless resource efficiently in the soft handoff mechanism. To manage wireless resource efficiently, we can decide the handoff timing-such as handoff start time and end time in soft handoff-and reduce the number of handoff by reducing unnecessary handoff.

The most basic parameter to decide handoff timing is RSSI, which is a receiving signal power of wireless radio. In the soft handoff mechanism, RSSI is used to select a bigger signal of BS in several cells and cuts off BSs with lower signals [3]. A wireless resource can be managed more efficiently by calculating more accurate handoff timing. Therefore, many parameters such as CIR (carrier-to-interference ratio), SIR (signal-to-interference ratio), BER (bit error ratio) and BLER (block error ratio) are used to decide accurate handoff timing [1].

Wireless resource is limited in mobile communications. By reducing the size of cell, the same wireless radio resource could be reused repeatedly in more cells. However, this mechanism has a major disadvantage in its large number of handoff procedures. Handoff procedures increase network burden to deal with a handoff and decrease the efficiency of resource usage due to more than one resource per data. Reducing the number of handoffs makes the handoff mechanism more efficient [4], [5].

A handoff mechanism starts the handoff procedure when a new BS signal is bigger than the absolute threshold value or the sum of the predefined margin value and the average previous signal value. This mechanism can reduce the number of handoff procedures by adding margin [6]. The adaptive prediction based handoff mechanism could use predicted RSSI to reduce unnecessary handoffs. The predicted RSSI could use the correlation features of signals and then reduce the system load [7]. Various handoff priority schemes reduce the drop-out of existing calls by giving preference when assigning wireless resource because the existing calls affect service quality [8].

3 Proposed Wireless Resource Management Scheme Using Differential RSSI in Soft Handoff

Our proposed algorithm using DRSSI uses wireless resource efficiently in handoff procedure by assigning a resource based on the priority of a MN and by releasing unnecessary bandwidth as soon as possible. First, we consider the operation environment of the proposed algorithm and later describe DRSSI.

3.1 Operation Environment of the Proposed Algorithm

To support our algorithm, we define the state of resource of a BS and a MN. Firstly, the resource of a BS is divided into active bandwidth and reserved

bandwidth. Active bandwidth is occupied by a MN and reserved bandwidth is reserved by a MN in the overlapped region during soft handoff. A reserved bandwidth could be assigned temporarily to another node before it changes to an active bandwidth and could be released if it is unnecessary.

Secondly, a MN is classified as a high priority user or a low priority user. A high priority user has a higher priority to use a reserved bandwidth than a low priority user. The state of a MN is dependent on its handoff time because a high priority user should complete the usage of a reserved bandwidth before a low priority user uses it. The handoff time is dependent on the movement speed of a MN [9].

Therefore, we know that a high priority user should be a fast MN that is fast enough to go out of a cell before a low priority user escapes the overlapped region. A slow MN could be a low priority user of reserved bandwidth because it should stay in the overlapped region until a high priority user goes out of a cell. In our algorithm, we use DRSSI to determine the priority of a MN.

Additionally, a MN changes its movement direction to other cells during handoff. In this case, its reserved bandwidth is not needed because the reserved bandwidth is not used. If a BS could predict the movement direction of MN, we could reduce the wasted time of reserved bandwidth. In our algorithm, we use DRSSI to monitor the movement of a MN.

This scheme could be applied when most of the BS resource is assigned to MNs and there is no available resource except for a reserved bandwidth. In this case, if the reserved bandwidth could be assigned to a high priority user temporarily and released by the BS when it is not needed, it could be used more efficiently for other MNs.

DRSSI is used to monitor the movement speed and direction of a MN in our algorithm at a BS. In doing so, the BS can determine the priority of a MN and release the reserved bandwidth of the MN that changes its direction. This concept is newly proposed and examined in the next section in detail.

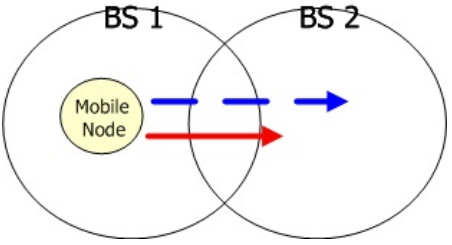
3.2 Differential Received Signal Strength Indicator (DRSSI)

DRSSI is the most important parameter in this paper. It is used to determine the priority of a MN and to decide the movement direction of a MN by monitoring its movement in the overlapped region of soft handoff. DRSSI is the differential value between a current measured value and a previous measured value that is measured by a BS periodically.

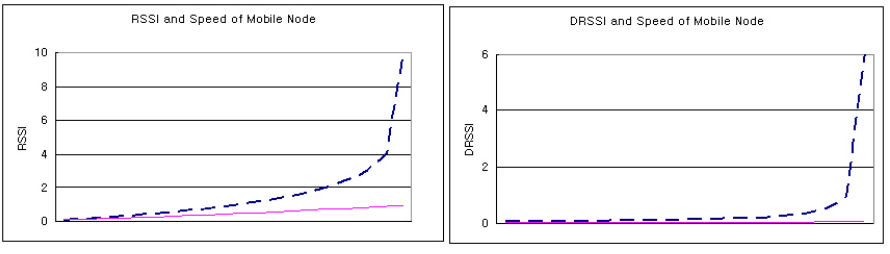
$$DRSSI_i = RSSI_i - RSSI_{i-1} \tag{1}$$

We describe the two properties of DRSSI in this section.

The first property of DRSSI is that it has a different variation depending on the speed of MNs. For example, there are two nodes that approach a BS, one is a fast MN and the other is a slow MN. In the same amount of time, they move a different distance. The RSSI variation between the two MNs is different according to the propagation characteristic of wireless radio.



(a) The movement of Mobile Nodes.



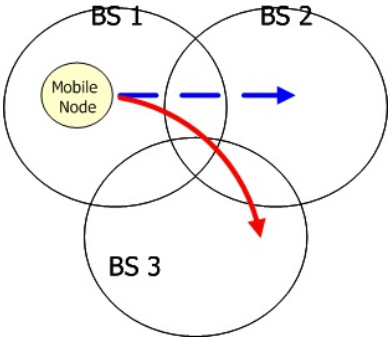
(b) RSSI and DRSSI with moving speed of Mobile Node.

Fig. 1. Relation between DRSSI and the moving speed of Mobile Node

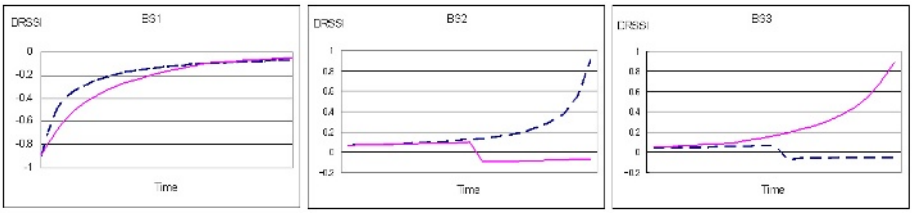
A MN is the faster, DRSSI the bigger. Figure 1 shows this feature. Figure 1 (a) shows the movement of two MNs. The MN along the blue dashed line is a fast MN and the MN along the red solid line is a slow MN. Figure 1 (b) shows the variation of RSSI and DRSSI according to time and increasing rate. A fast MN has larger DRSSI value because the RSSI variation is dependent on distance [9]. From this property, we can know which MN is faster. Therefore, by using this property of DRSSI, we can assign high priority to a fast MN and low priority to a slow MN.

Another property of DRSSI is that it changes its signs (+,-) if a MN changes its movement direction. When a MN approaches a BS, its RSSI increases and the DRSSI is a positive value and also increases. However, when a MN goes far away from a BS, the RSSI value decreases gradually and the DRSSI is a negative value and converges to zero. Therefore, if a MN grazes the cell barrier, the DRSSI of a MN is a positive value firstly, but the DRSSI of MN changes to a negative value after a MN changes its movement direction to another cell. Figure 2 describes the differing DRSSIs in each cell according to the movement of MNs.

In BS 1, the DRSSIs of two MNs become a negative value and converge to zero by going away from BS 1. In BS 2, the MN along the blue dashed line goes to BS 2 and its DRSSI gets bigger and bigger. However, the MN along the red solid line grazes BS 2 and its DRSSI increases firstly, and then changes to a negative value after changing its movement direction. Lastly, the DRSSI of the MN along the red line increases with a positive value in BS 3 by approaching BS 3. By using the DRSSI of a MN, the movement of a MN can be predicted. When a BS knows the movement of a MN with the DRSSI of a MN as in BS



(a) The movement of Mobile Nodes.



(b) DRSSI values of Mobile Node in each cell.

Fig. 2. Relationship between DRSSI and the movement of a Mobile Node

2, the BS is able to release the resource of the MN instantly. By doing so, the resource waste of the BS is reduced and the released resource assigned to other MNs.

4 Numerical Analysis

In this section, we evaluate our efficient wireless resource management scheme using the DRSSI in soft handoff. We use an M/M/k/k queuing model to analyze the handoff blocking probability of our algorithm and graph our results.

4.1 Analysis of the Proposed Algorithm

We assume that a MN carries only one connection. This means that a MN is not allowed any bulk arrival. Another assumption is that the maximum bandwidth of a BS is B number of Basic Bandwidth Unit (BBU) in a cell. This BBU is a connection of a BS and is not shared by MNs.

When a connection arrival process is a poison process with arrival rate λ, this arrival rate consists of new connection arrival rate λ_n and handoff connection arrival λ_h. New connection is the generated connection within a cell. Handoff connection is a handoff-in connection. It is a summation between the handoff arrival rate of a high priority user (λ_{h1}) and the handoff arrival rate of a low priority user (λ_{h2}).

Handoff arrival rate is dependent on how long a MN stays in a cell before handoff. This is referred to as the cell residence time of MN μ_h. It is an exponential distribution with average value $1/\mu_h$. And, the resource occupancy time $1/\mu$ in a cell is defined as the duration between when a resource is occupied due to a new connection arrival or handoff connection arrival and when it is released due to completion of the connection or handoff-out of connection. Therefore, resource occupancy time depends on cell residence time and connection holding time that is an exponential distribution with average rate $1/\mu_c$.

In Section 3, we divide the state of resource into an active bandwidth and a reserved bandwidth. Therefore, the state of a BS is dependent on these two types of resource. It is defined by (n, m), where "n" is the number of active bandwidth and "m" is the number of reserved bandwidth.

We analyze the behavior of a MN based on its priority and the behavior of a MN based on its movement direction. However, it is very difficult to analyze the behavior of MNs. Therefore, we assume these behaviors to simplify the analysis with two parameters, λ_f and λ_d. The first one is for the high priority user to use the reserved band-width of the low priority user. The second one is for the MN to change its movement direction.

$$\lambda_f = HP * \lambda_{h1} \tag{2}$$

$$\lambda_d = CP * \lambda_h \tag{3}$$

λ_f: The arrival rate of a high priority user that is fast enough to use a reserved bandwidth.
λ_d: The direction change rate of a MN in the overlapped region of soft handoff.
HP: The ratio of a high priority user that is fast enough to use a reserved bandwidth.
CP: The ratio of moving direction change in handoff.

Therefore, the state probability of the BS is $P_{(n,m)}$. The resource transition matrix of our algorithm is in Figure 3. Our algorithm is applied when active bandwidths are occupied B-1 and B. When most of the resource of the BS is occupied by a MN, our algorithm is able to support one more high priority user. We can see that the state probability is divided into two cases: one is the case where the occupied bandwidth is from 0 to B-2; and the other is more than B-1. State probability P_{B-1} and P_B are defined like the following equation from (4) to (5).

$$P_{B-1} = P_{(B-1,0)} + P_{(B-1,1)} \tag{4}$$

$$P_B = P_{(B,0)} + P_{(B,1)} \tag{5}$$

With equations (4) and (5), we are able to calculate the other remained state probabilities.

$$P_{(B-1,0)} = \frac{\mu^{B-1}}{(B-1)!\mu^{B-1}} P_0 \tag{6}$$

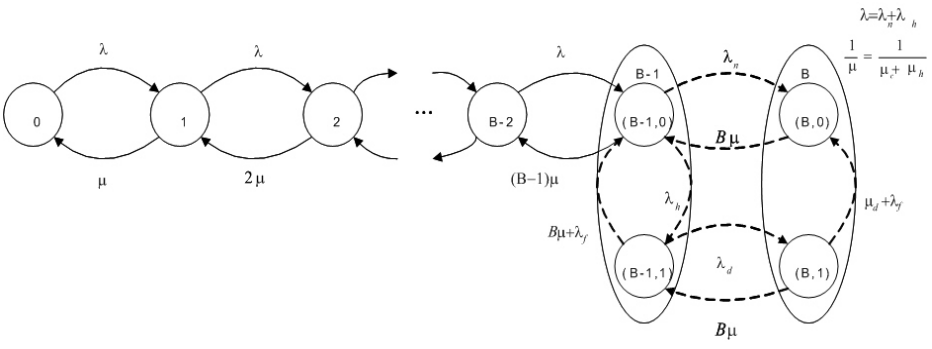


Fig. 3. State transition matrix of a base station in our proposed algorithm

$$P_{(B-1,1)} = \frac{\lambda_h}{(B\mu + \lambda_d + \lambda_f) - \frac{B\mu\lambda_f}{B\mu+\mu_f+\lambda_d}} P_{(B-1,0)} \tag{7}$$

$$P_{(B,0)} = \{\frac{\lambda}{B\mu} - \frac{(B\mu + \lambda_d)\lambda_h}{(B\mu + \lambda_d + \lambda_f) - \frac{B\mu\lambda_f}{B\mu+\mu_f+\lambda_d}}\} P_{(B-1,0)} \tag{8}$$

$$P_{(B,1)} = \frac{\lambda_f\lambda_h}{(B\mu + \mu_f + \lambda_d)(B\mu + \lambda_d + \lambda_f)} P_{(B-1,0)} \tag{9}$$

Equations (6) to (11) can be calculated after P_0 is known. It can be calculated with the property of $\sum_{n=0}^{B} P_n = 1$. Then, all state probabilities can be calculated by inserting P_0 in each equation.

4.2 Numerical Result

We evaluate our proposed algorithm in terms of handoff blocking probability from the state probability of Figure 3. The handoff blocking probability is an important parameter to evaluate the performance of a MN because the handoff blocking of a MN affects the performance of mobile service more seriously than the blocking of a connection request at first.

Handoff blocking means that connection could not be maintained continuously during handoff. We divide handoff blocking into high priority user handoff blocking probability (P_{HBH}) and low priority user handoff blocking probability (P_{HBL}).

$$P_{HBH} = P_B \tag{10}$$

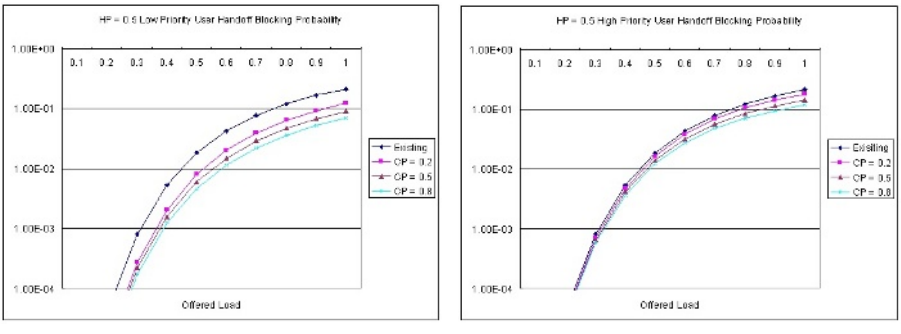
$$P_{HBL} = P_{(B,0)} + P_{(B-1,1)} \tag{11}$$

The handoff request of a high priority user is blocked when B number of active bandwidth is used by a MN. The handoff request of a low priority user is blocked when B number of active bandwidth is used by a MN or B-1 active bandwidths

are used and 1 reserved bandwidth is reserved. We need the following parameters to get the result as a graph according to queuing analysis.

B=10, $1/\mu_c$=100s, $1/\mu_c$=500s, CP=0.2, 0.5, 0.8, HP=0.5

By using the equations from (4) to (9) and by inserting these parameter values, we get the blocking probability graphs of our proposed algorithm and compare them with the existing algorithm. These graphs are shown in Figure 4. Figure 4 shows the case when 50% of high priority users satisfy the condition of our algorithm and can use the reserved bandwidth of low priority users temporarily. If the movement change ratio of a MN is the higher, the handoff blocking probability is the lower.



(a) Low priority user (b) High priority user

Fig. 4. Handoff Blocking Probability

5 Conclusion

We proposes an efficient wireless resource management scheme using DRSSI in soft handoff by reducing the unnecessary usage of wireless resource. Our proposed algorithm considers the states of BS and MN. The state of a BS is dependent on active bandwidth and reserved bandwidth. The state of a MN is dependent on the speed of the MN and its movement direction.

To know these states of a MN, a BS uses the DRSSI of a MN. DRSSI has a different variation according to the speed of a MN and changes its signs due to the movement direction of a MN. By monitoring the DRSSI of a MN in handoff, a BS can classify a MN into high priority user and low priority user. Based on this priority, a high priority user can request the reserved bandwidth of a low priority user in handoff and use the reserved bandwidth when it is fast enough. And, the BS can release the reserved bandwidth of a grazing MN when a MN goes far away from the BS.

By doing so, we could use wireless resource more efficiently because our algorithm improves the availability of resource. In Section 4, we prove our algorithm

by numerical analysis and show that our algorithm reduces the handoff blocking probability of a MN.

Acknowledgement

This work was supported in part by the Institute of Information Technology Assessment (IITA) through the Ministry of Information and Communication (MIC) and the Korea Science and Engineering Foundation (KOSEF) through the Ministry of Science and Technology (MOST), Korea.

References

1. Kaven Pahlavan, Prashant Krishnamurthy, et al: "Handoff in Hybrid Mobile Data Networks", IEEE Personal Communications, April (2000) 34-47
2. Yi-Bing Lin, Ai-Chun Pang: "Comparing soft and hard handoffs", IEEE Transactions on Vehicular Technology, Vol. 49, Issues 3, May (2000) 792-798
3. Ken-Ichi Itoh, et al.: "Performance of Handoff Algorithm Based on Distance and RSSI Measurements", IEEE Transactions on Vehicular Technology, Vol. 51, No. 6, November (2002)
4. Gregory P. Pollini: "Trends in Handover Design", IEEE Communications Magazine, March (1996) 82-90
5. Mika Gudmundson: "Analysis of Handover Algorithm", Vehicular Technology Conference 1991, May (1991) 537-542
6. Zhang, N., Holtzman, J. M., "Analysis of handoff algorithms using both absolute and rela-tive measurements", IEEE Transactions on Volume 45, Issue 1, Feburary (1996) 174-179
7. V. Kapoor, G. Edwards, R. Sankar: "Handoff Criteria for Personal Communication Net-works", SUPERCOMM/ICC '94, May (1994) 1297-1301
8. Gamini N. Senarath and David Everitt: "Performance of Handover Priority and Queueing Systems under Different Handover Request Strategies for Microcellular Mobile Communi-cation Systems", Vehicular Technology Conference 1995, vol. 2, July (1995) 897-901
9. Seong Gon Choi, Ok Sik Yang, Jun Kyun Choi, "An efficient resource allocation scheme during handoff in mobile wireless networks" IEEE/ICACT 2005, Feburary (2005)

Performance Evaluation of Public Key Based Mechanisms for Mobile IPv4 Authentication in AAA Environments*

Jung-Muk Lim, Hyung-Jin Lim, and Tai-Myoung Chung

Internet Management Technology Laboratory,
Scool of Information and Communication Engineering,
SungKyunKwan University, Korea
{jmlim, hjlim, tmchung}@imtl.skku.ac.kr

Abstract. With the proliferation of mobile terminals, use of the Internet in mobile environments is becoming more common. In order to support mobility in these terminals, Mobile IPv4 was proposed, representing the standard in IPv4 environments. In this environment, authentication should be mandatory, because mobile terminals can utilize Internet services in all foreign domains. Mobile IPv4 provides symmetric key based authentication using the default HMAC-MD5. However, symmetric key based authentication creates a problem, when it comes to key distribution. In order to solve this problem, public key based authentication mechanisms were proposed. In this paper, the performance of each of these mechanisms is evaluated. The results demonstrate that, among these mechanisms, partial certificate based authentication results in superior performance, and certificate based authentication results in the worst performance. This paper creates the possibility of public key based authentication mechanisms being used for future mobile terminal authentication, although current public key based authentication mechanisms result in lower performance than symmetric key based authentication.

1 Introduction

Semiconductor and telecommunications technology has evolved steadily, the size of computers is continuously being reduced, and communications is progressing from wired environments to the wireless environments. These trends represent the foundation of mobile computers and new forms of communication. Therefore, it is natural for mobile terminals to continuously utilize Internet services, while in motion, and at any location. The current Internet network protocol standard, IPv4, does not support mobility for mobile terminals. Thus, Mobile IPv4 as an IPv4 extension must be implemented to support mobility.

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 62–71, 2006.

Within a mobile terminal, a user may request Internet services from a foreign domain instead of a home domain of which the user is registered. Therefore, terminal authentication is required, unlike wired networks in which a user is only connected to his/her domain. Authentication is classified into terminal authentication, granted by a service agent and service agent authentication granted by a terminal. The former represents preprocessing for authorization and accounting, and the latter represents processing for preventing attackers from masquerading as service agents.This mechanism is required for mobile terminals to interact with any other domain, because the terminals can request Internet services in any foreign domain. This is Authentication, Authorization, and Accounting (AAA), a framework to manage authentication, authorization, and accounting comprehensively. Consequently, Mobility of mobile terminals requires an AAA infrastructure.

Mobile IPv4 provides authentication, using HMAC-MD5 by default. However, this method suffers from the key distribution problem in which secret keys must be distributed in advance, due to the requirements of symmetric cryptography. Although a key may be distributed between a Mobile Node (MN) and a corresponding Home agent (HA), it is almost impossible for a key to be distributed between Foreign Agents (FAs) and MN, or between FAs and HA. Furthermore, performance is reduced be-tween domains. To solve this key distribution problem, certificated based authentication was proposed [4].

However, public key based mechanisms cannot be applied directly to mobile environments because this application results in noticeably slower operation over symmetric key based mechanisms. In addition, it suffers from the problem that mobile terminals do not have sufficient memory for certificates. To solve this public key based mechanism problem, a partial certificate based authentication mechanism was pro-posed [5]. The public key is used only between a FA and the HA, as both have high computation power. An identity based authentication mechanism was also proposed [6]. This mechanism does not require a certificate based infrastructure.

2 Mobile IPv4 Authentication Mechanisms

In this section, Mobile IPv4 authentication mechanisms are described. These mechanisms consist of default authentication, certificate based authentication, partial certificate based authentication, and identity based authentication.

2.1 Default Authentication

In order to use HMAC-MD5, the Mobile IPv4 default authentication mechanism requires that a Security Association (SA) between a MN and HA must be established in advance [1]. The registration process using default authentication is presented in [Figure 1].

The RRQ consists of M_1 and $< M_1 > K_{MN-HA}$. The M_1 is the RRQ's body including the MN's nonce and HA's previous nonce within the identification field. The $< M_1 > K_{MN-HA}$ is the Message Authentication Code (MAC) of the M_1 using HMAC-MD5 and a previously shared 128 bit secret key. The RRQ is

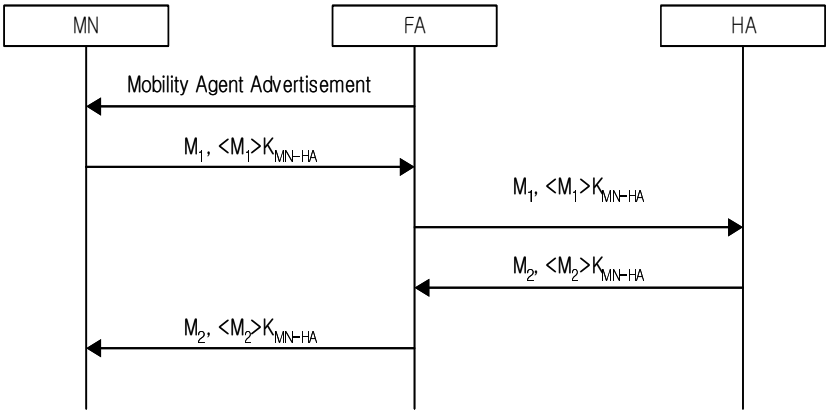


Fig. 1. Registration Process using Default Authentication

forwarded to the HA through the FA. The HA confirms whether its nonce in the RRQ is identical to the nonce previously sent to the MN. If they are not identical each other, the HA returns an error code in the RRP. If they are identical, the HA verifies the MAC. If the MAC is incorrect, the HA transmits an error code in the RRP to the MN, through the FA. If the MAC is correct, the HA updates its binding information and transmits a success code in the RRP to the MN, through the FA. Herein, the authentication between the FA and HA is omitted but authentication be-tween the FA and HA must be achieved if accounting is to be considered.

The RRP consists of M_2 and $< M_2 >K_{MN-HA}$. The M_2 is the RRP's body including the MN's nonce and HA's nonce within the identification field. The MN's nonce was in the RRQ and HA's nonce will be used for the next registration by the MN. The $< M_2 >K_{MN-HA}$ is the MAC of the M2 using HMAC-MD5 and the previously shared 128 bit secret key. Herein, the authentication between the FA and HA is omitted but authentication be-tween the FA and HA must be achieved if accounting is to be considered.

Default authentication assumes that previously shared secret keys exist between MN and HA, between MN and FA, and between FA and HA. It is slightly cumber-some, in that a MN and HA share a secret key between them in advance. Furthermore, it is almost impossible for a MN and FA, or a FA and HA, to share a secret key be-tween them in advance. To solve this problem, another mechanism is required. The requirement that secret key must be distributed in advance, can be solved by distributing keys dynamically. However, this solution is not suitable because of excessive overhead. Alternatively, this problem can be solved using public key cryptography.

2.2 Certificate Based Authentication

In order to solve the problem of the Mobile IPv4 default authentication mechanism being based on symmetric key, a public key based authentication

mechanism was proposed [2][4]. This mechanism, which has a different basis to the Mobile IP default authentication mechanism, solves the key distribution problem by transmitting certificates, which include the public key, in the registration process. A registration process, using certificate based authentication, is presented in [Figure 2].

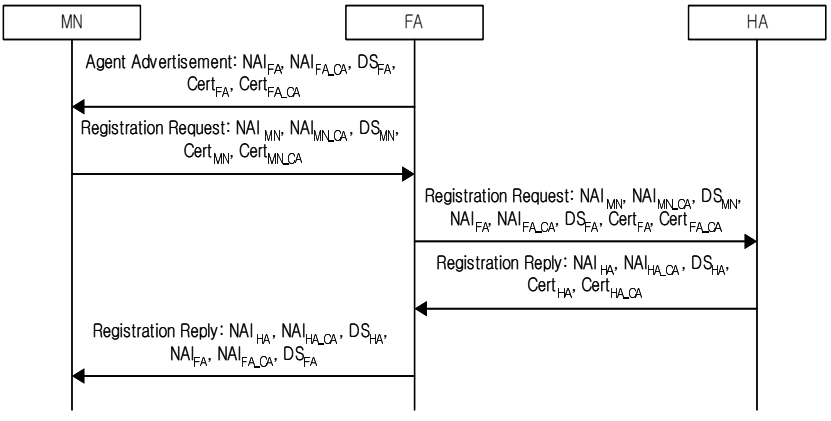


Fig. 2. Registration Process using Certificate based Authentication

The FA transmits an Agent Advertisement message including its NAI, CA's NAI, signature, certificate, and CA's certificate to the MN. The MN authenticates the Agent Advertisement message by verifying the FA's signature using the FA's certificate and the CA's certificate of the FA. The MN transmits a RRQ, including its NAI, CA's NAI, signature, certificate, and CA's certificate to the FA. The FA authenticates the MN by verifying the MN's signa-ture using the MN's certificate and the CA's certificate of the MN.

The FA transmits the RRQ including the MN's NAI, MN's CA's NAI, MN's signature, its NAI, CA's NAI, certificate, and CA's certificate to the HA. The HA authenticates the MN by verifying the MN's signature using the MN's certificate and the CA's certificate of the MN, which are shared in advance. It also authenticates the FA by verifying the FA's signature using the FA's certificate and CA's certificate of the FA, in the RRQ received from the FA. The HA transmits a RRP including its NAI, CA's NAI, signature, certificate, and CA's certificate to the FA. The FA authenticates the HA by verifying the HA's signature using the HA's certificate and the CA's certificate of the HA.

The FA transmits the RRP including HA's NAI, the CA's NAI of the HA, HA's certificate, its NAI, and CA's NAI to the MN. The MN authenticates the HA by verifying the HA's signature using the HA's certificate and the CA's certificate of the HA which are shared in advance. It also authenticates the FA by verifying the FA's signature using the FA's certificate and the CA's certificate of the FA.

In these flows, mutual authentication between the MN and FA, between the MN and HA, and between the FA and HA are achieved. However, public key based authentication requires much more computation than symmetric key based authentication. Thus, it is not suitable to use in devices, which have low computation power such as mobile terminals. Furthermore, it has another problem where a MN must store certificates, despite the limited memory space of the MN.

2.3 Partial Certificate Based Authentication

Instead of protecting the entire registration process, a mechanism was proposed where certificate based authentication is used only in places where the MN does not require processing of the public key algorithm and does not require storage of the certificate [5]. The registration process using partial certificate based authentication is presented in [Figure 3].

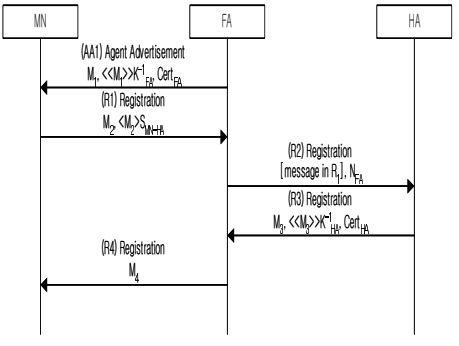


Fig. 3. Registration Process using Partial Certificate based Authentication

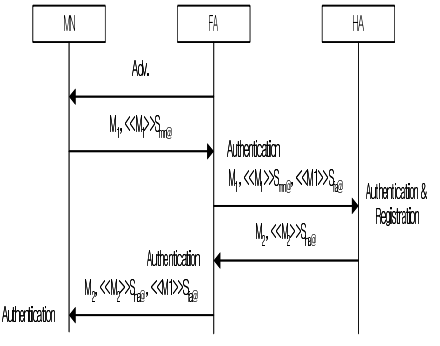


Fig. 4. Registration Process using Identity based Authentication

The FA transmits an Agent Advertisement including M_1 and signature of the M_1, its certificate to the MN. The M_1 includes its id and the MN's CoA. Without any authentication process to the FA, the MN transmits a RRQ including the FA's id, HA's id, its home address, its CoA, previous HA's nonce, its nonce, M_2, and the MAC of the M_2 using the secret key shared with the HA to the FA. M_2 represents the Agent Advertisement message received from the FA. The FA appends its nonce to the RRQ received from the MN and transmits it to the HA. The HA prevents a malicious person from deploying a replay attack, by confirming its previous nonce. It authenticates the FA by verifying the FA's signature using the FA's certificate, and authenticates the MN by verifying the MN's MAC, using the previously shared secret key. Thus, the FA and MN are authenticated by the HA.

The HA transmits a RRP, which includes M_3, signature of the M_3, and its certificate to the FA. The M_3 includes M_4, MAC of the M_4 using the secret key,

which is shared with the MN, and FA's nonce. The M_4 includes the FA's id, its id, the MN's home address, its next nonce, and the MN's nonce. The FA prevents malicious individuals from deploying a replay attack by confirming its nonce in the RRP received from the HA. It authenticates the HA by verifying the HA's signature using the HA's certificate and authenticates the MN by confirming the registration result in the RRP received from the HA. The FA transmits the M_4 to the MN. The MN authenticates the HA by verifying the HA's MAC using the secret key which is shared with the HA, and authenticates the FA by confirming the registration result in the RRP received from the FA. Thus, the MN and HA are authenticated by the FA, and the FA and HA are authenticated by the MN.

This mechanism requires that a MN and HA must have a previously shared secret key and a public key infrastructure must exist for the FA and HA.

2.4 Identity Based Authentication

To solve the problem of storing certificates by the MN, and reducing network over-head by transmitting certificates, identity based authentication was proposed [3][6]. The registration process using identity based authentication is presented in [Figure 4].

The MN receives an Agent Advertisement message from the FA and then transmits M1, which is the RRQ's body, and its signature of the M_1 to the FA. The FA authenticates the MN by verifying the MN's signature using the MN's identity, appends its signature to the RRQ, and then transmits it. The HA authenticates the MN by verify-ing the MN's signature using the MN's identity and authenticates the FA by verifying the FA's signature using the FA's identity.

The HA transmits M_2, which is RRP's body, and its signature of the M_2 to the FA. The FA authenticates the HA by verifying the HA's signature using the HA's identity, appends its signature to the RRP, and then transmits it. The MN authenticates the HA by verifying the HA's signature using the HA's identity and authenticates the FA by verifying the FA's signature using the FA's identity.

3 Performance Evaluation

In this section, the previously described mechanisms are modeled, and their performance is evaluated.

3.1 Modeling

To evaluate each authentication mechanism, this model is as follows. There is only one AAA server in one domain. A handoff is classified into two types, a handoff in the same domain, and a handoff between different domains. The former is called intra-handoff, and the latter is called inter-handoff. Authentication during the intra-handoff process occurs in a local AAA server and authentication during the inter-handoff process occurs in the home AAA server [7]. The network topology for modeling is presented in [Figure 5].

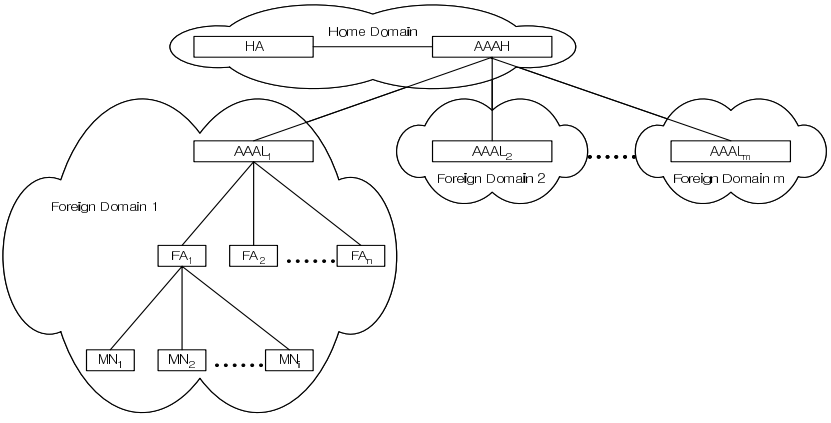


Fig. 5. Network Topology for Modeling

If ND_i^{RRQ} represents the network delay in each link, when each node transmits RRQ and ND_i^{RRP} represents network delay in each link when each node transmits RRP, total network delay ND is expressed using the following equation:

$$ND = \sum ND_i^{RRQ} + \sum ND_i^{RRP} \tag{1}$$

Herein, network delay of the same link is calculated separately in each direction because the size of RRQ and the size of RRP can be different from each other and the size of the packet affects network delay.

If PD_i^{RRQ} represents the general processing delay occurring by routing and register-ing the RRQ in each node, and PD_i^{RRP} is a general processing delay, occurring through routing and registering the RRP in each node, total routing and registration processing delay PD is expressed using the following equation:

$$PD = \sum PD_i^{RRQ} + \sum PD_i^{RRP} \tag{2}$$

If AD_i^{RRQ} is the RRQ authentication processing delay in each node and AD_i^{RRP} represents the authentication processing delay of RRP in each node, total authentication processing delay AD is expressed using the following equation:

$$AD = \sum AD_i^{RRQ} + \sum AD_i^{RRP} \tag{3}$$

During inter-handoff, the total delay D_{INTER} is expressed using the following equation:

$$D_{INTER} = ND + PD + AD \tag{4}$$

Similarly, during intra-handoff, total delay D_{INTER} is expressed by the following equation:

$$D_{INTER} = L_{ND} + L_{PD} + L_{AD} \tag{5}$$

If N networks exist, and the average number of networks in one domain is k, M which represents the average amount of MN movement during inter-handoff, is ex-pressed using the following equation:

$$D_{INTER} = D_{INTER} * M + D_{INTER} \tag{6}$$

3.2 Results

The cumulative handoff delay is calculated using the previously derived equations and system parameters in Table 1.

<Table> System Parameters[8][9][10]

Network Delay			
Bit Rate		Propagation Time (1 hop)	
Wired	100/10 Mbps	Wired	500 ns
Wireless	10 Mbps	Wireless	65 ns
Distance between hops		Number of hops	
Wired	100 m	MN-FA, FA-AAAL, AAAH-HA	1 hop
Wireless	50 m	AAAL-AAAH	5 hops
Processing Delay			
Routing and Registration Time		1 ms	
Authentication Time	MD5	5.12 μs	
	RSA-512 Signature	1.92 ms	
	RSA-512 Verification	0.13 ms	

Fig. 6. System parameter [8][9][10]

The wireless environment parameters are based on 11Mbps, semi-open office using 802.11b wireless LAN standard. In wired environments, the propagation speed is between $2.0*10^8$ and $3.0*10^8$, and in this paper, $2.0*10^8$ is used. Network delay includes transmission delay and propagation delay. Based on bit rates, transmission delay of wired and wireless is 10/100ns, and 100ns respectively. In wired environments, the distance between hops is assumed to be 100m while in wireless environments, the distance between hops is assumed to be 50m. In the same domain, the number of hops between nodes is assumed to be 1 hop while in different domains, the number of hops between nodes is assumed to be 5 hops.

In 100Mbps wired environments, the relationship between the number of handoffs and cumulative handoff delay is presented in [Figure 7, Left]. The performance rank is ordered as default authentication, partial certificate based authentication, identity based authentication, and certificate based authentication. Partial certificate based authentication reduces authentication processing delay and network delay to transmit certificates using partial symmetric key based authentication. Identity based authentication eliminates network delay when transmitting certificates, by eliminating the requirement for a certificate. The reason partial certificate based authentication is superior over identity based

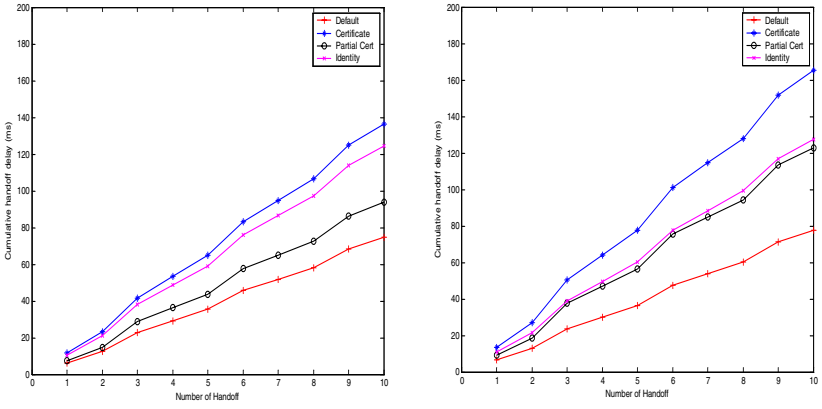


Fig. 7. Relationship between the Number of Handoffs and Cumulative Handoff Delay (Left: 100Mbps Wired Environments, Right: 10Mbps Wired Environments)

authentication, is that the network delay used to send certificates is mitigated due to the high speed wired environments.

In 10Mbps wired environments, the relationship between the number of handoffs and cumulative handoff delay is presented in [Figure 7, Right]. Similarly, performance rank is ordered as default authentication, partial certificate based authentication, identity based authentication, and certificate based authentication. However, performance difference between partial certificate based authentication and identity based authentication is smaller than that in 100Mbps wired environments, because network delay when sending certificates increases. If the bit rate of a wired environment is much less than 10Mbps, identity based authentication is expected to provide superior performance over partial certificate based authentication.

4 Conclusion

Symmetric key based authentication, using HMAC-MD5 provided in Mobile IPv4 standard is fast but suffers from the key distribution problem. Key distribution between a MN and HA is slightly cumbersome, but possible, however, key distribution between a MN and FA or between a HA and FA is impossible because a MN can move to any network in any domain. To solve this problem, public key based authentication mechanisms were proposed. The previously proposed pure certificate based authentication is not suitable for a mobile terminal suffering from low network band-width and low computation power, because large network overhead is created when sending certificates and a large processing overhead is required when processing the public key algorithm. To solve these problems, partial certificate based authentication and identity based authentication are proposed. However, they still create more over-head over symmetric key based authentication. This paper evaluates these public key based authentication

mechanisms, presenting the current direction of public key based authentication mechanisms, providing an indication of future mechanisms.

In the future, advantages from the previously proposed public key based authentica-tion mechanisms will be extracted, and disadvantages will be eliminated, creating a new authentication mechanism.

References

1. C. Perkins, 'IP Mobility Support for IPv4', RFC 3344, August 2002.
2. U.S. DEPARTMENT OF COMMERCE / National Institute of Standards and Technology, 'ENTITY AUTHENTICATION USING PUBLIC KEY CRYPTOCRAPHY', February 1997.
3. A. Shamir, 'IDENTITY-BASED CRYPTOSYSTEMS AND SIGNATURE SCHEMES', in Proc. of Crypto '84, LNCS, vol. 196, pp. 47-53, Springer-Verlag 1985.
4. S. Jacobs, S. Belgard, 'Mobile IP Public Key Based Authentication', INTERNET DRAFT, draft-jacobs-mobileip-pki-auth-03.txt, July 2001.
5. Sufatrio, Kwok Yan Lam, 'Registration Protocol: A Security Attack and New Secure Mini-mal Public-Key Based Authentication', ISPAN'99, June 1999.
6. Byung-Gil Lee, Doo-Ho Choi, Hyun-Gon Kim, Seung-Won Sohn, Kil-Houm Park, 'Mobile IP and WLAN with AAA Authentication Protocol using Identity-based Cryptography', ICT 2004, February 2003.
7. A. Hess, G. Schaefer, 'Performance Evaluation of AAA / Mobile IP Authentication', Tech-nical Report TKN-01-012, Telecommunication Networks Group, Technische Universit?t Berlin, August 2001.
8. Hoseong Jeon, Hyunseung Choo, Jai-Ho Oh, 'Identification Key Based AAA Mechanism in Mobile IP Networks', Springer-Verlag Lecture Notes in Computer Science, vol. 3043, pp. 765-775, May 2004.
9. C. L. Beaver, D. R. Gallup, W. D. NeuMann, M. D. Torgerson, 'Key Management for SCADA', SAND2001-3252, March 2002.
10. Proxim Corporation, 'ORiNOCO AP-2500 Access Point User Guide', Sofeware v2, March 2004.

Network-Initiated Fast Handover Scheme Using Virtual Connection over All-IP-Based Wireless Systems*

SungHo Kim[1], JaeJoon Cho[1], Yong Kim[2], and Sunshin An[1]

[1] Computer Network Lab., Dep. Of Electronics Engineering, Korea University
Sungbuk-gu, Anam-dong 5ga 1, Seoul, Korea, Post Code: 136-701,
Phone: +82-2-925-5377, Fax: +82-2-3290-3674
{shkim, jjj, sunshin}@dsys.korea.ac.kr
[2] KT Convergence Research Lab
yongkim@kt.co.kr

Abstract. Most of mobility protocols use mobile host (MH)-initiated schemes. However they may cause excessive signaling traffic and long latency for packet delivery. We propose cross-layer, i.e., layer-2 (L2) and layer-3 (L3), handover mechanism with network-initiated handover. It makes virtual connection between the home agent (HA) and the next neighboring Access Control Routers (ACRs), called *candidate ACRs*, which the MH is predicted to be attached to, with information such as required bandwidth, Quality-of-Service (QoS), and estimated handover delay. Proposed scheme searches candidate ACRs using predicted traveling distance of MH for L3 handover and makes virtual connection prior to L2 handover. In this paper, our goals are to decrease packet loss rate and packet delivery latency which occur due to frequent handover over all-IP-based wireless systems. We have evaluated the performance of our scheme through a series of simulations using the Network Simulator 2 (ns-2).

1 Introduction

Over the past years, wireless networks are evolving towards IP-based infrastructures to allow a seamless integration between wired and wireless networks [1]. Mobile IP (MIP) protocol which is standardized by IETF (Internet Engineering Task Group) has known very well for supporting layer-3 (L3) mobility [3]. However, it is also well-known that MIP might show relatively large handover latency and packet loss rate when a mobile host (MH) moves into another domain,[1] so that MIP is only suitable for provisioning macro-mobility. In order to complement the weak point of MIP, a number of IP-level micro-mobility protocols such as Cellular IP (CIP), Hawaii, and MIPv4 Regional Registration protocols have been proposed, designed, and

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

[1] In this paper, the terminology "domain" stands for a group of subnets.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 72 – 81, 2006.

implemented, and those can cope with the problem of MIP by providing fast, seamless, and local handover control [2].

CIP is proposed to provide local mobility and support handover for frequently moving MHs. It supports fast handover and paging in CIP access networks. To support mobility between different CIP networks, it can work together with MIP which provides macro-mobility. CIP also uses distributed paging cache and distributed routing cache for location management and routing respectively [4]. Hawaii is a domain-based approach to support micro-mobility. All issues related to mobility management within one domain are handled by a gateway, called domain root router. When an MH is in its home domain, packets which are destined to the MH are routed using typical IP routing. When the MH is located in a foreign domain, however, packets which are heading for the MH are intercepted by its home agent (HA) first. The HA establishes a tunnel to forward the packets between HA, itself and the domain root router which is serving the MH now. The domain root router routes the packets to the MH using the host-based routing entries [5].

Whenever a MH moves into a new domain in CIP and Hawaii, basically, the MH sends a Routing Update (RU) Message to the domain root router which is located in the domain. Accordingly, the amount of transmitted control messages between domain root router and each base station are increased in this domain because routing table of all nodes should be updated. Therefore, handover delay and packet loss rate might be increased in proportion to the number of MHs which come into or moves out of the current domain [8].

Link layer-assisted MIP scheme uses a Medium Access Control (MAC) bridge which is connected to different wireless LANs (WLANs) and the MAC Bridge is configured not to send MAC frames unless their destination MAC addresses are registered in the filtering database [9].

It becomes generally known that L3 required handover latency is 7-8 seconds approximately and layer-2 (L2) average handover latency is about 3-4 ms [9].

In this paper, we provide a cross-layer (i.e., L2 and L3) handover mechanism which is mainly initiated by network side. In our proposed scheme, virtual connections are established between HA and neighboring ACRs, which are called *candidate ACRs*, to achieve fast handover in both the L2 and L3 and to decrease overall handover latency.

The rest of the paper is organized as follows. Section 2 provides the proposed network model. Then we describe the new micro-mobility scheme in Section 3. The evaluation of our scheme is presented in Section 4 and we provide some concluding remarks in Section 5.

2 Proposed Network Model

2.1 Network Model Description

Fig. 1 shows a network model with all-IP-based wireless systems. We assume that frequent L3 handover happens in this network. Mobility management which is supported by our proposed scheme uses network-initiated handover mechanism. It

means that the initialization of handover is triggered firstly by not MH, but network-side entity, that is, ACR. The reason why network-initiated handover is used is that it can minimize L3-handover latency quite much via using the tremendous amount of bandwidth resource of emerging all-IP-based wired network. Therefore, it is clear that our network-initiated handover scheme can show higher performance of L3-handover latency than that of the legacy protocols such as CIP, Hawaii and MIP, since those protocols fundamentally use different approach, i.e., MH-initiated handover. In order to achieve our goal properly, we use a new concept of *virtual connection* that integrates information of L2 and L3 handovers.

Definition 1 - *Virtual connection*
Virtual connection is a kind of IP tunneling between candidate ACRs and HA (or domain route router in micro-mobility protocols) for accelerating L3 handover.

Old ACR (oACR; ACR 6 on Fig. 1) broadcasts message, in which MH identification (ID), oACR IP address, required QoS information, and estimated handover latency are included, to neighboring ACRs (ACR 3, 5, and 7 on Fig. 1). It means that a MH may move neighboring cells or RAS. ACRs that reply to oACR's message become candidate ACRs. After becoming candidate ACRs, they request registration to HA for the moving MH, and hence the HA replies to them. As a result, virtual connections are established for MH between candidate ACRs and HA. The actual mapping between a virtual connection and a real communication session is made after finishing L2 handover finally. When L2 handover occurs at RAS, MH sends its ID to nACR (new ACR), and nACR checks the validity of the ID comparing it with the previous information, *connection descriptor* at temporary tunneling table (TTT) as the final step. After a time period, each candidate ACR deletes an obsolete entry on its TTT. Otherwise, if a candidate ACR receives *virtual connection delete* message from oACR, the entry about MH is removed from TTT of the candidate ACR. More detailed mechanism about our proposed scheme will be discussed in the later section.

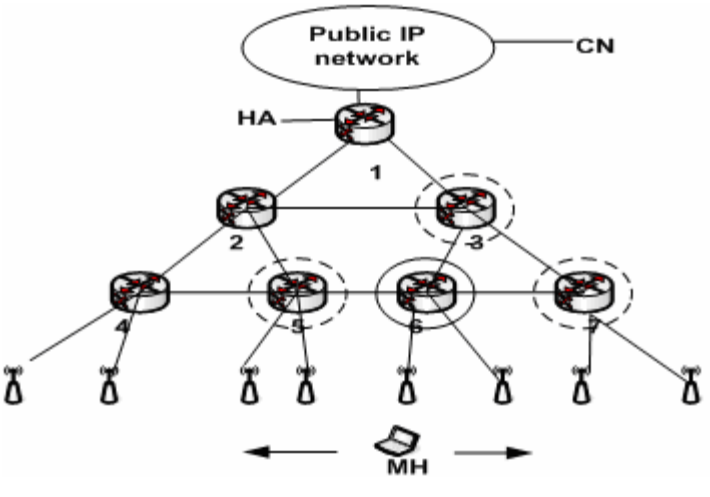


Fig. 1. Example of proposed all-IP-based network architecture

3 Network-Initiated Fast Handover Scheme Using Virtual Connection

3.1 Goals

Mobile IP to support wireless internet mobility has triangular problems. So a number of IP Micro mobility protocols have been proposed, designed, and implemented that complement mobile IP protocol by providing fast, seamless and local handover control. Handover mechanisms supported by most of micro-mobility in one domain have caused lots of traffic between domain gateway and each access router. These problems are that they increase handover latency and packet loss because of updating lots of routing tables. We propose new scheme that aims to cope with above problems.

We define new concepts called virtual connection. Candidate ACRs enable virtual connections to fast handover for a MH.

There are main differences betweens proposed scheme and other schemes.

First, proposed scheme implements mobility management using handover information of cross layer L2/L3.

Second, it searches next moving candidate ACRs for L3 handover and makes virtual connection before L2 handover happens.

Third, proposed scheme provides mechanism that hold continuous session even when MH does not complete handover at previous ACR and the MH move new candidate ACR then. It is possible because temporary tunneling for candidate ACRs has multi-path about moving MH. Fig. 2 shows handover processing.

In this paper, our goals are fast handover to solve handover delay and packet loss using virtual connection

3.2 The Mobility Model

In this paper, we have to know predicted motion of MH to search candidate ACRs.

Predicted motion can be made using The Gauss-Markov Mobility Model(GMMM). It is suitable for our scheme because of motion randomness and memories from previous time steps. A MH's velocity is assumed to be correlated in time and modeled by a Gauss-Markov process. The speed v_n of a MH at a time instant t_n can be written as [11]:

$$v_n = \alpha v_{n-1} + (1-\alpha)\mu + \sqrt{1-\alpha^2}\, x_{n-1} \tag{1}$$

Where $0 \le \alpha \le 1$, α is a tunable parameter that shows different levels of randomness, $\alpha = 0$ means linear motion and $\alpha = 1$ means Brownian motion, μ is the asymptotic mean of v_n as $n \to \infty$, and x_n is an independent and uncorrelated.

Based on (1), the moving direction d_n and the speed v_n of MH at a time instant t_n can be rewritten as [12]:

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{1-\alpha^2}\, x_{n-1} \tag{2}$$

$$v_n = \alpha v_{n-1} + (1-\alpha)\bar{v} + \sqrt{1-\alpha^2}\, x_{n-1} \tag{3}$$

Where $\bar{d}$ and $\bar{v}$ is asymptotic mean of the moving direction and speed $t_n \to \infty$. Using (2) and (3), location of MH (x_n, y_n) is as follows:

$$x_n = x_{n-1} + v_{n-1}\delta t \cos d_{n-1} \tag{4}$$

$$y_n = y_{n-1} + v_{n-1}\delta t \sin d_{n-1} \tag{5}$$

Where δt is the difference between the time instant t_n and t_{n-1}.

We assumed that a MH move based on GMMM, traveling distance of MH within time duration $\triangle t = (t_b - t_a)$ is made by summing the x and y directions:

$$\triangle x_{a,b}^M = \sum_{n=a+1}^{b} v_n^M \delta t \cos d_n^M \tag{6}$$

$$\triangle y_{a,b}^M = \sum_{n=a+1}^{b} v_n^M \delta t \sin d_n^M \tag{7}$$

Where $\delta t = \dfrac{\triangle t}{(a-b)}$, d_n^M and v_n^M mean direction & speed of MH at time t_n.

Therefore, we can obtain MH_{dv} value through ($\triangle x_{a,b}^M$,$\triangle y_{a,b}^M$).

We can find candidate ACRs using MH_{dv} value and then will establish virtual connection for the MH.

3.3 Handover Processing

As follows are procedures about virtual connection for L3, handover procedure for L2 and maintenance for virtual connection.

Virtual connection for L3

Step 1 The oACR broadcasts handover message (MH'ID, oACR' IP addr, required BW and QoS, estimated time to HO) to candidate ACRs using MH_{dv}

Step 2 Candidate ACRs check their resource and response to oACR

Step 3 The oACR sends handover confirm message to candidate ACRs

Step 4 The oACR requests registration about candidate ACRs for a MH using multiple CoA to HA

Step 5 The HA responses registration confirm message to oACR and candidate ACRs

Step 6 Candidate ACRs transmit connection message with each connection descriptor to oACR

Step 7 The candidate ACRs send virtual connection confirm message to HA

Step 8 Setup temporary tunneling for candidate ACRs

Handover procedure for L2

Step 1 The oACR broadcasts candidate ACRs information (in Fig 2, nACR1,2 and 3) to MH

Step 2 The MH requests candidate ACRs to oACR

Step 3 The oACR response information about candidate ACRs to MH

Step 4 The MH performs scanning procedure

Step 5 The MH selects best nACR among candidate ACRs regarding required parameters

Step 6 The MH requests HO message about nACR to oACR

Step 7 The oACR requests HO message (MH's ID) to nACRs

Step 8 The nACR responses HO message (Connection Descriptor) to oACR

Step 9 The oACR responses HO message (nACR's Connection Descriptor) to MH.

Step 10 The nACR makes complete tunneling to integration session between L2 and temporary tunneling for MH

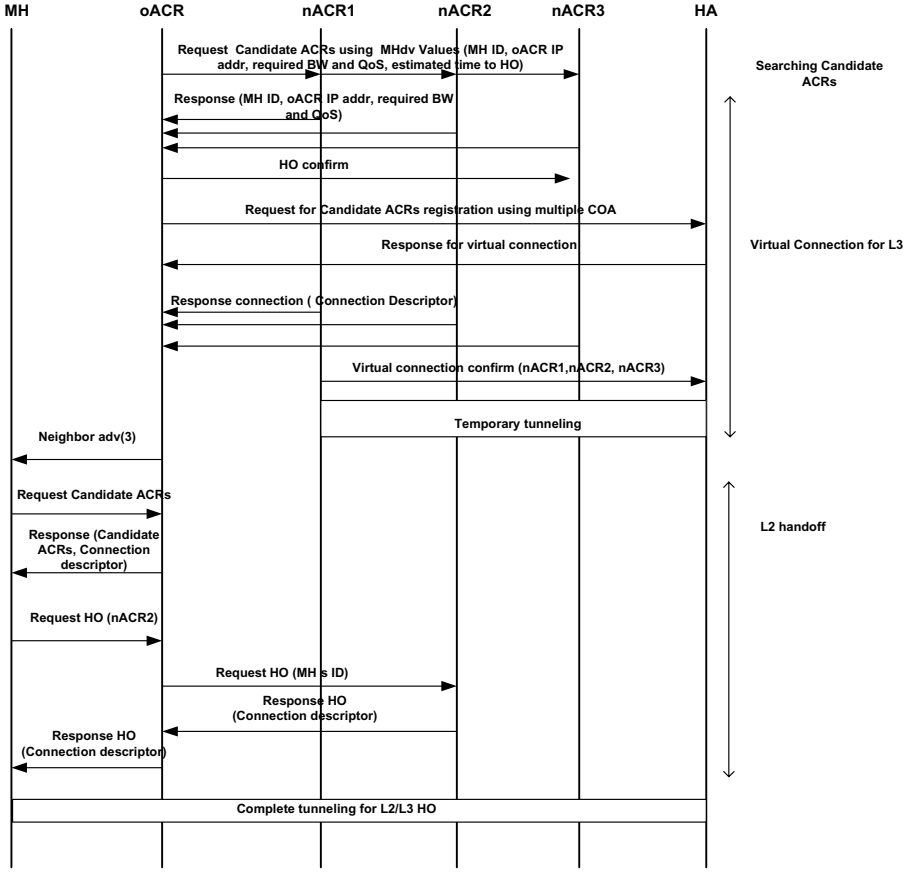


Fig. 2. Handover Process

Maintenance for virtual connection

We present two methods to release a MH on virtual connection table.

First, candidate ACRs and HA check their virtual connection table for a MH. If it holds idle state after time period, remove entry for the MH on virtual connection table.

Second, the nACR selected among candidate ACRs sends message to oACR and then the oACR sends release message about virtual connection for a MH to candidate ACRs .

4 Performance Evaluation

We have tested a performance evaluation of new fast handover scheme using virtual connection through simulations using CMU's wireless extensions for the Network Simulator (ns-2) and Columbia IP Micro-mobility software (CIMS). The nodes use 802.11 radio and MAC model provided by the CMU extensions. Radio propagation range for each node was 250 meters. Constant bit rate sources were used to generate traffic data, the size of which payload is 512, 1024bytes. We have evaluated our scheme using three metrics – packet receive ratio (that is packet loss ratio), control traffic overhead and packet delay. All simulations are performed using network topology shown in Fig. 1.

4.1 Simulation Model and Results

Fig. 3 shows packet delivery ratio when CN sends FTP packets to MH using TCP in Cellular IP, Hawaii and our scheme.

As you see, Hawaii MSF gets lots of packet loss and our scheme is better than CIP.

When an MH is in MH's foreign domain, HA intercepts first packets destined for the MH. The HA tunnel the packets to the domain root router serving the MH. In Fig. 1, domain root router is ACR 1. Hawaii MSF sends lots of message from old ACR to new ACR for path setup. So it takes much handover time required.

Our scheme shows good performance caused by reducing handover delay with virtual connection. It means that L3 has much more handover time required than L2 handover.

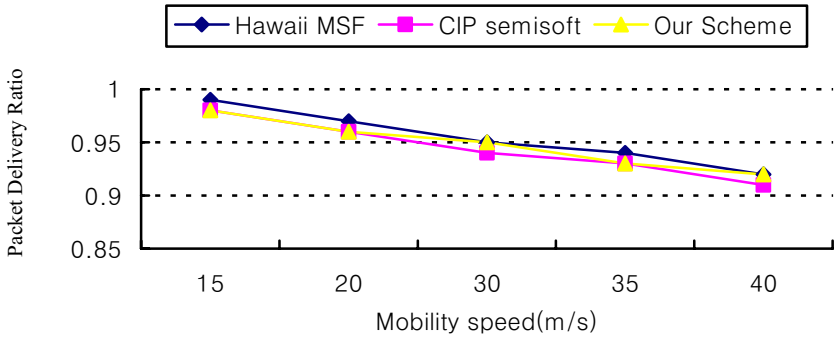


Fig. 3. Packet Delivery Ratio using TCP

Fig. 4 shows overhead about control traffic. As you see, proposed scheme gets lots of control messages. The reason is that oACR broadcasts frequently HO notification for the MH Handover to neighbor ACRs, so candidate ACRs response HO message with parameters. Handover procedure of proposed scheme gets successful fast handover, but it has lots of control traffic because of virtual connection. We think that these overheads of control traffic are of no significance in wired network.

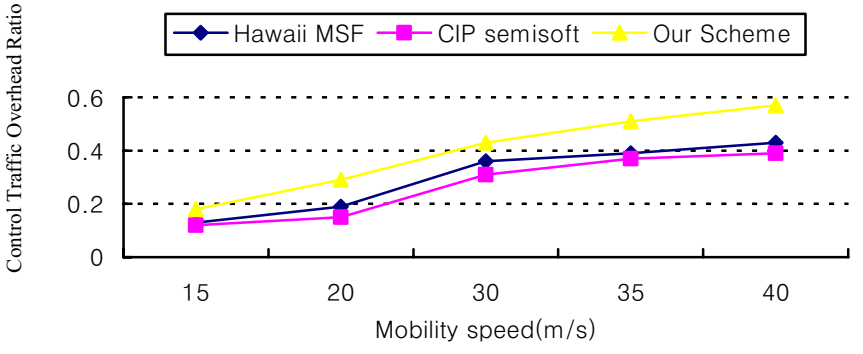


Fig. 4. Control Traffic overhead

Fig. 5 shows packet delay time, When CN sends FTP packets to MH using TCP in Cellular IP, Hawaii and our scheme. When handover happens frequently, Hawaii MSF sends lots of control message for path setup. So it delayed packets destined to MH. Also our scheme setups L3 handover using candidate ACRs before L2 handover occurs. It can reduce packet delay caused by cross layer handover mechanism.

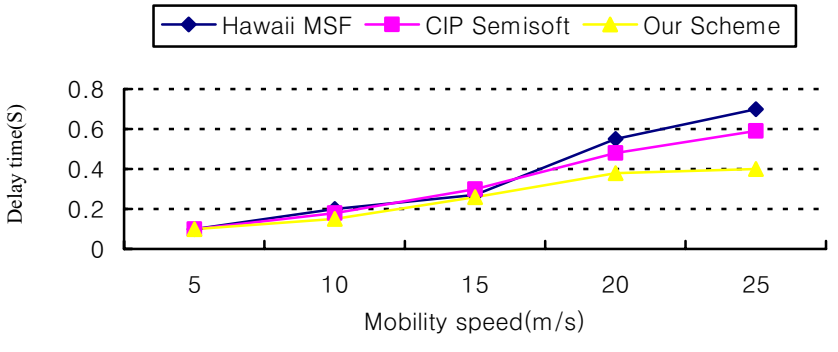


Fig. 5. Delay time (sec)

Fig. 6 shows packet delivery ration when CN sends FTP packets to MH using in Cellular IP, Hawaii MSF and our scheme. CBR packet size is set 10,000 here.

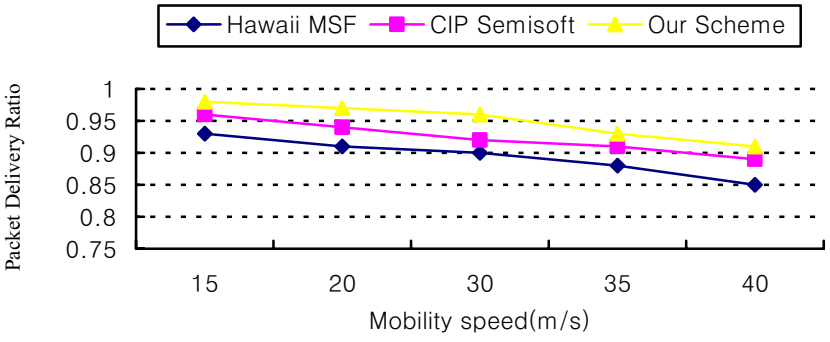


Fig. 6. Packet Delivery Ratio using UDP

5 Concluding Remarks

A number of IP Mobility protocols have been proposed for provisioning fast, seamless and locally controlled handover. Moreover, it becomes generally known that L3 handover takes approximately 7-8 seconds and required L2 handover latency is about 3-4 ms [9]. In this paper, we propose cross-layer handover mechanism in which any handover is activated by network-side entity such as ACR. It establishes virtual connection between HA and the next neighboring ACRs, which is called candidate ACRs in this paper, in order to achieve fast handover and decrease handover latency over all-IP-based wireless systems. As shown in of the Section 4, we evaluated that our scheme shows better performance of packet loss rate and packet latency than the legacy protocols. Although our proposed scheme needs relatively large amount of control traffic for establishing virtual connection, we consider that these overheads of control traffic are of no significance in the emerging wired network architecture thanks to its tremendous amount of available bandwidth. In the future work, we plan to reduce the number of virtual connection and to conduct further research to enhance performance of our proposed architecture especially in the emerging the next-generation broadband wireless access network like the WiBro (Wireless Broadband) in Korea.

References

[1] Campbell, and J. Gomez, “IP Micro-Mobility Protocols,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 4, pp. 45-53, October 2000.

[2] A. Campbell, J. Gomez, S. Kim, and C. Wan, “Comparison of IP Micro-mobility Protocols,” *IEEE Wireless Communications*, pp. 72-82, February 2002.

[3] C. Perkins, “IP Mobility support for IPv4,” *IETF RFC3344*, August 2002.

[4] A. Valko, “Cellular IP: A New Approach to Internet Host Mobility,” *ACM Computer Communication Review*, January 1999.

[5] R. Ramjee, T. La Porta, Salgarelli, S. Thuel, and K. Varadhan, “IP-Based Access Network Infrastructure for Next-Generation Wireless Data Networks,” *IEEE Personal Communications*, pp. 34-41, August 2000.

[6] E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IPv4 Regional Registration," online link; ftp://ftp.ietf.org/internet-drafts/draft-ietf-mobileip-reg-tunnel-07.txt, October 2002.

[7] Liesbeth Peters, Ingrid Moerman, Bart Dhoedt, and Piet Demeester, "Micro-Mobility Support for Random Access Network Topologies," in *Proc. IEEE WCNC'04*, March 2004.

[8] IAN F. Akyildiz, Jiang Xie, and Shantidev Mohanty, "A survey of Mobility Management in Next-Generation All-IP-Based Wireless Systems," *IEEE Wireless Communications*, August 2004.

[9] Hidetoshi Yokota, Akirad Idoue, Toru Hasegawa and Thoshihiko Kato, "Link layer Assisted Mobile IP Fast Handover Method over Wireless LAN Networks," in *Proc. ACM MobiCom'02*, September 2002.

[10] Jiang Xie and I.E. Akyildiz, "A Distributed Dynamic Regional Location Management Scheme for Mobile IP," in *Proc. IEEE INFOCOM'02*, vol. 2, pp. 1069-1078, June 2002.

[11] B.Liang, and Z. Haas, "Predictive Distance-Based Mobility Management for PCS networks" Proceeding of the ACM International workshop on Modeling and Simulation of Wireless and Mobile systems, August 1999.

[12] Kai-Ten Feng and Tse-En Lu, "Velocity and Location Aided Routing for Mobile Ad hoc networks" 2004 IEEE

Efficient Mechanism for Source Mobility in Source Specific Multicast*

Hoyoung Lee[1], Sunyoung Han[1,**], and Jin Pyo Hong[2]

[1] Department Computer Science and Engineering, Konkuk University, 1 Hwayang-dong, Kwangin-gu, Seoul, 143-701, Korea
{hylee, syhan}@cclab.konkuk.ac.kr
[2] Department Information and Communications Engineering, Hankuk University of Foreign Studies, San 89 Wangsan-ri, Mohyun-myun, Yongin-si, Kyungki-do, 449-791, Korea
jphong@hufs.ac.kr

Abstract. This paper describes an efficient mechanism for multicast tree reconstruction at source mobility in a source specific multicast (SSM). Source mobility makes its address changed so that the existing multicast tree corresponding to the home of address (HoA) of the mobile source should be rebuilt. But this causes a large amount of packet delay during the reconstruction time. This paper describes the reuse of the legacy multicast tree for a minimization of packet delay. So we expect to improve quality of multicast service in mobile environment because of the reduction of the tree rebuilding time and the followed overhead.

1 Introduction

With the growth of Internet-related service and base technology, Internet has greatly developed and it is widely spread to public users. In these days, the multimedia services such as an audio/video streaming service included a text service are mainly supported to users instead of legacy text-only services. The important things in these multimedia services are delivery speed and quality of service. The multicast protocol is more suitable than the unicast protocol for these services.

The multicast protocol was developed for a group communication in a difference from the legacy unicast and broadcast protocol. It can be divided into two modes according to the distribution of group members, which are dense-mode and sparse-mode. It also can be divided according to the tree construction methods, which are the shared tree and the shortest path tree. The classification such as SSM (Source Specific Multicast) and ASM (Any Source Multicast) follows whether the sender is one or many[3]. ASM is a suitable protocol for many-to-many group communication such as videoconference, but it has a serious difficulty of a deployment because it is hard to discovery the each source

* This research is supported by University IT Research Center Project.
** Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 82–91, 2006.

location and the complicated multicast tree has followed overheads. SSM is simpler than ASM because only the one-to-many model is supported and it is easy to deploy. It is suitable for the audio/video streaming service with only one source such as Internet TV.

SSM builds the shortest path trees rooted at the specific source. In ASM, receivers are easily to join in a group communication with a group address regardless of source address. It often causes an address collision problem when the same group address is used at the same time in same domain by a different group. But in SSM, the subscribers should use both source and group address for joining at a group. This address pair can be a global unique address.

Using the address pair, both source address and group address, SSM creates a (S,G) state in the intermediate multicast router during a joining process. It is different from (*,G) state in ASM. SSM is very efficient way at one-to-many communication.

In recent ubiquitous Internet, the host mobility becomes very important issues, and it is the same at the multicast. The mobility support of multicast protocol has been researched up to date, so it is called mobile multicast.

Two approaches for mobile multicast based on IETF Mobile IP have been proposed, that is remote subscription (RS) and bi-directional tunneling (BT) [6][7]. BT approach is that a mobile host sends and receives all multicast packets from its home network using unicast tunnels when it moves to another network. And RS is the approach for reconstruction of multicast tree by sending re-subscription message at the network where the mobile host is visiting.

Both approaches have an advantage and a disadvantage. BT has the advantage of offering multicast service without the existence of multicast router at the visited network. However, this has some serious drawbacks. First, BT approach has the problem of inefficient routing path because all multicast packets should be delivered via home agent at home network. Second, the approach causes network congestion by a phenomenon called the tunnel convergence problem. On the other hand, RS approach has the advantage of offering the optimized multicast routing path. However, this approach assumes that there should be one multicast router at least in the visited network, and it causes a packet delay by reason of multicast tree re-construction.

Host mobility should be supported not only at the receiver side but also at the sender side, which is source. ASM uses the (*,G) state in tree construction so that there are no difference between source and receiver mobility at both BT and RS. But because SSM uses the (S,G) state, when source moves from home network to another network, it causes the source address change from the home of address (HoA) to the care of address (CoA), and it affects greatly whole subscribers in RS, not BT. In BT, source packets are always sent to home agent at its home network so that subscribers can be received the packets from an existing routing tree. It is only necessary that HA just switches the source address from CoA to HoA. But it causes a BT problem, triangular routing, and it affects more seriously whole multicast traffic and reduces the whole bandwidth. This BT problem can be solved in RS with optimized routing path, but whole

receivers have to re-subscribe to the changed source address. According to this re-subscribe message, the (S,G) state alternates a (S',G) state with new source address S'. This process has a serious overhead at whole system.

In this paper, for the purpose of the source mobility support in SSM, we propose an efficient mechanism for the reduction of delay by multicast tree re-construction in RS.

2 Architecture and Operation

In SSM, the user wants to join in a group sends a subscribe message to a source with both source and group address. The intermediate multicast router bypassing this subscribe message creates a (S,G) state on itself. Finally, it is constructed that the multicast routing tree is rooted at the source based on this (S,G) state.

If the source address is changed by its moving to another network such as S' from S, the prior (S,G) state should be changed to the (S',G) state. To do this process, whole group members should send again the re-subscribe message to the source S'. This causes the overhead of whole system, and if the source moves more often to other networks, the whole system loads might be much heavier.

For the purpose of a solution of this problem, we propose the method that is the re-use of the prior constructed multicast tree, which has the (S,G) state. This method reduces the total amount of a delay for rebuilding whole multicast tree.

2.1 Initial Multicast Tree Construction

Fig. 1 describes an initial tree construction. A source is located in its home network, and it has a home of address S. Each intermediate multicast router keeps up a multicast routing table (MRT), and now it sets up the (S,G) state. The multicast packets originated from the source forward to each subscribers based on this (S,G) state.

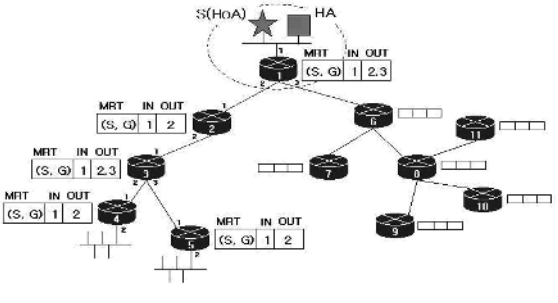


Fig. 1. Initial multicast routing tree with a (S,G) state

2.2 Initial Routing Update

If the source S leaves its home network and attaches to another network such as Fig. 2, it gets a new care of address S' from its attached network, and it tries to do a binding update with HA based on Mobile IP. After these processes are done, in legacy RS, the new CoA is notified to each group member by some ways such as the HA notification. Then each group member tries to re-subscribe with both this new CoA and the group address at the almost same time. Finally, it is constructed that the multicast routing tree is rooted at the source S', but it takes long time for completing the tree construction because of the heavy traffic of re-subscription message.

On the other hand, in our approach, the re-subscription process doesn't start from a receiver side but from a source side. The source creates the initial routing update message and sends it to a home agent (HA). Each intermediate multicast routers bypassing this update message adds the (S',G) state on its MRT. If there

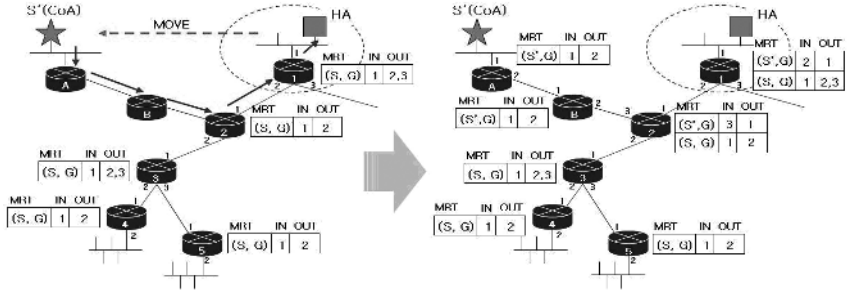


Fig. 2. A source moves to another network and sends a initial routing update message to its HA. The routing state changes a (S,G) to a (S',G) on intermediate multicast router.

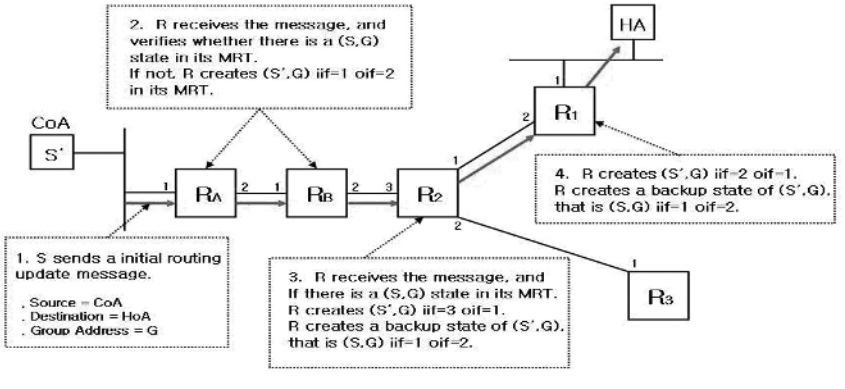


Fig. 3. Example: an initial routing update process

is the (S,G) state in its MRT, it alternates the prior (S,G) state with this new state. In the case of the alternation, the prior (S,G) state becomes a backup state. Fig. 3 shows an example of this initial routing update process.

2.3 Routing Update from Home Agent

In Fig. 4, after the HA received the initial routing update message from the source S', it creates the routing update message and multicasts it with a source address S and a destination address G. This update message is forwarded to each receiver, and intermediate multicast routers alternate the (S,G) state with the (S',G) state. It also lets the (S,G) state be the backup state. After this update is done, the tree re-construction process is completed, and the source S' can forward the multicast packets using the new routing tree with the (S',G) state. Fig. 5 shows an example of this update process from HA.

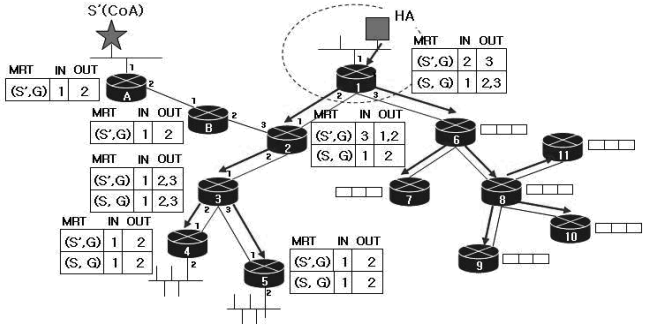


Fig. 4. HA multicasts a routing update message to whole group members

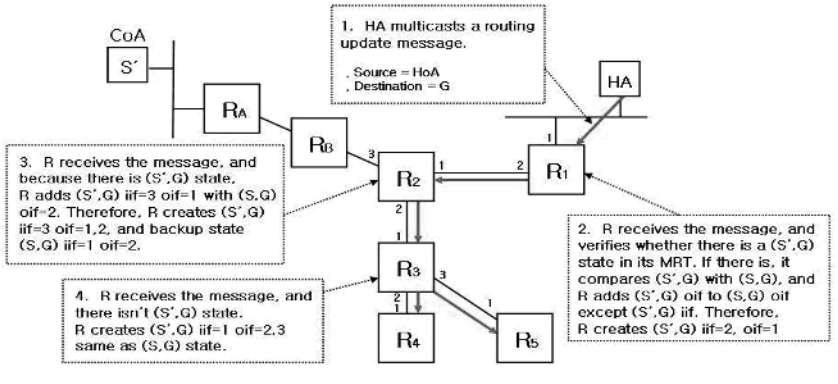


Fig. 5. Example: a routing update process from HA

2.4 Routing Update in Source Handoff Toward Another Network

If the source S' leaves its attached network and moves to another network such as Fig. 6, source address S' is changed to another CoA S". The source S" begins the initial routing update such as Fig. 2, and then HA creates the routing update message and multicasts it through a backup state (S,G). Each intermediate multicast router just alternates the (S',G) state with the (S",G). The multicast router on the previous network detects no source, and sends the prune message to HA, then the (S',G) state is deleted on intermediate multicast routers. In the result of these processes all (S',G) states are removed.

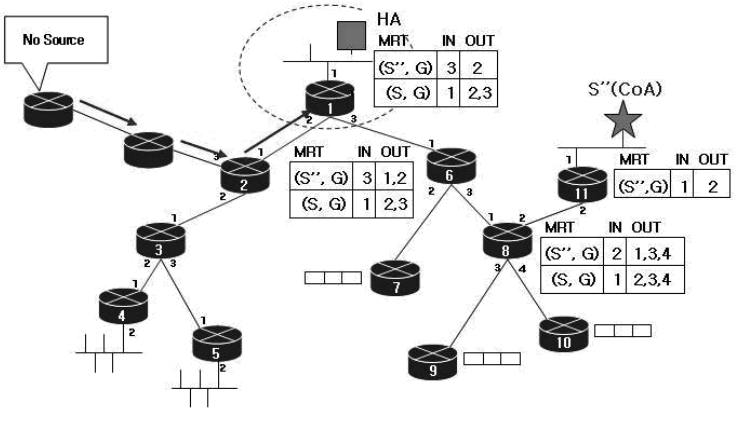


Fig. 6. Routing update when the source S' moves to another network

2.5 Member Join

A host wants to join newly in a group sends a subscribe message to the source using its destination HoA whether it is at home network or not. The subscribe message on going to a source creates the (S,G) state on the intermediate multicast router. If the source is at its home network, immediately a shortest path tree is constructed with the (S,G) state. If the source isn't at home network but at another network, HA unicasts the routing update message to the subscriber, then it creates the new state with a source CoA. This join process lets the backup state kept for all multicast members

3 Performance Evaluation

For the purpose of performance evaluation, this section compares our mechanism to the legacy BT and RS approach in mobile multicast when a source moves to another network.

This paper assumes that a multicast tree is made up with subsets of perfect k-ary trees of depth n. Also it assumes that only one multicast member is located at each leaf node. Therefore, the total number of members is k^{n-1}. Although in a real world the multicast members are located at any node in multicast tree, the assumption should be enough to calculate the network load with the increase of the number of member. We suppose that all links have the same propagation delay d and a single link between nodes. We ignore its own processing and queuing delay for packet delivery.

When a source moves from root to another node in this multicast environment, let RsT be the tree reconstruction time in RS approach. First, each member detects a source handoff, creates and sends a re-subscription message to new source address. It will make a new multicast tree with new (S',G) state. If all members send this re-subscription message at the same time for the sake of simplicity, some messages should pass through a root node for reaching the source. Otherwise the others shouldn't pass through it. Let RsT_{Ro} be a delivery time to source passing through a root node, and let RsT_{Rx} be the others. Although each message is sent at the same time, each arriving time at the source is different because the link between nodes allows only one packet to pass through.

If there is a mobile source at the p level, the first and the last arriving time at both RsT_{Rx} and RsT_{Ro} are expressed as follow.

$$\begin{aligned} RsT_{RxFirst} &= (n-p)\cdot d \\ RsT_{RxLast} &= \left(k^{n-p-1}+n-p-1\right)\cdot d \\ RsT_{RoFirst} &= (n+p-2)\cdot d \\ RsT_{RoLast} &= \left\{(k-1)\,k^{n-2}+n+p-3\right\}\cdot d \end{aligned} \tag{1}$$

It might meet at a same link in the case that messages pass through the root or not, and then it makes a delivery delay, $RsT_{overlay}$, and it expressed as follow.

$$\begin{aligned} RsT_{overlay} &= \left(RsT_{Ro}\bigcap RsT_{Rx}\right) = RsT_{RxLast} - RsT_{RoFirst} \\ &= \left(k^{n-p-1}-2p+1\right)\cdot d \end{aligned} \tag{2}$$

By the equation (1) and (2), the final arriving time at the source is expressed as follow.

$$\begin{aligned} RsT_{First} &= RsT_{RxFirst} = (n-p)\cdot d \\ RsT_{Last} &= \left\{RsT_{RoLast}+\sum_{i=2}^{p}\left(k^{n-i-1}-2i+1\right)\right\}\cdot d \\ &= \left\{(k-1)\,k^{n-2}+n+p-3+\sum_{i=2}^{p}\left(k^{n-i-1}-2i+1\right)\right\}\cdot d \end{aligned} \tag{3}$$

Finally, equation (3) is the time for a reconstruction of multicast tree. On the other hand, the tree reconstruction time $EsmT$ in our approach for source mobility adds the arriving time of a tree update message from a mobile source to

its home network, that is root, and the arriving time to each members by HA's multicasting a tree update message. The tree update time $EsmT$ is expressed as follow.

$$EsmT = \{(p-1) + (n-1)\} \cdot d = (n+p-2) \cdot d \tag{4}$$

We assume that a packet delivery time between nodes is $d = 1ms$ and k-ary tree is a binary tree. Fig. 7 illustrates the comparison result of tree re-build time between RS and our approach. Fig. 7-(a) shows the comparison result that a source moves to each level of multicast tree in a group size 2^{10}. Fig. 7-(b) shows the tree reconstruction time according to a group size when a source moves to the specific level of tree.

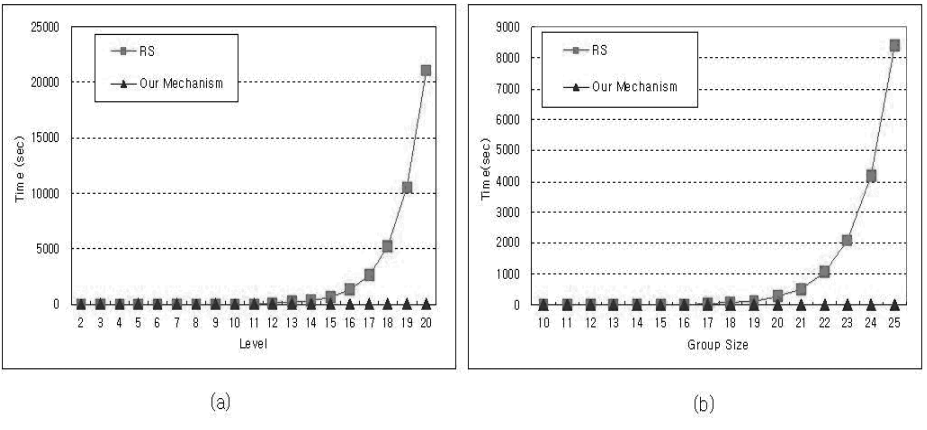


Fig. 7. Multicast tree reconstruction time at source handoff

Note that the curve increases by geometric progression in RS approach as a source moves more distant from its home network or a group size increases more and more. On the other hand, the tree reconstruction time is much faster and more stable in our approach.

In this multicast tree, we compare our approach to both BT and RS approach in the point of the packet delivery time from the mobile source to each member. The packet delivery time DT of both BT and RS approach is expressed as follow.

$$\begin{aligned} DT_{RS} &= (n-1) \cdot d \\ DT_{BT} &= \{(p-1) + (n-1)\} \cdot d = (n+p-2) \cdot d \end{aligned} \tag{5}$$

On the other hand, the average delivery time DT_{ESM} in our approach is expressed as follow.

$$DT_{ESM} = \left\{ \frac{1}{P} \sum_{i=0}^{p-1} (2i + n - p) \right\} \cdot d \tag{6}$$

Likewise we assume the packet delivery time between nodes is $d = 1ms$, Fig. 8 illustrates the delivery time to each member in a group size $n = 20$. In this graph we obtain: the farther source moves away from its home network, the more delivery time in BT approach. Note, however, the curve corresponding to our approach is much smoother and increases much fewer such as RS.

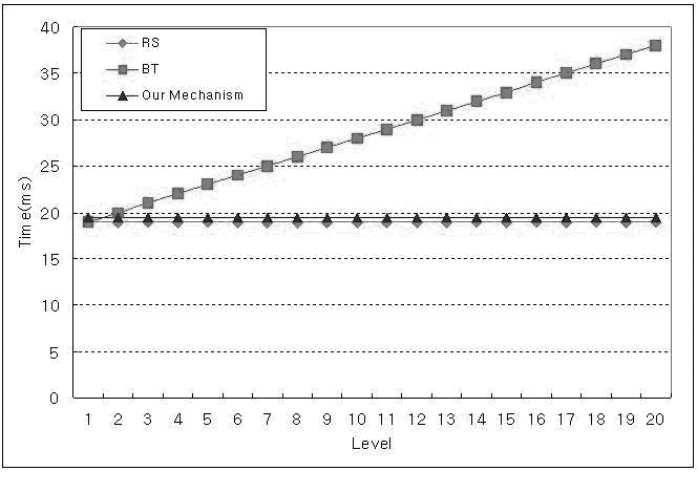


Fig. 8. The packet delivery time at source handoff

4 Conclusion

In the recent ubiquitous Internet, host mobility is an important issue. This issue is also important at the multicast protocol that is suitable for the large multimedia service. Two approaches for a mobility support at the multicast protocol have been researched, that is bi-directional tunneling and remote subscription. But these approaches have some drawbacks, so it's not acceptable to apply as they are.

In this paper we presented the mechanism to improve efficiency at the source mobility. In Source-Specific Multicast, source handoff affects whole routing tree, and this causes a low service quality. So we proposed the mechanism that makes the change of the routing tree minimized, and consequently we expect that this might improve a service quality much higher.

References

1. C. Perkins (ed.): IP Mobility Support for IPv4, RFC 3344, August 2002.
2. B. Cain et al.: Internet Group Management Protocol, Version 3, RFC 3376, October 2002.
3. S. Bhattacharyya (ed.): An Overview of Source-Specific Multicast (SSM), RFC 3569, July 2003.

4. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, RFC2362, June 1998.
5. Rolland Vida, Luis H.M.K. Costa and Serge Fdida: M-HBH-Efficient Mobility Management in Multicast, NGC 2002, October 2002.
6. George Xylomenos and George C. Polyzos: IP Multicast for Mobile Hosts, IEEE Commun. Mag., vol. 35, no. 1, 1997, pp. 54-58.
7. Tim G. Harrison, Carey L. Williamson, Wayne L. Mackrell and Richard B. Bunt: Mobile Multicast (MoM): Multicast Support for Mobile Hosts, ACM/IEEE MOBICOM '97, September 1997.
8. G. Phillips, S. Shenker and H. Tangmunarunkit: Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law, SIGCOMM, September 1999.
9. C. Janneteau, Y. Tian, S.Csaba, T. Lohmar, H. Y. Lach and R. Tafazolli: Comparison of Three Approaches Towards Mobile Multicast, IST 2003, Feburary 2003.
10. V. Chikarmane, R. Bunt and C. Williamson: Mobile IP-based Multicast as a Service for Mobile Hosts, Proc. 2nd Int'l Workshop on Services in Distributed and Networked Environment, 11-18, 1995.
11. V. Chikarmane, C. L. Williamson, R. B. Bunt and W. Mackrell: Multicast Support for Mobile Hosts using Mobile IP: Design Issues and Proposed Architecture, ACM/Baltzer Mobile Network and Applications, vol. 3, no. 4, 1999, pp. 365-79.
12. Imed Romdhani, Mounir Kellil, and Hong-Yon Lach: IP Mobile Multicast: Challenges and Solutions, IEEE Communications Surveys, First Quarter 2004, Volume 6, No. 1.

A Reliable Multicast Routing Scheme in Mobile IP Networks*

Hong-ju Yeom[1], Hwa-sung Kim[1], and Sang-ho Lee[2]

[1] Dep. of Electronic and Communications Eng., Kwangwoon Univ., Korea
{nanta0201@, hwkim@daisy.}kw.ac.kr
[2] IP Mobility Research Team, ETRI, Korea
shlee@etri.re.kr

Abstract. The various multicast routing methods that have been proposed so far are classified into the two classes: HA (Home Agent) based multicast routing and FA (Foreign Agent) based multicast routing. However HA based multicast routing has problems of a non-optimal route. FA based multicast routing also has the problem in that it requires the frequent reconstruction of the multicast tree. In this paper, we propose the hybrid scheme for the reliable multicast routing. It is based on FA based hierarchical multicast routing method, in which the network is managed hierarchically to reduce the frequency of reconstruction of the multicast tree while guaranteeing the optimal route. Additionally, the proposed method tries to speedup the handoff by using the Mobile IPv4 Low latency handoff method by executing the L3 handoff before the L2 handoff is completed. And the adaptive buffering mechanism is adopted in a gateway or nFA (New Foreign Agent) to solve the multicast data loss.

1 Introduction

Recently, the mobile computer is publicized and the development of wireless networking technology makes the mobile computing environments practicable. And, the research for accepting demands like multicast service is required. In the mobile network environment, the location of the mobile host must be managed dynamically and the multicast tree must be reconstructed whenever the mobile hosts move. Therefore, it is inadequate to use the multicast protocol for the static network as it stands in the mobile networks, because it has various problems such as construction of a defective tree, the multicast packet loss, and the expensive cost.

In this paper, we propose the FA based hierarchical multicast routing method, in which the network is managed hierarchically to reduce the frequency of reconstruction of the multicast tree while guaranteeing the optimal route. At the

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)and the Research Grant from Kwangwoon University in 2005.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 92–101, 2006.

same time, the proposed method tries to speedup the handoff by using the Mobile IPv4 Low latency handoff method and by executing the L3 handoff before the L2 handoff is completed. The proposed method also tries to solve the multicast data loss by using the adaptive buffering mechanism in a gateway or nFA (New Foreign Agent) according to the actions.

This paper is organized as follows. Section 2 will explain the overview of the related work. Section 3 describes the proposed multicast mechanism. Section 4 presents the simulation results. Finally section 5 is the conclusion.

2 Related Works

The current IETF mobile-IP specification briefly proposes two approaches for supporting multicast service to mobile hosts [1-3]: foreign agent-based multicast (referred to as remote-subscription) and home agent-based multicast (referred to as bi-directional tunneling).

In foreign agent-based multicast, a mobile host has to subscribe to multicast groups whenever it moves to a foreign network. It is a very simple scheme and does not require any encapsulations. This scheme has the advantages of offering an optimal routing path and nonexistence of duplicate copies of datagram. However, when a mobile host is highly mobile, its multicast service may be very expensive because of the difficulty in managing the multicast tree. Furthermore, the extra delay incurred from rebuilding a multicast tree can create the possibility of a disruption in multicast data delivery.

In home agent-based multicast, data delivery is achieved by unicast mobile IP tunneling via a home agent. When a home agent receives a multicast datagram destined for a mobile host, it encapsulates the datagram twice (with the mobile host address and the care-of address of the mobile host) and then transmits the datagram to the mobile host as a unicast datagram. This scheme takes advantage of its interoperability with existing networks and its transparency to foreign networks that a mobile host visits. However, the multiple encapsulation increases the packet size, and a datagram delivery path is non-optimal since each delivery route must pass through a home agent. Furthermore, if multiple mobile hosts belonging to the same home network visit the same foreign network, duplicate copies of multicast datagram will arrive at the foreign networks.

2.1 MoM (Mobile Multicast) Protocol in the HA Based Routing

In the HA (home agent) based multicast routing, called MoM (Mobile Multicast), a home agent is responsible for tunneling multicast datagram to the mobile host. In home agent-based multicast schemes, a home agent forwards a separate copy of multicast datagram for each mobile host. even if all mobile hosts that wish to receive the multicast datagram are in the same foreign network. However, by MoM protocol, the home agent forwards only one copy of the multicast datagram to each foreign network that contains its mobile hosts. Upon receiving the

multicast datagram, a foreign agent delivers it to mobile hosts using link-level multicasting. This scheme reduces the number of duplicate multicast datagram and the additional load on low bandwidth wireless links. But there still exists a problem, referred to as the tunnel convergence problem [4,5], resulting from the fact that multiple tunnels from different home agents can terminate at one foreign agent. Thus, when multiple home agents have mobile hosts on the same foreign network, one copy of every multicast datagram is forwarded to the same foreign agent by each home agent. Therefore, the foreign agent suffers from the convergence of tunnels set up by each home agent.

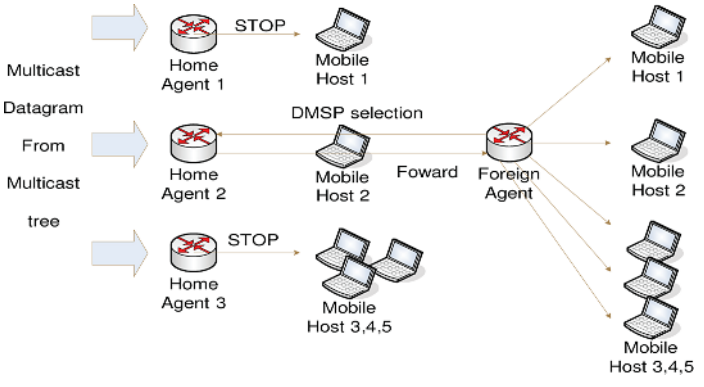


Fig. 1. DMSP selection in MoM protocol

To solve this problem, the foreign agent appoints one home agent as the DMSP (Designated Multicast Service Provider) for the given multicast group. The DMSP forwards only one datagram into the tunnel, while other home agents that are not the DMSP do not forward the datagram, as shown in figure 1.

2.2 MMA (Multicast by Multicast Agent) Protocol in the FA Based Routing

MMA protocol introduces the Multicast Agent (MA) and the Multicast Forwarder (MF). MAs provide multicast service to mobile hosts. Each MA has the information of a single MF selected among several MFs, per multicast group. MF of an MA (e.g., MA1) is the MA selected among MAs which are located near the MA1 and belongs to the multicast tree of a given multicast group. The MF is in charge of forwarding multicast datagram to MA1. The MF of an MA may be the MA itself when its local network is included in the multicast tree, or the MF can be an MA in another network that belongs to the multicast group when its local network is not covered by the multicast tree.

In the MMA protocol [6], two distinct methods are used depending on whether a mobile host's visiting network belongs to a multicast tree or not. If the visiting network belongs to a multicast tree, the mobile host directly receives multicast

data from the local multicast router in the network. If the visiting network does not belong to a multicast tree, multicast data are delivered to a mobile host through tunneling from an MA that is included in the multicast tree of a given multicast group and located in a network close to the mobile host's visiting network. In the former case, the MF of the MA is configured with the MA itself while, in the latter case, the MF of the MA is changed to an optimal MF value by using the MF information of the mobile host.

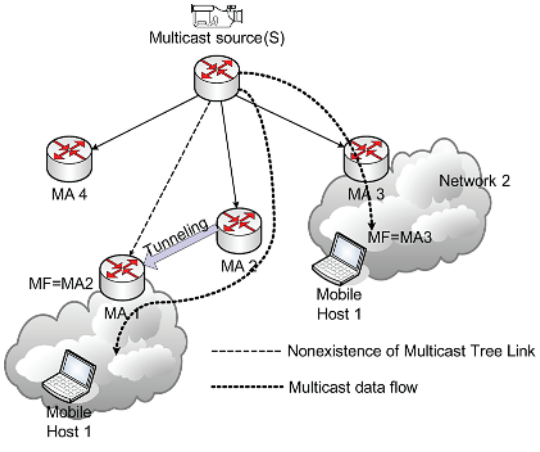


Fig. 2. Operation of MMA protocol

Initially, if a mobile host wants to join a group in a foreign network, subscription is done through an MA in the foreign network, which must be a tree node of the multicast group. If not, the MA starts a tree joining process. This MA configures the MF value of the multicast group with the MA itself, and delivers multicast datagram to the mobile host in the network.

The multicast tree joining process may be required by visiting mobile hosts according to the optional function of mobile hosts. Whenever a host moves to a new network and registers with a new MA, the new MA executes the multicast tree joining process, which is similar to that in the foreign agent based multicast protocols. While setting up a connection to the multicast tree, a mobile host receives forwarded data from its MF, and thus there is no service disruption period. When the joining process finishes, multicast datagram are delivered directly to the mobile host through an optimal path, just as in the foreign agent based multicasting. This joining process creates an overhead of reconstructing the multicast tree and extra time, but it presents an optimal delivery route. During the time delay, the disruption of multicast data delivery is reduced since datagram are forwarded to the mobile host through tunneling from the MF.

3 Proposed Multicast Mechanism

In this paper, we propose the FA based hierarchical multicast routing method, in which the network is managed hierarchically in order to decrease the frequency of reconstruction of a multicast tree while it guarantees the optimal route. The proposed method also tries to speedup the handoff by using the Mobile IPv4 Low latency handoff method by executing the L3 handoff before the L2 handoff is completed [7]. The proposed method also tries to solve the multicast data loss by using the adaptive buffering mechanism in a gateway or nFA (New Foreign Agent) according to the situation. Of course, there will be a cost increase in the form of added signaling, but we believe it is more desirable to guarantee the service quality than the cost of additional messages. Using this method, the service requirement that is delay-sensitive is satisfied. Using the three methods mentioned above, problems that occur in multicast service (i.e., packet loss during handoff and frequent reconstruction of multicast tree) will be solved. Figure 3 shows the schematic description of the proposed mechanism.

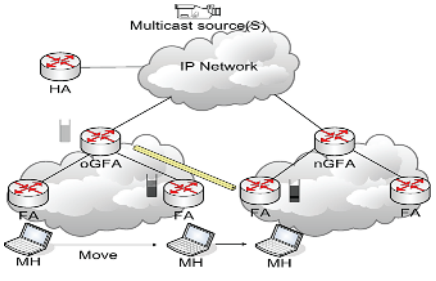


Fig. 3. Mechanism for proposed IP based mobile communication system

In the proposed architecture, the domain consists of one GFA (Gateway Foreign Agent) and several FAs. The mobile host will be connected to a FA and assigned CoA (Care of Address) and GCoA (Global CoA). With these two CoAs, the mobile host can distinguish the FA and GFA where it is attatched to. The GFA is the router that acts as a domain gateway. And the FA is a router that acts as the access router having a direct connection with the mobile host.

Since the proposed mechanism has a hierarchical structure, the largest registration delay for receiving the multicast service is incurred when a mobile host moves between domains. If there is already a member who is receiving the same multicast group service in the domain where a mobile host moves, the registration delay will be reduced. However, if there is no member, there is much delay time until joining multicast tree. In this case, the mobile host in new domain will send a Tunneling Start message that requests continuous multicast service to the oGFA (old GFA). When the oGFA receives a Tunneling Start message, it sends multicast data to the nFA in a new domain. The nFA then starts buffering. Therefore we can reduce the packet loss resulted from the registration delay

when the mobile hosts move to new domain. The details of intra or inter-domain handoff is described in the next section.

3.1 Intra-domain Handoff

In the case of intra-domain handoff, if the mobile host moves to the other FA, the mobile host performs L3 handoff before L2 handoff is completed. If the intra-domain handoff is predicted when the mobile host moves to the other FA, it sends a ProxySol (Proxy router Solicitation) message to the oFA and the oFA sends s ProxyRtAdv (Proxy Router Advertisement) message as a response. If the mobile host receives a ProxyRtAdv message, it sends a multicast Join message with registration request message to the nFA and sends a Buffer Start message to the GFA in order to start packet buffering of the multicast data.

When there is already any multicast group member in the nFA, the GFA sends the Buffer start message to the nFA in order to start the packet buffering for the mobile host. And after the mobile host completes the registration procedure with the nFA, the nFA sends buffered packets by unicast before sending newly incoming multicast data to the mobile host. Figure 4 shows intra-domain handoff procedure when there is multicast group member in the nFA.

When there is no multicast member in the nFA, the GFA that received the Buffer Start message starts a packet buffering for the mobile host. Because there is no multicast group member in the nFA, the nFA sends a Join message to the GFA. The GFA sends buffered packets by unicast before sending the newly incoming multicast data to the mobile host a with Join Ack. Figure 5 shows intra-domain handoff procedure when there is no multicast group member in the nFA.

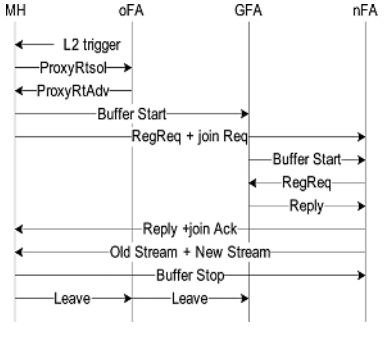


Fig. 4. Intra-domain handoff procedure when there is multicast group member in nFA

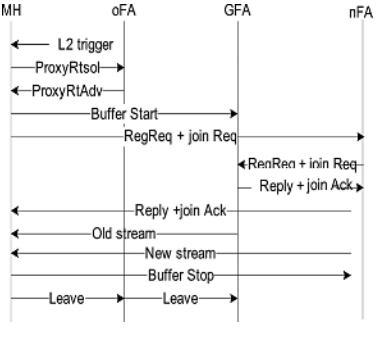


Fig. 5. Intra-domain handoff procedure when there is no multicast group member in nFA

3.2 Inter-domain Handoff

In the case of the Inter-domain handoff, the mobile host sends the Buffer Start message to the oGFA (old Gateway Foreign Agent). The oGFA sends the

Membership request message to the nGFA (New Gateway Foreign Agent) in order to verify if there is already multicast group member in the nGFA region. The nGFA confirms the own Group-list and sends the Membership response message to the oGFA.

When there is a multicast member in the nFA, the oGFA sends the Buffer start message to the nFA in order to start the packet buffering for the mobile host. And after the mobile host completes the registration procedure with the nFA, the nFA sends buffered packets by unicast before sending newly incoming multicast data to the mobile host. Figure 6 shows inter-domain handoff procedure when there is a multicast group member in the nFA.

When there is no multicast group member in the nFA, the oGFA starts the packet buffering and the mobile host sends the Tunneling Start message to the oGFA. The oGFA sends buffered packets by unicast before sending newly incoming multicast data to the mobile host. After the nFA joins multicast tree and receives multicast data, the mobile host sends a Tunneling Stop message to the oGFA, at this moment buffering is also stopped. Figure 7 shows inter-domain handoff procedure when there is no multicast group member in the nFA.

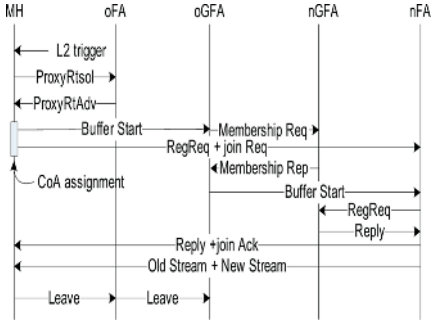


Fig. 6. Inter-domain handoff procedure when there is a multicast group member in nFA

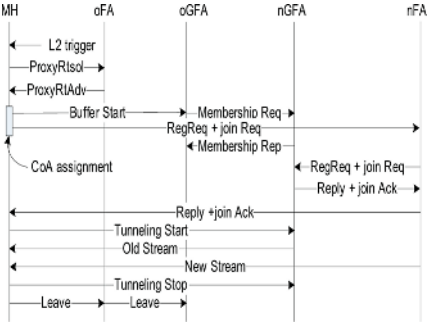


Fig. 7. Inter-domain handoff procedure when there is no multicast group member in nFA

4 Analysis

The simulation was performed using NS-2 simulator in order to verify that the proposed mechanism incurs less frequency of multicast tree reconstruction and produces less data loss when handoff occurs. Figure 8 shows the network topology used for the simulation. The mobile host moves from router1 to router4 along router2,router3 etc., and it reverses the direction moving from router4 to router1. When the mobile host arrives at each router, it joins the same multicast group and receives the same packet from source. We assume that sender of multicast data packet is static host while the receiver is mobile host. We also assume that the probability of existence of the same multicast group member in the new FA is 0.5.

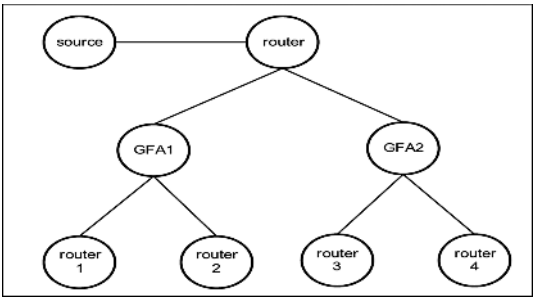


Fig. 8. Simulation topology

Figure 9 shows the number of multicast tree reconstructions. Bi-directional tunneling that is the HA based approach does not need to reconstruct the multicast tree, because HA is the end point of the multicast tree. On the other hand, MMA that is the FA based approach needs more multicast tree reconstructions than the proposed method, because it does not adopt the hierarchical structure. The number of Join message increased rapidly to reconstruct the multicast tree as the number of handoffs increases. The proposed multicast mechanism based on the hierarchical structure needs less Join message than MMA.

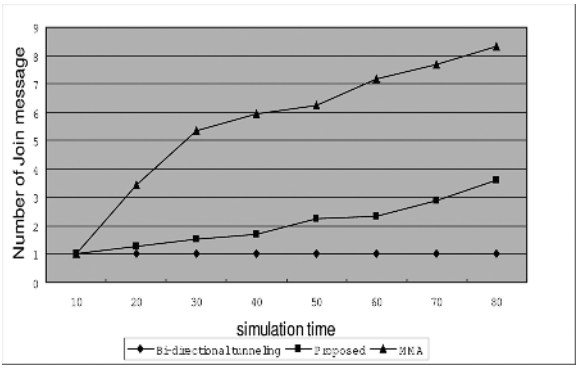


Fig. 9. Number of Join message according to number of handoffs

Figure 10 shows that the total number of packet loss increases as the number of handoffs increases. The proposed method yields the least number of packet loss because it adopts the buffering scheme. Even though MMA protocol does not use the buffering scheme, it yields less number of packet less than bi-directional tunneling, because it adopts the tunneling method during the handoff.

Figure 11 shows the number of the duplicated packets during the handoff. Although the proposed mechanism produces less packet loss, it produces more duplicated packets than MMA protocol. This is resulted from the fact that it

adopts the buffering scheme. Bi-directional tunneling does not produces the duplicated packets, because it does not use the buffering scheme. However, considering the results of figure 9 and 10, we can say that the amount of the decreased packet loss that results from buffering is more than the amount of duplicated packets. Therefore, we can conclude that packet buffering is needed in the mobile multicast environment during the handoff in order to guarantee the service quality.

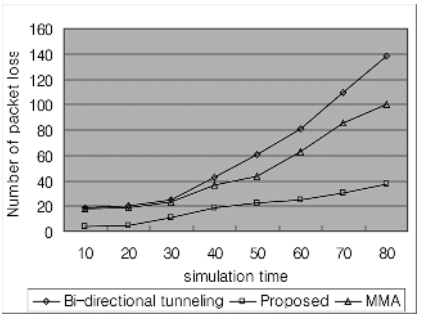


Fig. 10. Number of packet loss according to number of handoffs

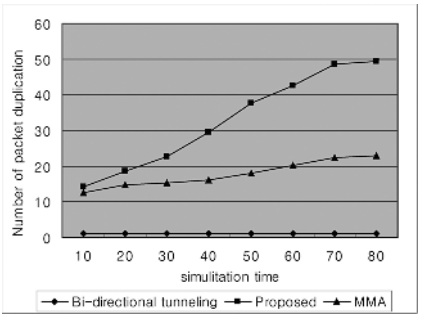


Fig. 11. Number of duplicated packet according to number of handoffs

Figure 12 and Figure 13 show the simulation results that traced the packet ID of the UDP traffic. UDP is an unreliable protocol, so it does not react to the packet loss and the out-of-sequence packets. Figures 12 and 13 show that the mobile host can receive more packets when there is a multicast group member in the nFA than if there is not a multicast group member in the nFA.

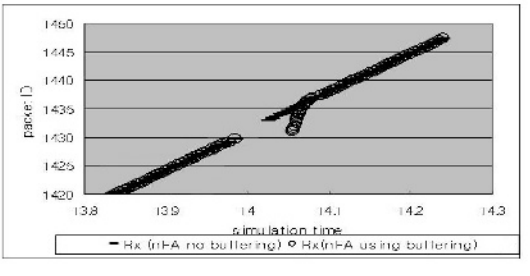


Fig. 12. Trace of packet ID when there is a multicast group member in nFA

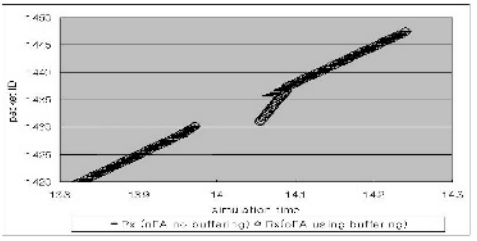


Fig. 13. Trace of packet ID when there is no multicast group member in nFA

5 Conclusion

In a mobile communication network environment, two approaches are progressing for an efficient multicast service. First method is to adopt the fast handoff during the handoff of the mobile host. Second method is to optimize the route by reducing the frequency of multicast tree reconstruction.

In this paper, we proposed the hybrid scheme for the reliable multicast routing in a mobile network environment. It is based on FA based hierarchical multicast routing method, in which the network is managed hierarchically to reduce the frequency of reconstruction of the multicast tree while guaranteeing the optimal route. Additionally, the proposed method tries to speedup the handoff by using the Mobile IPv4 Low latency handoff method by executing the L3 handoff before the L2 handoff is completed. And the adaptive buffering mechanism is adopted in a gateway or nFA (New Foreign Agent) to solve the multicast data loss. The simulation results show that the proposed mechanism provides good performance in terms of the frequency of reconstruction of the multicast tree, the number of packet loss and the number of received packets.

References

1. C. Perkins, IP Mobility Support, RFC 2002, Mobile IP Networking Group.
2. C. Perkins, Mobile IP Design Principles and Practices (Addison-Wesley).
3. G. Xylonmenos and G. Polyzos, IP multicast for mobile hosts, IEEE Communications Magazine (January 1997) 54-58.
4. V. Chikarmane and C.L. Williamson, Multicast support for mobile host using mobile IP: design issues and proposed architecture, Mobile Networks and Applications (1998) 365-379.
5. T. Harrison, C. Williamson, W. Mackrell and R. Bunt, Mobile multicast (MOM) protocol: multicast support for mobile hosts, in: Proc. Of ACM MOBICOM97 (1997) pp. 151-160.
6. Y.-J. Suh, H.-S. Shin and D.-H. Kwon, "Multicast Routing protocol by Multicast Agent in Mobile Networks" Proc.2000
7. K.Malki, "Low latency Handoffs in Mobile IPv4", draft-ietf-mobileip-lowlatency-handoffs-v4-09.txt

Fast IP Handover for Multimedia Services in Wireless Train Networks

Hee-Dong Park[1], Kang-Won Lee[2], Sung-Hyup Lee[2], You-Ze Cho[2], Yoon-Young An[3], and Do-Hyeon Kim[4]

[1] Department of Computer Engineering, Pohang College, Pohang, 791-711, Korea
hdpark@pohang.ac.kr
[2] School of Electrical Engineering & Computer Science, Kyungpook National University, Daegu, 702-701, Korea
[3] ETRI, 161 Gajeong-dong, Yuseong-gu, Daejeon, Korea
[4] Faculty of Telecommunication & Computer Engineering, Cheju National University, Jeju-do, 690-756, Korea

Abstract. This paper proposes a fast IP handover scheme for multimedia services in wireless train networks. This scheme uses the peculiar mobility characteristics of public vehicles such as trains and buses. Their moving pattern has a tendency to be predictable, because the moving path and direction are very routine. This enables a mobile router on the train to predict and to prepare the next IP layer handover before the link layer handover occurs, thereby the service disruption time due to handover will be reduced to the link layer handover latency. In order to perform the predictive IP handover, this scheme uses three mechanisms, such as predictive handover decision, predictive handover initiation, and predictive binding update. Analytical results showed that the proposed scheme can provide excellent performance in terms of service disruption time and packet loss ratio, compared with the existing IP handover scheme for mobile networks.

1 Introduction

Recently, various projects to support network mobility in public transportation have been in progress [1][2]. In the middle of them, the commuter train is regarded as one of the most suitable platforms to support broadband wireless Internet connectivity on terrestrial vehicles. Internet multimedia applications such as audio and video streaming, currently common in wired networks, will be demanded in this rapidly mobile environment. However, the current Mobile IP assumes relatively low speeds, which makes it very suitable for macro-mobility and nomadic environments. Therefore, high-speed express trains can reduce the effectiveness of the Mobile IP protocol and diminish the quality of its services. That is because the handover latency of the Mobile IP is too high to support their mobility and real-time multimedia services [3][4].

The NEMO basic support protocol suggests a bi-directional tunnel between a mobile router (MR) and its home agent (HA) [5]. The mechanism to maintain the

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 102–111, 2006.

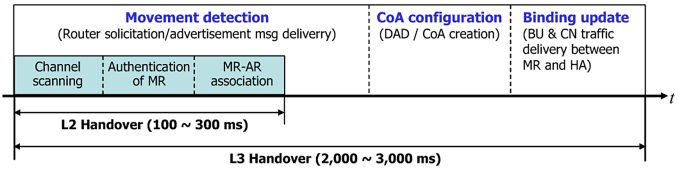


Fig. 1. Components of handover latency

MR-HA bi-directional tunnel is essentially the same as that of the MN (mobile node)-HA tunnel of Mobile IP. Therefore, when an MR moves into or out of a subnet, it suffers from the same handover problems as does an MN in the Mobile IP.

This paper proposes a fast handover scheme for mobile networks moving on the predetermined path such as trains and buses. In this scheme, each MR maintains a database about the list of access routers (ARs) and their network prefixes on the path. The MR therefore knows in advance network prefixes and its care-of addresses (CoAs) of all subnets on the moving path, without beacon signals from other subnets. This enables the MR to prepare the next IP layer (L3) handover through preregistration to its HA, before link layer (L2) handover occurs. Therefore, the service disruption time due to handover will be reduced to the L2 handover latency.

The rest of this paper is organized as follows: First, we review some of the related work in Section 2, then Section 3 presents the proposed scheme, and Section 4 evaluates the performance of the proposed scheme. Finally, the conclusion is given in Section 5.

2 Related Work

2.1 Handover Latency for Mobile Networks

The NEMO basic protocol will be built on Mobile IPv6 with minimal extensions. Therefore, as mentioned in Section 1, the handover mechanism of an MR is essentially the same as that of an MN in Mobile IP. The handover is classified into two components, L2 handover and L3 handover. Usually, the L3 handover is not dependent on the L2 handover, although it must precede the L3 handover.

Fig. 1 shows the components of handover latency in the NEMO basic solution. L2 handover involves channel scanning, authentication, and MR-AR association. The total L2 handover latency is about 100 to 300 msec. And L3 handover involves movement detection, new CoA configuration, and binding updates, which lead to about 2 to 3 seconds latency during an L3 handover.

Fig. 2 shows the message diagram of the IP layer handover for mobile networks based on Mobile IPv6 [6]. While an MR stays in an AR's coverage area, the MR receives periodic router advertisement messages from the AR. If the MR does not receive any messages from the AR during a predetermined time, it sends a router solicitation message to the AR to confirm its reachability. Nevertheless,

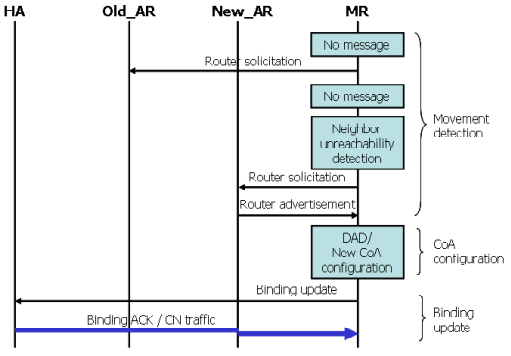


Fig. 2. IP layer handover message diagram

if the AR does not respond, the MR detects its unreachability to that AR and sends router solicitation messages to new ARs for re-association. If a new AR replies with a router advertisement message, the MR receives the network prefix information from the AR and forms an association with the new AR by creating a CoA. Then, the MR sends a binding update (BU) to its HA. After receiving the BU message, the HA can deliver data traffic from a correspondent node (CN) to the MR via the new AR. It is during this completion of the re-association process that the MR suffers from a handover delay.

2.2 Proactive Fast Handover Schemes

There are a number of proposals to reduce the latency and packet loss due to the handover. Among these proposals, we focus on proactive fast handover schemes performing the L3 handover before the L2 handover in advance.

Pre-registration handover scheme [7] and Predictive handover scheme [8] aim at low latency L3 handover based on L2 information through the use of L2 triggers. The L2 trigger indicates that an MN will soon be handed over. These schemes require additional functions in all existing ARs, increasing network deployment costs. Another limitation of these schemes is that they are not applicable in the network that does not have sufficient overlapping area between ARs.

Shim et al. introduced a fast handover scheme with neighborcasting to reduce the handover latency [9]. Each foreign agent (FA) in this scheme maintains a neighbor FA table where all neighboring FAs are recorded. Before L2 handover, the MN notifies the old FA to forward duplicated packets to all neighboring FAs without considering the MN's moving direction. The L3 handover latency in this scheme is reduced significantly. But, this scheme can cause unnecessary handover preparations and forward too many duplicated packets to all neighbor FAs.

Hsieh et al. proposed a seamless handover architecture for Mobile IP [10]. It builds on top of the hierarchical approach and the fast handover mechanism [8]

in conjunction with a new software-based movement tracking technique. This scheme successfully reduces L3 handover latency and packet loss, but it is centralized, requires extra signaling and imposes a bound on the speed of MNs.

Feng and Reeves proposed explicit proactive handoff with motion prediction for Mobile IP [11]. With the prediction of MN's motion, this scheme can reduce the handover latency significantly. But, this scheme also utilizes the L2 trigger mechanism to inform the MN of an impending L2 handover. For this, the MN should always measure received signal strength. Moreover, the MN uses the path prediction algorithm with the information in its movement history cache and pattern database.

2.3 Limitations of the L2 Trigger Mechanism

Generally, the above proactive handover schemes provide better performance, because the L3 handover is performed before the L2 handover by using L2 triggers. But, the L2 trigger mechanism involves some limitations as follows: First, the L2 trigger is based on fluctuating wireless channel states. Therefore, the handover anticipation using the L2 trigger may sometimes be incorrect. This incorrect anticipation may lead to unnecessary L3 handover, resulting in service disruption, packet loss, and unnecessary waste of buffer space. Second, the L2 trigger doesn't necessarily indicate L3 handover. That is, an L2 handover indication may or may not imply L2 movement, and L2 movement may or may not imply L3 movement. Therefore, unless it is well-known that an L2 handover indication is likely to imply L3 movement, instead of immediately multicasting a router solicitation it may be better to attempt to verify whether the default router is still bi-directionally reachable. Finally, using the L2 trigger diverges from the clean separation of layer 2 and layer 3. This means that the Mobile IP is not fully independent of the link layer. Given the wide diversity of wireless devices, it is difficult to define the operation and interaction of these radios in a global mobility aware network, without falling into link specific definitions. There is a need to define an open radio API that captures the essence of each wireless technology without exposing complex link specific details.

3 The Proposed Handover Scheme

This section describes a new proactive handover scheme for mobile networks moving on the predetermined path. In order to perform the predictive L3 handover without the L2 triggers, this scheme uses three mechanisms as follows: predictive handover decision, predictive binding update, and packer forwarding between neighboring ARs.

3.1 Predictive Handover Decision

In this scheme, each MR maintains a mobility database about the list of ARs, their network prefixes, and cell radii on the moving path. The MR therefore

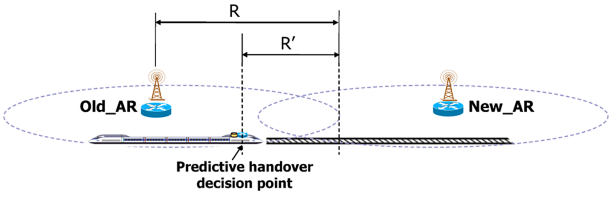


Fig. 3. Predictive handover decision point

knows in advance the network prefixes and CoAs of all subnets without beacon signals. That is, the MR can configure a next CoA in advance. With this database and the current location information, the MR can prepare L3 handover by pre-registration to its HA before L2 handover occurs. This makes total handover latency to be close to that of L2 handover.

In order to decide a handover execution time, the MR should recognize the predictive handover decision point, which may or may not be located in overlapping area between neighboring ARs, as shown in Fig. 3. The point can be indicated by sensors/GPS (global positioning system) or estimated by the MR. In the former case, the sensors can be laid at the predictive handover decision points along the moving path in order for the MR to sense the points. In the latter case, the MR estimates the points with the mobility database plus its current geographic location and moving speed. The MR can pinpoint its exact location with the aid of GPS or sensors laid on the side of railroad. When the MR reaches the predictive handover decision point, prior to entering the new AR's coverage area, it sends Predictive handover initiation (HI) and Predictive BU messages (described in subsection 3.2) to the old AR and to its HA, respectively. Unlike the proactive handover schemes described in subsection 2.2, this scheme does not utilize the L2 trigger mechanism. This allows a clean separation between layer 2 and layer 3 of the protocol stack. Whether or not the L2 trigger information is better than sensor or GPS information depends on system environments but does not depend on technical points.

3.2 Predictive HI and Predictive BU

As mentioned in section 3.1, when the MR reaches the predictive handover decision point, the MR sends a Predictive HI message to the old AR, in order to notify the old AR of an impending handover. After receiving the Predictive HI message containing the next CoA of the MR, the old AR starts forwarding data packets destined for the MR to the new AR. The forwarded packets are delivered to the MR after it moves to the new AR's coverage area.

With the predictive HI message, the MR sends a Predictive BU message to its HA through the old AR before entering the new AR's coverage area. The Predictive BU message has the same format as that of the general BU message, but it contains not the current CoA but the next CoA in the option field. If the MR cannot receive a Predictive binding ACK message from the HA, it

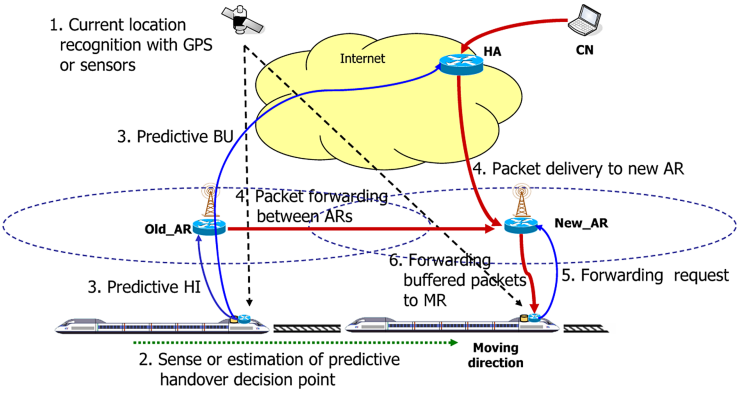


Fig. 4. Handover procedures

considers that the pre-registration proves to be a failure, then sends the general BU message again in the new AR's area. With the Predictive HI and BU, the proposed scheme performs the L3 handover before the L2 handover. Therefore, the total handover latency is close to that of the L2 handover.

3.3 Handover Procedures

Fig. 4 and 5 show the handover procedures and message diagram of the proposed scheme, respectively. The handover procedures are described as the following:

① The MR on a train should always recognize its current location with the aid of the GPS or sensors laid on the side of railroad.
② The MR senses or estimates the predictive handover decision point with the mobility database.
③ When the MR reaches the handover decision point, it sends a Predictive HI message to the old AR. At the same time, the MR sends a Predictive BU message to its HA through the old AR before the L2 handover. The two messages contain the next CoA commonly. In case that the predictive handover decision point is located in the overlapping area between neighboring ARs, the MR may confirm the reachability to the next AR.
④ when the old AR receives the Predictive HI message, it starts forwarding the data packets destined for the MR to the new AR, while the HA updates the binding and delivers data packets to the new AR. The new AR can buffer the packets to minimize packet loss.
⑤ As soon as the MR detects reachability to the next AR on the new link by solicited or unsolicited Router advertisement messages, it sends a Forwarding request message to the new AR.
⑥ When the new AR receives the Forwarding request message, it forwards the buffered data packets to the MR.

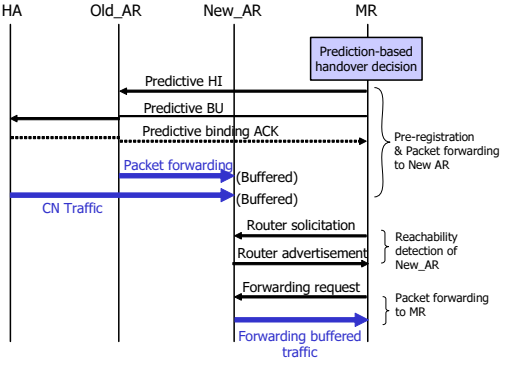


Fig. 5. Message diagram of the proposed scheme

Table 1. Parameter definitions

Parameters	Definition
T_{HO}	Total handover latency
T_{MD}	Time required for movement detection
$T_{CoA-Conf}$	Time required for CoA configuration
T_{BU}	Time required for BU
τ	Router advertisement interval
RTT_{MR-AR}	Round-trip time between MR and AR
RTT_{AR-HA}	Round-trip time between AR and HA

4 Performance Evaluation

This section analyzes and compares the performance of the proposed handover scheme and the existing IP layer handover used in the NEMO basic solution. Two critical performance issues are service disruption time and packet loss.

4.1 Service Disruption Time

Service disruption time during a handover can be defined as the time between the reception of the last packet through the old AR until the first packet is received through the new AR.

In this paper, we regard the service disruption time as the total handover latency, T_{HO}. As mentioned in Section 2, the handover of an MR involves L2 and L3 mobility. In the NEMO basic solution, the L2 handover must precede the L3 handover. Table 1 shows the parameters for performance evaluation.

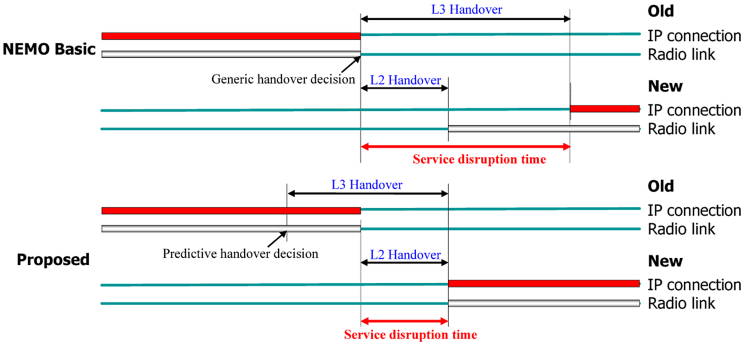


Fig. 6. Service disruption time during a handover

The total handover latency during a handover in the NEMO basic solution can be expressed as a sum of its components and with the signaling delay time shown in Fig. 2. This is given by:

$$\begin{aligned} T_{HO} &= T_{MD} + T_{CoA-conf} + T_{BU} \\ &= 2\,\tau + 2RTT_{MR-AR} + RTT_{AR-HA} \end{aligned} \tag{1}$$

where the delays for encapsulation, decapsulation, and the new CoA creation are not taken into consideration. Generally, the L3 movement detection delay, T_{MD}, includes the L2 handover latency.

As shown in Fig. 6, the total handover latency of the proposed scheme, however, will be close to T_{L2}, the L2 handover latency, because the MR performs the L3 handover before the L2 handover in advance, with keeping the reachability to the old AR. This makes the L3 handover latency to be minimized in the new ARs coverage area.

Fig. 7 compares the service disruption time between the proposed scheme and the NEMO basic scheme, according to RTT_{AR-HA}. We assume that T_{L2} is 200 msec, the router advertisement interval is 1 second, the radius of AR cell coverage is 1 Km, and RTT_{MR-AR} is 10 msec. As shown, the service disruption time of the NEMO basic solution is about 2 to 2.5 seconds, while the service disruption time of the proposed scheme is close to 200 msec. This means that the proposed scheme can support seamless network mobility.

4.2 Packet Loss Ratio

Since packet loss does not occur during the time when the CN traffic travels from the HA to an MR after the completion of the BU, the packet loss period during a handover can be expressed as T_{HO} - $0.5RTT_{MR-HA}$. Therefore, from (1) the packet loss period is given by:

$$T_{loss} = 2\,\tau + 1.5RTT_{MR-AR} + 0.5RTT_{AR-HA} \tag{2}$$

Also, the packet loss amount can be expressed as a product of the packet loss period and the bandwidth of the Internet:

$$L = T_{loss} * BW \tag{3}$$

where L represents the packet loss amount, and BW represents the bandwidth of the Internet. In the case of the proposed scheme, the packet loss time will be around T_{L2}. Nevertheless, there is no packet loss during a handover, due to the mechanism of packet forwarding between ARs and the new ARs buffering.

Packet loss ratio (ρ_{loss}) is defined as the ratio of the number of lost packets during a handover to the total numbers of transmission packets in a cell. This can be also expressed as:

$$\rho_{loss} = \frac{T_{loss}}{T_{cell}} \times 100 \quad (\%) \tag{4}$$

where T_{cell} is the time it takes an MR to pass through a cell.

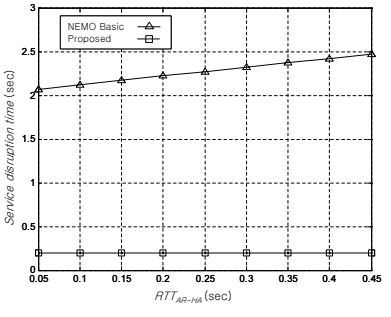


Fig. 7. Service disruption time

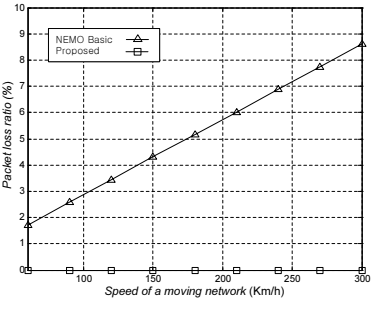


Fig. 8. Packet loss ratio

Fig. 8 shows the packet loss ratio according to the speed of a moving network. In the figure, RTT_{AR-HA} is assumed to be 100 msec. As shown, the packet loss ratio of the NEMO basic solution is proportional to the speed of a moving network, while the packet loss ratio of the proposed scheme will be constantly zero when the new AR buffers data sent from HA.

5 Conclusion

This paper proposed a new fast handover scheme for public transportation such as trains and buses. This scheme uses the predictable mobility characteristics of them. Therefore, the MR can configure in advance the next CoA and register it with its HA before the L2 handover, so the total handover latency and packet loss can be reduced significantly. The additional advantages of the proposed scheme, in comparison with the existing proactive handover schemes, are as follows: First,

the proposed scheme does not use the L2 trigger mechanism, which can makes it fully independent of the link layer. Second, the proposed scheme is applicable in non-overlapping networks as well as overlapping networks. Finally, the proposed scheme does not require any complicated motion prediction algorithm, because it uses the peculiar characteristics of the public transportation. The overhead of the proposed scheme, in comparison with the NEMO basic support, involves the cost to maintain mobility database, the cost of additional signaling messages, and buffering.

Acknowledgment

This work was supported in part by the KOSEF (contract no.: R01-2003-000-10155-0) and the ITRC of the Ministry of Information and Communication (MIC), Korea.

References

1. MORANE (Mobile Radio for railway Networks in Europe), http://gsm-r.uic.asso.fr/morane.html.
2. T. Ernst, K. Mitsuya, and K. Uehara, "Network Mobitity from the InternetCar Perspective," $Journal of Interconnection Networks (JOIN)$, June 2003.
3. C. Perkins, Ed., "IP Mobility Support for IPv4," $IETF\ RFC$ 3344, Aug. 2002.
4. D. Johnson et al., "Mobility Support in IPv6," $IETF\ RFC$ 3775, June 2004.
5. V. Devarapalli "Nemo Basic Support Protocol," $IETF\ RFC$ 3963, Jan. 2005.
6. E. K. Paik and Y. H. Choi, "Prediction-Based Fast Handoff for Mobile WLANs," in *Proc. of ICT*, vol. 1, pp. 748-753, Feb. 2003.
7. K. Malki, Ed., "Low Latency Handoffs in Mobile IPv4," *Internet Draft*, draft-ietf-mobileiplowlatency-handoffs-v4-09.txt, June 2004.
8. R. Koodli, Ed., "Fast Handovers for Mobile IPv6," *Internet Draft*, draft-ietf-mipshop-fastmipv6-03.txt, Oct. 2004.
9. E. Shim et al., "Low Latency Handoff for Wireless IP QoS with Neighborcasting," in *Proc. ICC* 2002, Apr. 2002.
10. R. Hsieh et al., "S-MIP: A Seamless Handoff Architecture for Mobile IP," in *Proc. INFOCOM* 2003, Mar. 2003.
11. F. Feng and D. Reeves, "Explicit Proactive Handoff with Motion Prediction for Mobile IP," in *Proc. of WCNC* 2004, vol. 2, IEEE, pp. 855-860, Mar. 2004.

Hierarchical Synchronized Multimedia Multicast for Mobile Hosts in Heterogeneous Wireless Networks

Ing-Chau Chang[1] and Chih-Sung Hsieh[2]

[1] Department of Computer Science and Information Engineering
National Changhua University of Education, Changhua, Taiwan, R.O.C.
icchang@cc.ncue.edu.tw
[2] Institute of Networking and Communication Engineering
Chaoyang University of Technology, Taichung, Taiwan, R.O.C.
s9330602@cyut.edu.tw

Abstract. For supporting handoff mobile users on heterogeneous wireless networks to synchronously receive and play out multicast multimedia stream data, we propose a two-layer Hierarchical Synchronized Multimedia Multicast (HSMM) architecture to enhance the single-layer Synchronized Multimedia Multicast (SMM) [1]. In HSMM, each wireless network operator can adapt its own management mechanism, such as routing protocol, access control, etc., and further define the range of Guarantee Region (GR) to satisfy different management requirements. Compared to SMM and the traditional Remote Subscription (RS) protocol, HSMM will significantly reduce total amounts of synchronization buffer of foreign agents, join latency and buffer replenishment time of mobile users, and finally achieve a better playback quality.

1 Introduction

In recent years, different kinds of wireless networks such as WLAN, GSM, GPRS, 3G cellular network are proliferated for public use, which is gradually formed a heterogeneous wireless environment. It is claimed that the 4G network [8] will build an integrated network among backbone Internet and these different wireless networks. How to support IP multicasting for mobile users in the forthcoming 4G network will be a great challenge [9]. In this paper, based on the single-layer Synchronized Mobile Multicast (SMM) scheme [1], which integrates with Mobile IP [2], CBT v2 [3] for IP Multicasting [4-5] and MPEG-4 fine granularity scalability (FGS) compression technique [6], we propose the two-layer Hierarchical Synchronized Multimedia Multicast (HSMM) scheme to achieve synchronized multimedia multicast through heterogeneous 4G networks and provide seamless playback of continuous media streams for the mobile receivers (MR) with bounded buffers and join/initial latencies in the handoff Guarantee Region (GR), even when the mobile sender (MS) and MR, hands over to wireless cells within the current multicast tree or not for infinite times, which are advantages that the traditional Home Subscription (HS) and Remote Subscription (RS) schemes [7] cannot support.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 112 – 121, 2006.

This paper is organized as follows. In section 2, the HSMM system architecture and its three operation phases are described. We analyze buffer requirements and join/initial latencies for the mobile to continuously receive multimedia data with SMM and HSMM. Simulations in section 3 exhibit that HSMM is more efficient than SMM to support continuous playback with guaranteed QoS when the MH hands over between different wireless networks. Finally, section 4 concludes this paper.

2 HSMM Architecture and Operations

2.1 HSMM Architecture

The HSMM architecture is shown in Figure 1. We have made following assumptions: (1) each layer 2 wireless network, like WLAN, 3G, can be managed by different network operator to execute SMM inside it and extend capabilities of the Gateway Router (GW) of WLAN or the GGSN of GPRS/3G to work as its own layer 2 CR which further interconnects with layer 1 Internet Backbone. (2) each GW or GGSN is a multicast-capable router (MCR). It can run as a layer 1 FA and a layer 2 CR simultaneously. (3) end-to-end delay (EED) between layer 2 FA and MH is the same.

In the HSMM architecture, layer 1 GR (GR^1) is defined as the range from layer 1 Core Router (CR^1), via layer 1 Foreign Agent (FA_i^1), i.e., the CR_i^2 of layer 2 GR_i^2, which is the GW/GGSN of layer 2 wireless network *i*, to the farthest MR with maximal EED $Max(D_{CR\_MR}^1)$. Similarly, each layer 2 GR for wireless network *i* (GR_i^2) is defined as the range from its layer 2 CR (CR_i^2), via layer 2 FA *j* of wireless network *i* ($FA_{i,j}^2$), to the farthest MR with maximal EED $Max(D_{CR\_MR}^2)$. For overcoming effects of the versatile EED suffered by the MR when handing over different layer 2 wireless networks, the HSMM scheme controls transmission of layer 1 CR^1 and FA_i^1 such that each layer 1 FA_i^1 can synchronously multicast the same multimedia data to its underlying GR_i^2, which in turn synchronously multicasts media data to all $FA_{i,j}^2$ and finally to MRs currently inside this layer 2 wireless network as the SMM does. In this way, all MRs can play out the same clip of media data simultaneously and continuously, no matter they hand over horizontally within the same wireless network, i.e., horizontal handoff, or vertically between two different ones, i.e., vertical handoff. Instead of adapting the original SMM scheme on this heterogeneous wireless environment by multicasting media data from the backbone CR, i.e., CR^1 in HSMM, to the corresponding FA of the MR, i.e., $FA_{i,j}^2$ in HSMM, through the SMM CBT multicast tree which regards GWs and GGSNs of wireless networks as original MCRs, our 2-layer HSMM architecture has following significant merits:

(1) each layer 2 network operator can define its own GR size to meet its policies.
(2) because buffer sizes of CR, FA and MR are directly proportional to the maximal EED, i.e., $Max(D_{cr\_mh})$, of the GR, the 2-layer HSMM can significantly reduce buffer sizes due to much smaller layer 2 GRs than the single huge GR of SMM.

(3) with the 2-layer hierarchy of the HSMM, except the first multicast group member, other group members only need to join local layer 2 multicast tree, which greatly reduces the MR's *join latency (JL)* to continue playback.

(4) if the MR horizontally hands over within the same layer 2 GR_i^2, it has to replenish its buffer locally with HSMM, instead of replenishing from CR^1 with SMM. The buffer replenishment time with HSMM is reduced significantly.

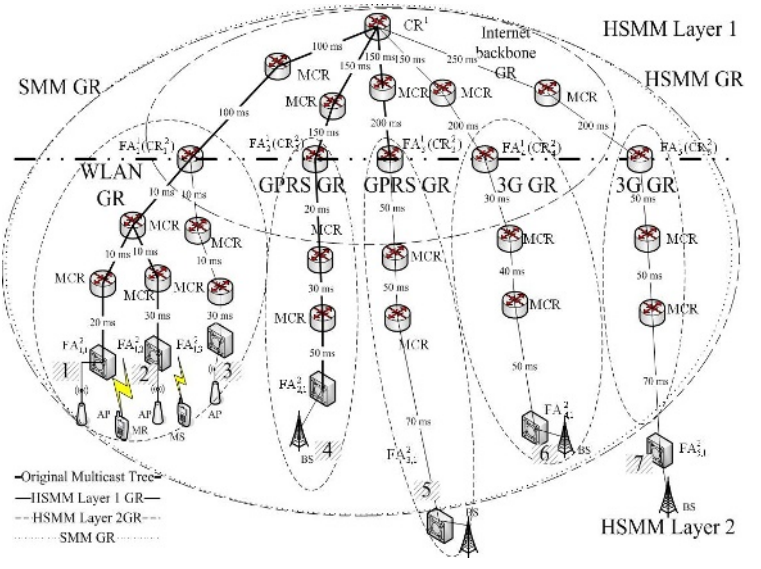


Fig. 1. HSMM Architecture

2.2 HSMM Operation Phases

HSMM is composed of the Join phase, Multicast phase and Handoff phase. Detail operations of these phases please refer to [1].

2.2.1 Join Phase

1. The MR sends an *IGMP Membership Report* to its $FA_{i,j}^2$ to join a multicast group.
2. Whenever the $FA_{i,j}^2$ receives IGMP Membership Report, it first checks whether other group members in the same $FA_{i,j}^2$ has joined the same multicast group. If not, it sends *CBT JOIN_REQUEST (CBT_JR)* along the shortest path to the CR_i^2. As soon as the $FA_{i,j}^2$ receives the *CBT JOIN_ACK (CBT_JA)* from the CR_i^2 or an intermediate MRT, it calculates the EED ($D^2_{CR_i\_FA_{i,j}}$) between it and its CR_i^2. At the same time if this CR_i^2 has not joined layer 1 CBT multicast tree, it acts as a layer 1 FA_i^1 to forward *CBT_JR* to CR^1 and calculates the layer 1 EED ($D^1_{CR\_FA_i}$) between FA_i^1 and CR^1. After that, the MR has joined layer 1 and 2 multicast trees.

2.2.2 Multicast Phase

Figure 2 illustrates the HSMM multicast flow. The MS is located within a layer 2 GR_i^2. It unicasts media data to its CR_i^2, then to CR^1 and finally multicasts to all MRs through layer 1 and layer 2 CBT multicast trees at normal playback rate of the MR.

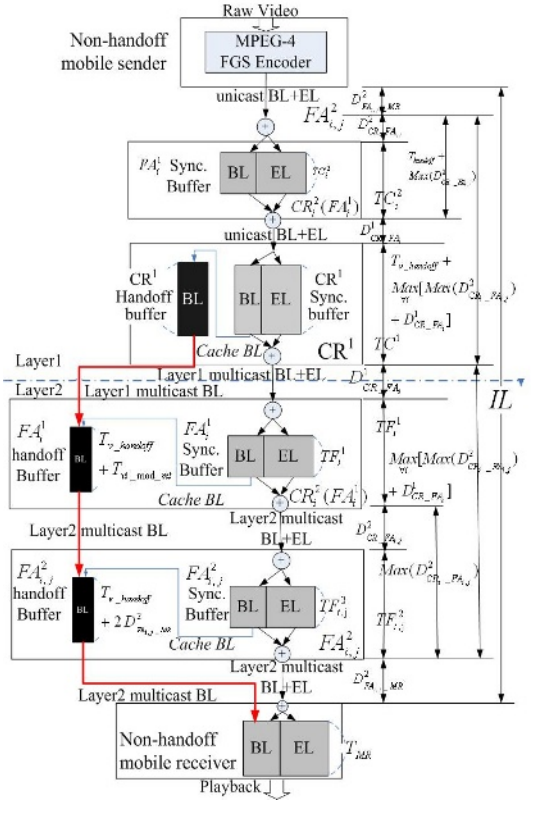


Fig. 2. HSMM multicast flow

(1) **From the MS to the** CR^1**:** To guarantee the MR can continuously receive the multimedia data as the MS hands over to another cell, the HSMM scheme allocates the "*core router synchronization buffer (CRSB)*" in the CR^1 and CR_i^2 to cache the initial data sent from the MS, instead of multicasting them to all the MRs immediately. If the MS hands over in the same layer 2 GR_i^2, the CR_i^2 CRSB must be large enough to cache two parts of media data: one is the missed data for the MS horizontal handoff duration, i.e., $T_{handoff}$; the other is the data for the maximal EED difference, i.e., $Max(D^2_{CR_i\_FA_{i,j}}) - D^2_{CR_i\_FA_{i,j}}$, between current cell and the farthest one in the layer 2 GR_i^2. Total time to cache data in the CR_i^2 SB is formulated with Equation 1. If the MS hands over across two different layer 2 GRs, the CR^1 CRSB must be large enough to cache

two parts of media data: one is the missed data for the MS vetical handoff duration, i.e., $T_{v\_handoff}$; the other is the data for the maximal layer 1 EED difference, i.e., $\underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] - Max(D^2_{CR_k\_FA_{k,j}}) - D^1_{CR\_FA_k} = Max(D^1_{CR\_MR}) - D^2_{FA_{i,j}\_MR} - Max(D^2_{CR_k\_FA_{k,j}}) - D^1_{CR\_FA_k}$, between the current cell in GR^2_k and the farthest one in the layer 1 GR. Total time to cache data in the CR^1SB is formulated with Equation 2.

$$TC^2_i = T_{handoff} + Max(D^2_{CR_i\_FA_{i,j}}) - D^2_{CR_i\_FA_{i,j}} \tag{1}$$

$$TC^1 = T_{v\_handoff} + \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] - Max(D^2_{CR_k\_FA_{k,j}}) - D^1_{CR\_FA_k} \tag{2}$$

(2) **From** CR^1 **to** FA^1_i (CR^2_i) **and from** CR^2_i **to** $FA^2_{i,j}$**:** For avoiding the playback interruption that results from different EEDs before and after the MR hands over, the HSMM scheme allocates the "*FA synchronization buffer (FASB)*" in the FA^1_i and $FA^2_{i,j}$ of the MR to cache the multicast data sent from the CR^1 and FA^1_i (CR^2_i), respectively. Based on SMM, all $FA^2_{i,j}$ s in layer 2 GR^2_i will send data to MRs at time $Max(D^2_{CR_i\_FA_{i,j}})$ after its CR^2_i receives the first data from layer 1 multicast tree. The $FA^2_{i,j}$ has to cache data for the $TF^2_{i,j}$ duration by Equation 3. HSMM controls all layer 2 FAs in all GRs to send media data to MRs at the same time. It is equal to the size of HSMM GR ($\underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}]$), i.e., the maximal total delays which consist of propagation delay from layer 1 CR^1 to FA^1_i (CR^2_i), i.e., $D^1_{CR\_FA_i}$, cache delay of FA^1_i (TF^1_i), propagation delay from layer 2 CR^2_i to $FA^2_{i,j}$, i.e., $D^2_{CR_i\_FA_{i,j}}$, and the cache delay of $FA^2_{i,j}$ ($TF^2_{i,j}$) for all GR *i*. TF^1_i of FA^1_i is calculated by Equation 4.

$$TF^2_{i,j} = Max(D^2_{CR_i\_FA_{i,j}}) - D^2_{CR_i\_FA_{i,j}} \tag{3}$$

$$TF^1_i = \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] - Max(D^2_{CR_i\_FA_{i,j}}) - D^1_{CR\_FA_i} \tag{4}$$

2.2.3 Handoff Phase

These buffers on the CR^1, FA^1_i (CR^2_i), $FA^2_{i,j}$ and MR are empty after the first handoff and multimedia playback will be interrupted when the handoff occurs again. For supporting infinite times of the MS or MR handoffs, the HSMM scheme replenishes these buffers with minimal extra bandwidth by employing the MPEG-4 *fine granularity scalability (FGS)* approach to encode the video into the *base layer (BL)* and *enhancement layer (EL)* streams. The BL stream can be decoded alone to show a video with the poorer quality. Oppositely, the EL stream is only used to combine with the BL to enhance the quality. The MS and MRs have five types of handoffs, which corresponding buffer replenishment processes are discussed below.

1. Intra-network horizontal handoff, new FA in Layer2 multicast tree

In this case, the handoff MR only connects to the new $FA^2_{i,j}$ without modifying the layer 2 multicast tree. Based on SMM, the HSMM scheme allocates the *FA handoff buffer (FAHB)* to cache one copy of the missed BL data for the $(T_{handoff} + 2\times D^2_{FA_{i,j}\_MR})$ duration as soon as the $FA^2_{i,j}$ starts to multicast media data from the FASB to its MRs when the FASB is full. After the MR finishes its handoff operations, its new $FA^2_{i,j}$ continues its multicasting. For replenishing buffers of all handoff MRs with the missed data, the HSMM allocates extra (BL+EL) wireless bandwidth for local multicasting the cached BL data in the FAHB to concatenate the BL data left in the handoff MR's buffer for the $\sum_{k=1}^{\infty}\left[\left(T_{handoff} + 2\times D^2_{FA^2_{i,j}\_MR}\right)\times Q^k\right] = \left(T_{handoff} + 2\times D^2_{FA^2_{i,j}\_MR}\right)\times\frac{Q}{1-Q}$ duration, where Q denotes the quotient of the BL bandwidth over the (BL+EL) one. The MR after its handoff suffers poorer BL video quality for the $\left(T_{handoff} + 2\times D^2_{FA^2_{i,j}\_MR}\right)\times\frac{1}{1-Q}$ duration.

2. Intra-network horizontal handoff, new FA not in Layer2 multicast tree

At the worse case, the new $FA^2_{i,j}$ has to join the layer 2 multicast tree by issuing CBT_JR all the way to the CR^2_i and waits for CBT_JA if no crossover MRT already in the layer 2 multicast tree. The MR spends two times of the maximal GR^2_i EED, i.e., $[\,Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR}]$, plus total delays, i.e., T_{mt}, for all intermediate MCRs to process CBT messages, which is formulated in Equation 5. In this case, the missed data during the MR handoff cannot be replenished from the new $FA^2_{i,j}$ such that HSMM allocates the *core router handoff buffer (CRHB)* in the CR^2_i to cache one copy of the BL data when the CR^2_i multicasts media data to other FAs from its CRSB for the $(T_{handoff} + T_{mod\_mt})$ duration. As the MR's CBT_JR is reached the CR^2_i, the CR^2_i simultaneously multicasts the new media data to all FAs and multicasts the cached BL data in the CRHB to the new $FA^2_{i,j}$ FAHB with the extra (BL+EL) bandwidth and immediately pipelines to the MR's buffer with local multicast bandwidth. After that, the new FA follows the operations mentioned in case 1 to replenish the MR's buffer.

$$T_{mod\_mt} = 2\times[\,Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR}] + T_{mt} \tag{5}$$

3. Inter-network vertical handoff, new GR in Layer1 multicast tree and new FA in Layer2 multicast tree

In this case, HSMM works the same as case 1, except the size of $FA^2_{i,j}$ handoff buffer must be able to cache one copy of the BL data for the $(T_{v\_handoff} + 2\times D^2_{FA_{i,j}\_MR})$ duration and replenish the vertical handoff MR's buffer for the $\sum_{k=1}^{\infty}\left[\left(T_{v\_handoff} + 2\times D^2_{FA^2_{i,j}\_MR}\right)\times Q^k\right] = \left(T_{v\_handoff} + 2\times D^2_{FA^2_{i,j}\_MR}\right)\times\frac{Q}{1-Q}$ duration.

4. Inter-network vertical handoff, new GR in Layer1 multicast tree and new FA not in Layer2 multicast tree

This case is similar to case 2. At worst, the MR spends two times of the maximal EED among all layer 2 GRs, i.e., ($\underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR}]$), plus total delays, i.e., T_{mt}, for all intermediate MCRs to process CBT messages, which is formulated in Equation 6. The MR buffer replenishment process in this case is similar to that of case 2 except that the CRHB in the new CR_i^2 has to cache one copy of the BL data for the ($T_{v\_handoff} + T_{vi\_mod\_mt}$) duration.

$$T_{vi\_mod\_mt} = 2 \times \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR}] + T_{mt} \tag{6}$$

5. Inter-network vertical handoff, new GR not in Layer1 multicast tree

After the MR hands off vertically to a new $FA^2_{i,\mathrm{j}}$ of a new GR_i^2 not in layer 1 multicast tree, HSMM has to spend $T_{vo\_mod\_mt}$ to modify both the layer 1 and layer2 multicast trees in the layer 1 GR^1 and this new layer 2 GR_i^2 by sending CBT_JR all the way to the CR^1, via the new CR_i^2 in the GR_i^2, with the maximal layer 1 EED and waiting for CBT_JA in the worst case. The MR spends two times of the maximal EED in layer 1 GR, i.e., ($\underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR} + D^1_{CR^1\_FA^1_i}]$), plus total delays, i.e., T_{v_mt}, for all intermediate MCRs in layer 1 and layer 2 multicast trees to process CBT messages, which is formulated in Equation 7.

$$T_{vo\_mod\_mt} = T_{v\_mt} + 2 \times \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR} + D^1_{CR^1\_FA^1_i}] \tag{7}$$

In this case, the missed data during the MR handoff cannot be replenished from the new $FA^2_{i,\mathrm{j}}$ and the new FA_i^1 (CR_i^2) such that HSMM allocates the *core router handoff buffer (CRHB)* in the CR^1 to cache one copy of the BL data when the CR^1 multicasts media data to other FA_i^1s from its CRSB for ($T_{v\_handoff} + T_{vo\_mod\_mt}$) duration. After the FA_i^1 (CR_i^2) has buffered for the TF_i^1 duration, it begins to multicast the media data to all $FA^2_{i,\mathrm{j}}$ in this new GR_i^2 through its layer 2 multicast tree and caches one copy of sent data in its FA_i^1 HB. As soon as the $FA^2_{i,\mathrm{j}}$ receives the cached BL data which is first from the CR^1 through extra layer 1 multicast tree to the CR_i^2 and then from the CR_i^2 through extra layer 2 multicast tree to it, it immediately pipelines the BL data to MR's buffer with local multicast bandwidth, which is shown in Figure 3.

At the worst case, when the MR suffers the case 5 handoff, the MR must wait for ($T_{v\_handoff} + T_{vo\_mod\_mt}$), which is called the *Join Latency (JL)* in Equation 8, to resume receiving the media data such that HSMM must allocate buffer to the MR for the T_{MR} = [($T_{v\_handoff} + T_{vo\_mod\_mt}$)+ 1GOP] duration, assuming the video player of the MR has to buffer at least one MPEG *Group of Picture (GOP)* data to decode the video data and start the playback. As shown in Figure 2, the HSMM *Initial Latency (IL)* from the MS

begins to send the first media data to the CR^1, through FA_i^1 and $FA_{i,j}^2$, until the MR starts the playback can be calculated by Equation 9.

$$JL = T_{v\_handoff} + T_{vo\_mod\_mt}$$

$$= 2\times \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^2_{FA_{i,j}\_MR} + D^1_{CR^1\_FA_i^1}] + T_{v\_mt} \quad (8)$$

$$IL = T_{v\_handoff} + \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] + \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] + D^2_{FA_{i,j}\_MR} + T_{MR}$$

$$= 2\times T_{v\_handoff} + 4\times \underset{\forall i}{Max}[(Max(D^2_{CR_i\_FA_{i,j}}) + D^1_{CR\_FA_i}] + 3\times D^2_{FA_{i,j}\_MR} + T_{v\_mt} + 1GOP \quad (9)$$

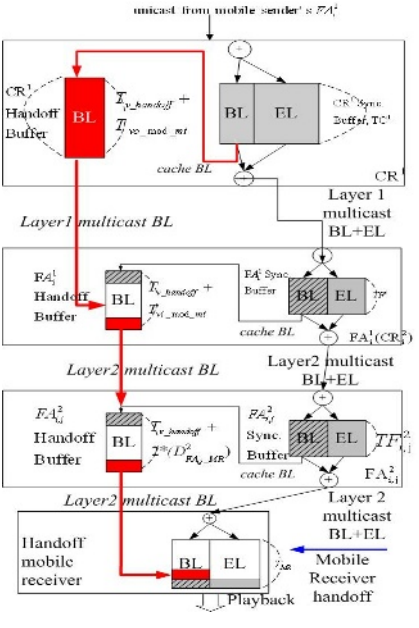


Fig. 3. Buffer replenishment of HSMM Inter-Network vertical handoff, new GR_i^2 not in Layer1 multicast tree

3 Simulation Results

Because the RS scheme was shown to has lower average loss ratio and playback interruption ratio than HS in [1], we will compare the multimedia reception and playback behaviors of the MR achieved by the RS, SMM and HSMM schemes here. The simulation environment and propagation delays between two adjacent nodes are shown in Figure 1. Queuing/processing delays of these nodes and T_{mt} / T_{v_mt} are assumed zero. The horizontal handoff delay, $T_{handoff}$, and the vertical handoff delay, $T_{v\_handoff}$, are

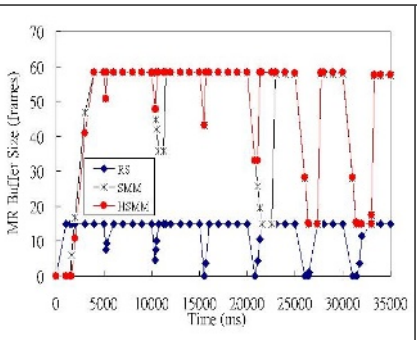


Fig. 4. Buffer size of RS, SMM and HSMM

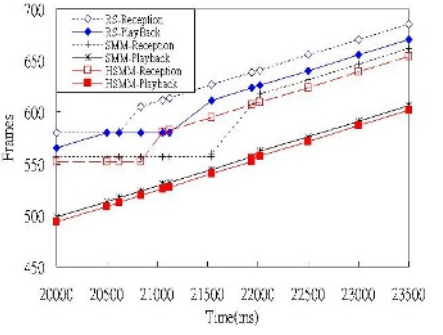


Fig. 5. MR reception and playback for RS, SMM and HSMM at 20000ms

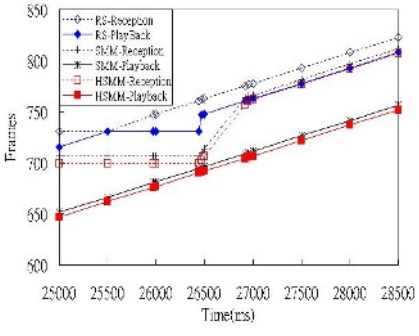


Fig. 6. MR reception and playback for RS, SMM and HSMM at 25000ms

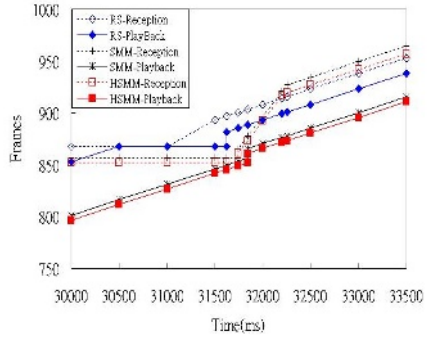


Fig. 7. MR reception and playback for RS, SMM and HSMM at 30000ms

assumed to be 250ms and 500ms respectively. The HSMM GR^1 and SMM GR are the same size, which is equal to 470ms. The five GR_i^2 sizes from left to right in Figure 1 are 70, 100, 120, 120 and 150ms, respectively. When the multicast session begins, the MS will multicast a FGS-encoded MEPG-4 video stream, which is 30 frames per second, to all MRs in the multicast group. Bandwidth requirements for the base layer and enhancement layer are 256 Kbps and 768 Kbps respectively such that Q=(256)/(256+768)=0.25. Further, the MR has to buffer at least one group of picture (GOP) video data for later playback, which is assumed to be 500ms for the common GOP sequence.

The simulation scenario is as follows. (1) The MS multicasts the video stream at time 0. (2) At 5000ms, the MR hands over from Location 1to Location 2 in Figure 1, which is an Intra-network horizontal handoff, new FA in layer 2 multicast tree. (3) At 10000 ms, the MR hands over from Location 2 to 3, which is an Intra-network horizontal handoff, new FA not in Layer2 multicast tree. (4) At 15000 ms, the MR hands over from Location 3 to 4, which is an Inter-network vertical handoff, new GR in Layer1 multicast tree and new FA in Layer2 multicast tree. (5) At 20000 ms, the MR hands over from Location 4 to 5, which is an Inter-network vertical handoff, new GR

in Layer1 multicast tree and new FA not in Layer2 multicast tree. New FA is out of layer 1 GR. (6) At 25000 ms, the MR hands over from Location 5 to 6, which is an Inter-network vertical handoff, new GR not in Layer1 multicast tree. (7) At 30000 ms, the MR hands over from Location 6 to 7, which is an Inter-network vertical handoff, new GR not in Layer1 multicast tree. New FA is out of layer 1 GR.

In Figure 4, MR of RS stops receiving data during handoff and requires to refill the MR buffer to a GOP (15 frames) after handoff for continuous playback, which results in significant playback interruption (loss and delay) for RS and is shown in Figures 5, 6, and 7. However, because layer2 CRs after MR handoffs are already in the layer 1 multicast tree at time 10000ms and 20000ms, the HSMM only needs to replenish the MR's buffer from layer2 CRs with less amounts of BL frames than SMM does from the CR at its tree root, which results in better playback quality with (BL+EL) frames and is shown in Figure 5 at 20000ms. However, if the layer 2 CR after handoff has not yet in the layer 1 tree (at 25000ms and 30000ms), HSMM and SMM need to replenish same amount of frames from the layer 1 CR, resulting in both of them stop receiving for same durations,. which is shown in Figures 6 and 7. Further, at 20000ms in Figure 5, because MR moves out of the Layer1 GR for 50ms, i.e., [(150+200)+(50+50+70)]-470, both SMM and HSMM fail to replenish frames needed for the extra 100ms(=2×50ms) such that both of them suffer from the same number of frame loss. Similarly, both SMM and HSMM lose 9 (=2×[((250+200)+(50+50+70))-470]/1000×30) frames at 30000ms in Figure 7.

4 Conclusions

In this paper, we have proposed a two-layer HSMM architecture under 4G heterogeneous wireless network environment to support synchronized multicast with less amounts of buffer, higher quality media playback and faster buffer replenishment for infinite numbers of handoff than SMM does.

References

1. Chang I.C., Huang K.S.: Synchronized Mobile Multicast Support for Real-Time Multimedia Services, IEICE Trans. on Communications, Vol.E87-B No.9 (2004) 2585-2595
2. Perkins C.E.: Mobile Networking Through Mobile IP, IEEE Internet Computing, Vol.2, (1998) 58-69 Jan./Feb.
3. Ballardie A.: Core Based Trees (CBT version 2) Multicast Routing, RFC 2189, Sep. (1997)
4. Tan C.L., Pink S.: Mobicast : A Multicast Scheme for Wireless Network, Mobile Networks and Applications, Baltxer, Vol. 5, (2000) 259-271, Dec.
5. Benslimane A.: Multimedia Multicast in Mobile Computing, IEEE International Symposium of Multimedia Software Engineering, (2000) 339 –346
6. Radha H.M., et al.: The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming Over IP, IEEE Transactions on Multimedia, Vol. 3, No. 1, Mar. (2001)
7. Perkins C.E.: IP Mobility Support for IPv4, IETF RFC 3220, Jan. (2002)
8. Niebert N., et al.: Ambient Networks : An Architecture for Communication Networks Beyond 3G , IEEE Wireless Communications, (2004) 14-22, Apr.
9. Varshney U., Jain R.: Issues in Emerging 4G Wireless Networks, IEEE Computer, (2001) 94-96, Jun.

Control Parameter Setting of IEEE 802.11e for Proportional Throughput Differentiation

Seung-Jun Lee, Chunsoo Ahn, and Jitae Shin

School of Information and Communication Engineering, Sungkyunkwan University, Suwon, 440-746, Korea
{lsj6467, navy12, jtshin}@ece.skku.ac.kr

Abstract. IEEE 802.11e has been heavily researched with regard to support the Quality of Service (QoS) in wireless LAN environments. The Enhanced Distributed Coordination Function (EDCF) mechanism of IEEE 802.11e has several control parameters for QoS provision, such as minimum/maximum contention window (CW_{min}/CW_{max}), retry limit, Arbitration Inter Frame Space ($AIFS$), and so on. Through varying of these parameters, differentiated services can be provided to different priority classes, in terms of throughput, packet loss rate, and delay. The parameters of the IEEE 802.11e EDCF mechanism that have the greatest effect on QoS are investigated in this paper. Especially, Proportional Throughput Differentiation Service (PTDS) is proposed through mathematical analysis for proportional service differentiation between adjacent priority classes and proper setting of control parameters. Finally, the minimum contention window size is confirmed to have the greatest effect with regard to throughput. The validation of PTDS is validated with well-known network simulator, NS-2.

1 Introduction

The IEEE 802.11 Distributed Coordination Function (DCF) mechanism uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to control access order in the Media Access Control (MAC) layer. The IEEE 802.11 DCF is used in order to reduce the collision probability of transmitted packets, through increasing the exponential backoff windows [1]. The DCF mechanism only provides a best-effort service, even though the importance of packets or types of packets (real-time or non real-time) exists. The IEEE 802.11 standardization committee established the IEEE 802.11e working group because of the requirement of transmitted packet priority, according to type of packet. The IEEE 802.11e working group has investigated IEEE 802.11e Enhanced DCF (EDCF) for supporting QoS. The MAC method of EDCF mechanism is similar to the DCF, except that EDCF applies a different value to control parameters supporting QoS, according to different packet types [3]. The Markov chain model is used frequently for mathematical analysis of the IEEE 802.11 mechanism [2] [7]. For mathematical analysis of the IEEE 802.11e EDCF mechanism, the Markov chain model is also used [6]. However the Markov chain analysis in ref. [2] has

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 122–131, 2006.

several problems that ignore cases of dropping packets and frozen slot time. Refs. [3] [5] are used in the modified Markov chain model for mathematical analysis of the IEEE 802.11e EDCF mechanism. In the case of [5], two cases that are not considered in [2] are included: the first is the case of dropping packets until the retry limit when they are not transmitted, and the second is the case that packet waiting for backoff time holds the frozen slot time of the busy channel conditions. Ref. [5] provides mathematical analysis of various factors, including, saturation throughput, packet dropping probability, and delay.

Based on [5], in this paper, the control parameters such as CW_{min} /CW_{max}, retry limit, $AIFS$, and so on in the IEEE 802.11e EDCF mechanism, having the most effect on throughput, will be considered in our analysis. Mainly, in order to obtain proportional and differentiated performance between adjacent priority classes, a method of setting of the most dominant parameter in QoS is proposed.

In Section 2, the formula for the throughput mathematical model for performance analysis in IEEE 802.11e is briefly described. The mathematical analysis of proportional throughput is derived, i.e., Proportional Throughput Differentiation Service (PTDS), among adjacent priority classes in Section 3. In Section 4, the validation of PTDS proposed through numerical and simulation results, is presented. Conclusions and further works are presented in Section 5.

2 Mathematical Model of IEEE 802.11e for Performance Analysis

2.1 Control Parameters in IEEE 802.11e EDCF Mechanism

In order to support QoS, the IEEE 802.11e EDCF classifies packets from each station in accordance with a priority selected among eight given user priorities, mapping into four Access Categories (ACs). AC denotes $AC[i]$ (i=0,1,2,3) in order to distinguish priority. The smaller value of i means a higher priority. Each $AC[i]$ is assigned different values of parameters ($CW_{min}[i]/CW_{max}[i]$, $AIFS[i]$, etc). If it exists $AC[i]$ and $AC[j]$ i is smaller than j, and the setting is like $CW_{min}[i] < CW_{min}[j]$, $CW_{max}[i] < CW_{max}[j]$ and $AIFS[i] < AIFS[j]$ according to priority order. That is, in a situation with an applied random backoff mechanism of IEEE 802.11 DCF, the packets of AC that has a higher priority have a higher transmission probability than the packets of AC that have a lower priority. The following Table 1 presents the default value of IEEE 802.11e EDCF parameters defined in ref. [3].

2.2 Mathematical Model

The IEEE 802.11e EDCF mechanism has four AC queues in a station, each AC queue consists of a priority class. However, in order to simplify mathematical analysis, it is assumed that each station belongs to one and only one priority class, and always has packets ready to send.

Table 1. Default value of IEEE 802.11e EDCF parameters

AC	CW_{min}	CW_{max}	AIFS
AC_BK(AC[3])	32	1024	7
AC_BE(AC[2])	32	1024	3
AC_VI(AC[1])	16	32	2
AC_VO(AC[0])	8	16	2

Fig. 1 is referred in ref. [5], for mathematical analysis, and presents a discrete Markov chain model, representing all states of the station having an i-class priority and applying the IEEE 802.11e EDCF mechanism. Each state is represented $b(i, j, k)$, where i, j and k are represented at a priority class, backoff stage, and contention window, respectively. The contention window in one value is chosen randomly from the backoff mechanism, among the values of [0 $CW_{i,j}$-1] with pre-decided $CW_{i,j}$ (i-th priority class, j-th backoff stage) depending on the current backoff stage such as Eq. (1).

$$CW_{i,j} = \begin{cases} 2^j CW_{i,0} & , 0 \le j \le R_i \\ 2^{R_i} CW_{i,0} & , R_i \le j \le L_i \end{cases} \tag{1}$$

where R_i and L_i are the maximum backoff stage and retry limit of i-th priority class, respectively. Let p_i denote the probability that a transmitted packet collides, and p_i also equals to the probability that a station in the backoff stage for the priority i class senses the channel as busy. The relationship between states in Fig. 1 is as that shown in the following Eqs. (2) $\sim$ (4).

$$b_{i,j,0} = p_i^j b_{i,0,0} \qquad (0 \le j \le L_i) \tag{2}$$

$$b_{i,j,k} = \frac{CW_{i,j} - k}{CW_{i,j}} \frac{1}{1 - p_i} b_{i,j,0} \qquad (0 \le j \le L_i, 1 \le k \le CW_{i,j} - 1) \tag{3}$$

$$\sum_{j=0}^{L_i} \sum_{k=0}^{CW_{i,j}-1} b_{i,j,k} = 1 \tag{4}$$

From Eq. (2) $\sim$ (4),

$$b_{i,0,0} = \frac{1}{\sum_{j=0}^{L_i} [1 + \frac{1}{1-p_i} \sum_{k=1}^{CW_{i,j}-1} \frac{CW_{i,j}-k}{CW_{i,j}}] p_i^j} \tag{5}$$

Through expanding Eq. (5),

$$b_{i,0,0} = \frac{2(1-p_i)^2}{[(2^{R_i}(L_i - R_i + 2) - 1)CW_{i,0} + (L_i + 1)(1 - 2p_i)](1 - p_i^{L_i+1})} \tag{6}$$

Let τ_i be the probability that a station in the priority i class transmits during a generic slot time,

$$\tau_i = \sum_{j=0}^{L_i} b_{i,j,0} = b_{i,0,0} \frac{1 - p_i^{L_i+1}}{1 - p_i} \tag{7}$$

Eq. (7) is expanded as Eq. (8) by inserting of Eq. (6),

$$\tau_i = \frac{2(1 - p_i)}{[2^{R_i}(L_i - R_i + 2) - 1]CW_{i,0} + (L_i + 1)(1 - 2p_i)} \tag{8}$$

Let n_i $(i = 0, ..., N-1)$ denote the number of stations in the priority i class. Then p_i , the probability that a transmitted packet collides, is as follows.

$$p_i = 1 - (1 - \tau_h)^{n_i - 1} \Bigg(\prod_{h=0, h \neq i}^{N-1} (1 - \tau_h)^{n_h} \Bigg). \tag{9}$$

Let p_b denote the probability that the channel is busy,

$$p_b = 1 - \prod_{h=0}^{N-1} (1 - \tau_h)^{n_h} \tag{10}$$

2.3 Throughput Analysis

Let $P_{s,i}$ $(i = 0, ..., N-1)$ denote the probability that successful transmission occurs in a slot time for the priority i class, and let P_s denote the probability that a successful transmission occurs in a slot time for all priority classes.

$$P_{s,i} = n_i \tau_i (1 - \tau_i)^{n_i - 1} \prod_{h=0, h \neq i}^{N-1} (1 - \tau_h)^{n_h} \tag{11}$$

$$P_s = \sum_{i=0}^{N-1} P_{s,i} \tag{12}$$

Let S_i $(i = 0, ..., N-1)$ denote the normalized throughput for the priority i class.

$$\begin{aligned} S_i &= \frac{E(\text{payload transmission time in a slot time for the } i \text{ class})}{E(\text{length of a slot time})} \\ &= \frac{P_{s,i} T_{E(L)}}{(1 - p_b)\delta + P_s T_s + (p_b - P_s) T_c} \end{aligned} \tag{13}$$

Let $\delta, T_{E(L)}, T_s, T_c$ denote the duration of an empty slot time, the time to transmit the average payload, the average time that the channel is sensed busy

because of successful transmission, and the average time that the channel has a collision, respectively. And where
$T_s = T_H + T_{E(L)} + SIFS + \gamma + T_{ACK} + DIFS + \gamma$
$T_c = T_H + T_{E(L^*)} + DIFS + \gamma.$

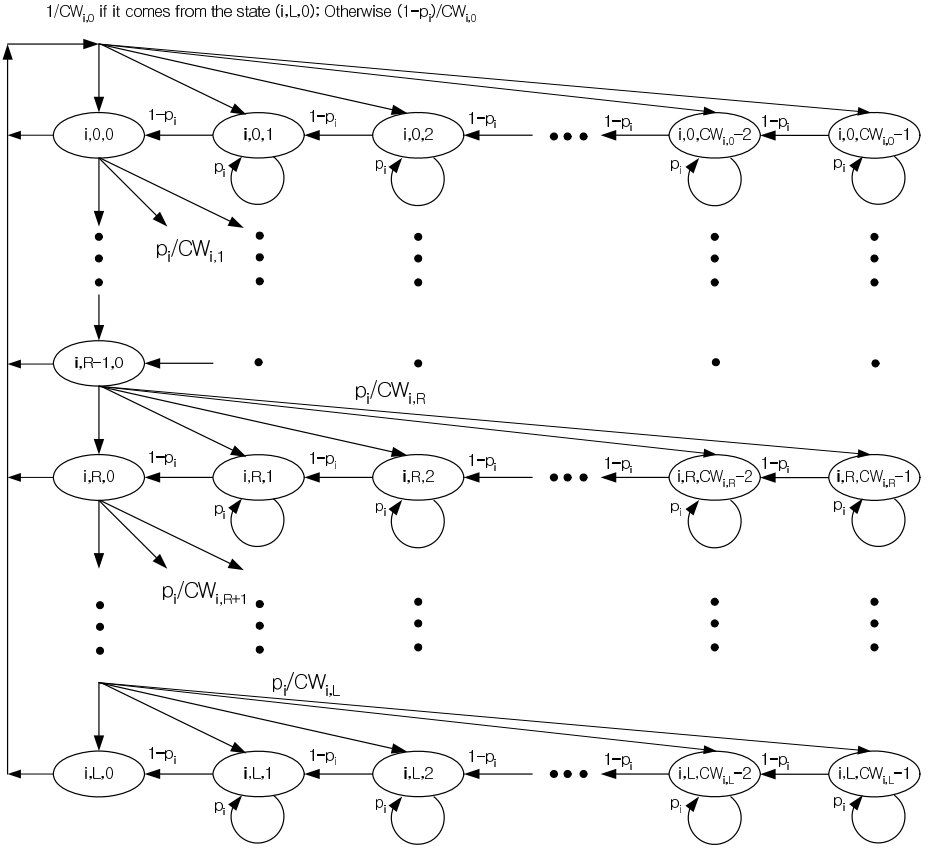


Fig. 1. Markov state diagram for the priority-class i

3 Analysis of Proportional Differentiation Service in Throughput

In Section 2, the mathematical model analyzed in Ref. [5] is shown, and mathematical analysis with regard to throughput is obtained. In this section, the guidance setting, meaning the method of setting control parameters in order to obtain proportional throughput differentiation between adjacent priority classes, is derived. Through this derivation, the method of the user obtaining the desired

proportional and differentiated services is deduced, among different priority classes in terms of throughput, and is denoted as PTDS.

3.1 Analysis of PTDS

In PTDS, the ratio of throughput between adjacent classes (i.e., i-th priority class and i+1-th priority class) is desired, as a certain given or desired value (K_T), presented in Eq. (14). It is assumed that the station number of each priority class is equal, and the maximum CW value is equal for all priority classes.

$$K_T \triangleq \frac{S_i}{S_{i+1}} = \frac{\frac{P_{s,i}T_{E(L)}}{(1-p_b)\delta+P_sT_s+(p_b-P_s)T_c}}{\frac{P_{s,i+1}T_{E(L)}}{(1-p_b)\delta+P_sT_s+(p_b-P_s)T_c}} = \frac{P_{s,i}}{P_{s,i+1}} \tag{14}$$

By inserting $P_{s,i}$ and $P_{s,i+1}$ in Eq. (11) and arranging Eq. (14),

$$\begin{aligned}\frac{S_i}{S_{i+1}} &= \frac{n\tau_i(1-\tau_i)^{n-1}\prod_{h=0,h\neq i}^{N-1}(1-\tau_h)^n}{n\tau_{i+1}(1-\tau_{i+1})^{n-1}\prod_{h=0,h\neq i+1}^{N-1}(1-\tau_h)^n} \\ &= \frac{\tau_i(1-\tau_{i+1})}{\tau_{i+1}(1-\tau_i)} = \frac{\frac{1}{\tau_{i+1}}-1}{\frac{1}{\tau_i}-1}\end{aligned} \tag{15}$$

Then the inverse of τ_i in Eq. (8) is considered.

$$\frac{1}{\tau_i} - 1 = \frac{[2^{R_i}(L_i - R_i + 2) - 1]CW_{i,0} + (L_i + 1)(1 - 2p_i) - (2 - 2p_i)}{2(1 - p_i)} \tag{16}$$

p_i is a smaller value than 1, L_i is not a large finite number. Therefore approximate value of $\frac{(L_i(1-2p_i)-1)}{2(1-p_i)}$ can be ignored in comparison with the first term.

$$\frac{1}{\tau_i} - 1 \simeq \frac{[2^{R_i}(L_i - R_i + 2) - 1]CW_{i,0}}{2(1 - p_i)} \tag{17}$$

Now, $\frac{1}{\tau_i}$ is inserted into Eq.(15). In Eq.(17).

$$\frac{S_i}{S_{i+1}} = \frac{\frac{1}{\tau_{i+1}}-1}{\frac{1}{\tau_i}-1} \simeq \frac{CW_{i+1,0}C_{i+1}}{CW_{i,0}C_i} \triangleq K_T \tag{18}$$

where $[2^{R_i}(L_i-R_i+2)-1]$ is denoted as C_i for simple notation. Then a guidance of setting minimum contention window can be deduced, when the desired ratio of throughput between adjacent classes is met, i.e., K_T.

$$CW_{i+1,0} = K_T \cdot \frac{C_i}{C_{i+1}} \cdot CW_{i,0} \tag{19}$$

4 Numerical and Simulation Results

In this section, the validation of PTDS is presented, through numerical and simulation results. For the basic parameter setting, IEEE 802.11 Frequency Hopping Spread Spectrum(FHSS) system parameters of [2] are referred to, as shown in Table 2.

Table 2. IEEE 802.11 FHSS system parameters [2]

Parameter	Value
Data rate	1 Mbps
Packet size	8192 bit
MAC header size	272 bit
Ack size	112 bit + PHY header
Phy header size	128 bit
Slot time	50 μs
Propagation time	1 μs
SIFS time	28 μs
DIFS time	128 μs

In both numerical and simulation results, priority classes are divided into class 0, class 1, and class 2. (class 0 is the highest priority). The number of each priority class station is assumed as equal. The numerical and simulation results are performed under usual conditions, at the number of each priority class station from 2 to 15. That is, the number of total stations (or nodes) varies from a minimum value of 6 to a maximum of 45 nodes.

In numerical analysis, p_i and τ_i $(i = 0, 1, 2)$ of each priority class are calculated both using Eq.(8), regarding packet transmission probability τ_i and Eq.(9) packet collision probability p_i for validation of PTDS. At this time, the value of CW_{min} among variable parameters is decided, according to the desired ratio (i.e., the ratio of between adjacent priority classes). When CW_{min} of class 0 as the highest priority class is CW_0, the CW_{min} of class 1 is $K_T \cdot CW_0$ $(= CW_1)$ and the CW_{min} of class 2 as the lowest priority class is $K_T \cdot CW_1$ $(= CW_2)$, i.e., if CW_0 is the value of 8, then CW_1 is 16 and CW_2 is 32. Then both the throughput of each priority class and the ratio of adjacent priority class are calculated.

In the simulation, the CW_{min} of each priority class is the same as that shown during numerical analysis. NS-2 experiments, according to changing of node number and time, have been performed.

To obtain the PTDS of the given ratio (e.g., K_T=2), the CW_{min} value of the adjacent priority class is provided with a differentiation factor of 2, from the derived guidance. Therefore, CW_{min} has the vlaue of 8 in the case of class 0, CW_{min} is 16 for class 1, and CW_{min} is 32 for class 2, respectively. The other control parameters for the PTDS are assumed as equal values, across different classes such as $CW_{\max \text{ for class } 0,1,2}$ is 1024, R_0 is 7, and $L_{0,1,2}$ is 8.

The throughput ratio between adjacent priority classes in Fig. 2 demonstrates that the PTDS is achieved with a reasonable number of nodes. The reason for inaccurate results when only a few nodes exist, can be explained by a combination of two reasons. The first reason is that the transmission of a packet among nodes in a higher priority class occurs only through the contention within same higher class when the node number is large.

However the case when only a few nodes exist, the transmission of a packet is achieved without contention, because the number of nodes in same priority

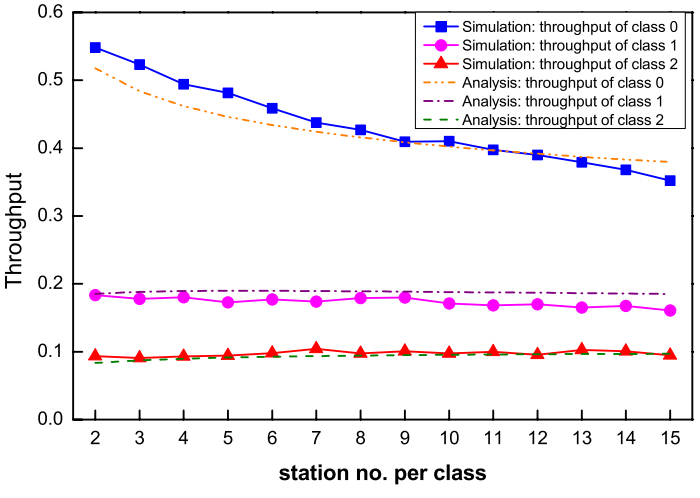


Fig. 2. Throughput when the CW_{min} is set as the value of 8 for class 0, 16 for class 1, and 32 for class 2, respectively

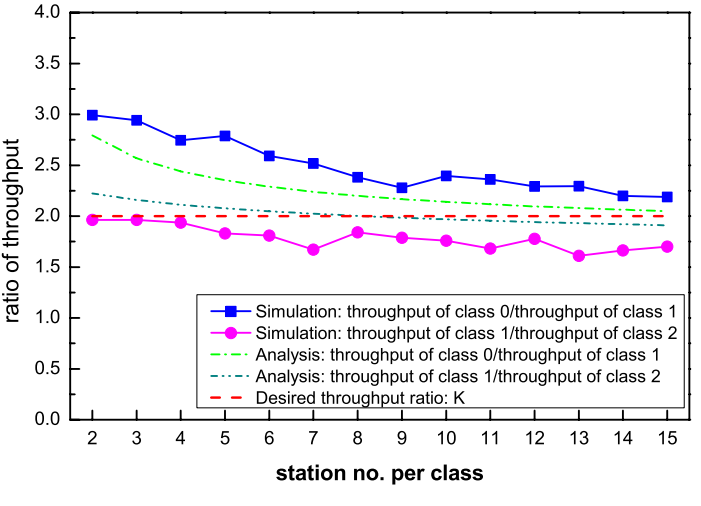


Fig. 3. Throughput ratio between adjacent priority classes

class is minimal. Therefore, the throughput of the high priority class is considerably higher than that of the low priority class. The second one is that nodes of a higher priority class have smaller CW_{min} values than the nodes in the lower priority class. Therefore, the discrepancy in throughput is larger than expected and the ratio of throughput by class appears larger than the desired ratio K_T.

In Fig. 3, it is demonstrated that the proposed PTDS is approximately reasonable. In particular, it can be confirmed that the PTDS between class 1 and class 2 completely corresponds to the ratio (e.g., K_T=2) of the desired throughput in both numerical and simulation results. The PTDS is approximately satisfied not only with the K_T value of 2 but also other values of K_T.

5 Conclusion

In this paper, mathematical analysis of the throughput for the proposed PTDS is derived in order to provide proportional and differentiated services among different priority classes in the IEEE 802.11e mechanism. In addition, validation of PTDS is provided through NS-2 simulation results.

It can also be pointed out which control parameters (i.e., minimum contention window size) are suitable for PTDS through mathematical analysis, some guidance for setting those parameters is provided.

The validation of guidance can be shown through NS-2 simulation results. Then the PTDS can be obtained among different priority classes, according to the given or desired proportional service ratio, in terms of throughput in the IEEE 802.11e EDCF mechanism.

For further works, research regarding proportional differentiation service of packet loss rate (PLDS) will be an interesting topic. The our effort is being conducted on finding parameters that provide the most effective packet loss rate, among parameters of the IEEE 802.11e EDCF mechanism for PLDS, through mathematical analysis and network simulation.

Acknowledgements

This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program supervised by the Institute of Information Technology Assessment (IITA) (IITA-2006-(C1090-0603-0011)).

References

1. IEEE Standard for Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications, P802.11, (1997)
2. G. Bianchi,: Performance Analysis of the IEEE 802.11 Distributed Coordination Function, IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, pp. 535-547, (2000)

3. IEEE 802.11 WG, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/D2.0, (2001)
4. Yang Xiao, Jon Rosdahl,: Throughput and Delay Limits of IEEE 802.11, IEEE COMMUNICATIONS LETTERS, vol. 6, NO. 8, (2002)
5. Yang Xiao,: Performance Analysis of IEEE 802.11e EDCF under Saturation Condition, IEEE Communications Society, (2004)
6. Jeffrey W. Robinson, Tejinder S. Randhawa,: Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function, IEEE Journal on Selected Areas in Communications, Vol. 22, NO. 5, (2004)
7. Yang Xiao,: A Simple and Effective Priority Scheme for IEEE 802.11, IEEE Communications Letters, Vol. 7, NO. 2, (2003)

A New Distributed Scheduling Algorithm to Guarantee QoS Parameters for 802.11e WLAN

Saeid Montazeri, Reza Berangi, and Mahmood Fathy

Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
Montazeri@comp.iust.ac.ir, rBerangi@iust.ac.ir, MahFathy@iust.ac.ir

Abstract. The existing distributed QoS mechanisms for WLAN MAC layer are only able to differentiate between various traffic streams without being able to guarantee QoS. On the other hand, most of the centralized QoS mechanisms are only able to guarantee QoS parameters for CBR traffic effectively. This paper addresses these deficiencies by proposing a new distributed QoS scheme that guarantees QoS parameters such as delay and throughput for both CBR and VBR traffics. The proposed scheme can also adapt to the various conditions of the network. To achieve this, three fields are added to the RTS/CTS frames that their combination with the previously existing duration field of RTS/CTS frames, guarantees the periodic access of a station to the channel. The performance of the proposed method has been evaluated with a specific simulator. The result are compared with IEEE 802.11e HCCA mechanism, and shown that it outperforms HCCA.

1 Introduction

The wireless LAN (WLAN) systems have received increasing popularity in recent years because they are cost effective, comfortable and have high capacity which makes them suitable for multimedia applications. Some of these applications like audio and video conferences require specific bandwidth, delay and jitter that need to be guaranteed as QoS parameters. IEEE 802.11 WLAN has defined a centralized QoS algorithm that is only applicable for small size system. This paper presents a novel distributed QoS algorithm for IEEE 802.11 that is suitable for both small and large size systems. IEEE 802.11 WLAN specifications define two different ways to configure a wireless network: infrastructure mode and ad hoc mode [1]. In the former there is an access point (AP) that connects stations to a distribution system but in the latter there is not such an AP and all stations connect in a distributed way. Distributed Coordination Function (DCF) is the fundamental medium access mechanism of 802.11 for both ad hoc and infrastructure modes. It uses carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In this mechanism if the channel is found idle for longer than a DIFS (distributed interframe space) then a station can transmit a packet otherwise, a backoff algorithm will start. In the backoff process a

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 132 – 145, 2006.

station computes a random value called backoff time between 0 and CW (Contention Window). The backoff time is decremented by one if the medium remains idle for a period of time longer than DIFS. When the backoff timer reaches zero the station can access the medium and start transmission. For each transmitted packet the source must receive an acknowledgement. If no acknowledgement is received, source assumes there have been a collision, schedules the packet for retransmission and doubles its CW to prevent more future collision [1]. There is not any service differentiation in the DCF and therefore all stations compete for accessing a channel with the same priority.

Point coordination function (PCF) is another medium access mechanism of 802.11 for infrastructure mode. It is a polling-based access mechanism which requires the presence of a base station that acts as an access point (AP). PCF is optional but DCF is mandatory for IEEE 802.11 MAC layer. If PCF is supported, both DCF and PCF coexist and the time is divided into superframes. A superframe starts with a beacon frame which is generated by the AP at regular intervals, so every station knows when the next beacon frame will arrive. After each beacon frame there are one Contention Free Period (CFP) that PCF can only operate and another Contention Period (CP) after that which DCF can only operate as illustrated in figure 1. During the CFP, AP sends poll frames to the stations. Each station that receives a poll can use the medium. The interframe space (IFS) between PCF data frames is shorter than DIFS so DCF is prevented from interrupting PCF mode. This IFS is called PCF interframe space (PIFS). A CFP ends with sending a CF_End frame by the AP and after that a CP will be started.

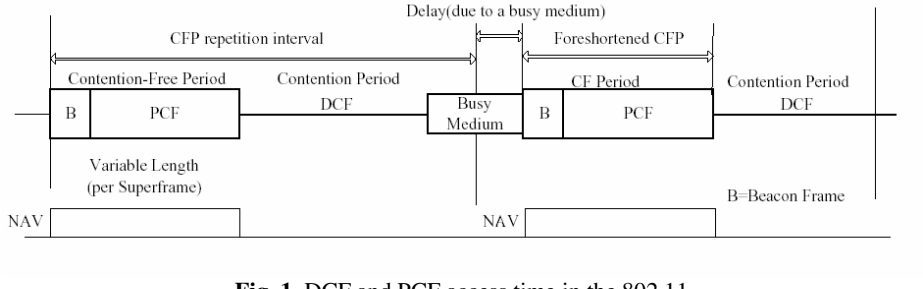


Fig. 1. DCF and PCF access time in the 802.11

IEEE Task Group e is now working on the support of QoS in a new standard, called IEEE 802.11e [2]. It introduces a new access method called Hybrid Coordination Function (HCF) which combines functions from the DCF and PCF mechanisms. HCF has two access mechanisms: enhanced DCF (EDCF) and controlled channel access mechanism (HCCA). These two methods support QoS. We will describe them more. In the HCCA there is a scheduler for scheduling different Traffic Streams (TSs) on different stations.

This paper proposes a Distributed Scheduling Algorithm (DSA) to guarantee QoS parameters for both VBR and CBR traffic streams. The performance of DSA is evaluated through computer simulations and compared with the performance of two

centralized methods, HCCA of 802.11e and FHCF [3], as well as one distributed method, EDCF of 802.11e.

The rest of paper is organized as follows: section 2 introduces 802.11e. Section 3 describes the proposed DSA algorithm. Section 4 describes the simulation results. Finally, section 5 concludes the paper.

2 IEEE 802.11e MAC

HCF has both contention based access method and polling based access method. EDCF introduces the concept of Access Categories (ACs), which can be considered as instances of the DCF access mechanism that provide support for the prioritized delivery at the station.

2.1 Enhanced Distributed Coordination Function (EDCF)

Like DCF, EDCF uses CSMA/CA protocol to access the wireless media and can only operate during CP. In EDCF method each AC within the stations contend for transmission opportunity (TXOP) independently. TXOP is defined as the interval of time when a particular station has the right to initiate the transmission onto the wireless channel. Each AC starts the backoff after detecting the channel to be idle for a time interval equal to the Arbitration InterFrame Space (AIFS). Each AC has its AIFS depends on the priority that is assigned to it. Figure 2 demonstrates the eight different queues for eight ACs.

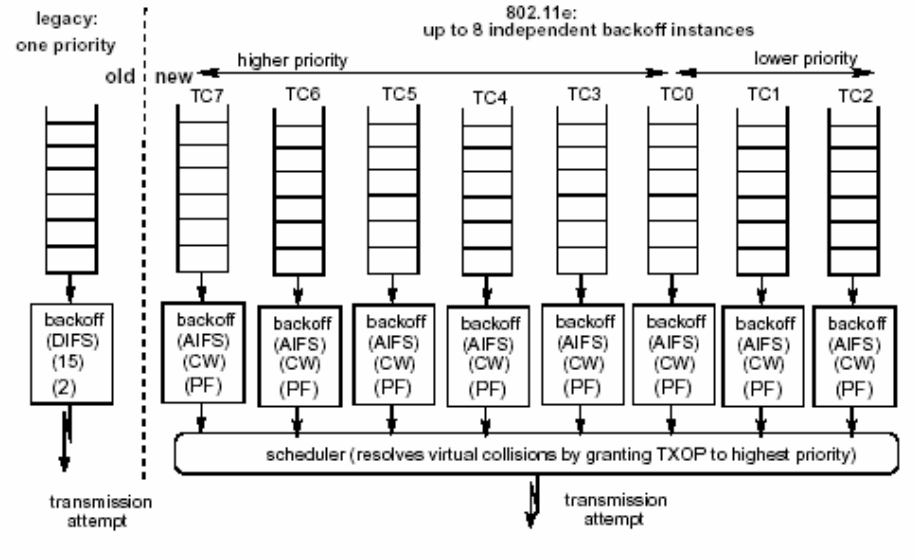


Fig. 2. Old DCF and EDCF

Each AC has own queue, $CW_{min}[AC]$, $CW_{max}[AC]$ and PF[AC]. Figure 3 shows the different ways to provide service differentiation.

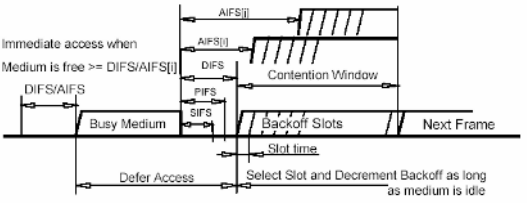


Fig. 3. Different AIFS for different priorities

For each AC, backoff is generated in the rang of [1, CW[AC]+1] which the initial value for the CW is $CW_{min}[AC]$. CW is increased whenever the node involves in a collision by the equation 1 up to $CW_{max}[AC]$.

$$newCW[AC] = ((oldCW[AC] + 1) * PF[AC]) - 1 \quad (1)$$

Where PF is the persistence factor, which is equal 2 by default. It determines the degree of increase for the CW when a collision happens. Among the various QoS parameters, EDCF can only differentiate between different priorities. Some researches have been carried out to improve the EDCF [4-8] but no one could guarantee QoS parameters.

2.2 HCF Controlled Channel Access (HCCA)

In upcoming IEEE 802.11e the polling based scheme of 802.11 is extended in the form of HCCA, in which there is a Hybrid Coordinator (HC) usually co-located with a QoS AP (QAP). HC can access channel after waiting for a time equal to the PIFS which is shorter than each AIFS and DIFS. Thus, HC can get the channel in both CFP and CP. During CP, TXOP for each station can be received in two ways: by using EDCF rules (EDCF_TXOP) or by receiving a poll from HC (Polled_TXOP). During CFP, TXOP is determined only by HC with poll frames. CFP is ended by a CF_End frame which is transmitted by HC.

2.3 802.11e HCF Scheduling Scheme

There is a simple scheduler in the 802.11e HCF. If a QoS enhanced station (QSTA) needs a strict QoS support, it should send QoS requirement packet to the QAP while QAP can allocate the corresponding channel time for different QSTAs according to their requirements. Figure 4 shows the new beacon interval of 802.11e and its CFP and CP. QAP can operate in both CFP and CP. During the CP, the QAP can starts several contention free burst, called controlled access period (CAPs) at any time to control the channel.

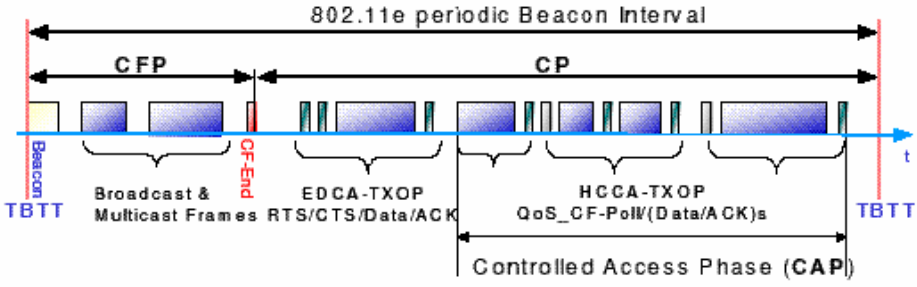


Fig. 4. CFP ,CP and CAPs in the 802.11e

Each station that requires contention free access send a QoS request frame to the QAP containing the mean data rate of the application (ρ), the maximum service interval (MSI) and MAC Service Data Unit (MSDU) size (L) to get a TXOP. QAP calculate the TXOP in two steps. First it determines the minimum value of all MSIs required by different traffic streams and chooses the highest submultiples value of the 802.11e beacon interval duration (duration between two beacons) as the selected SI which is less than the minimum of all requested MSIs. This selected SI is the time between two successive TXOPs for all streams. In the second step QAP should calculate TXOP for each TSs in different QSTAs. The TXOP should correspond to the duration required for transmitting all packets that is generated during a SI by the specific TS. Figure 5 shows the CPs, CFPs, selected SI and EDCF time.

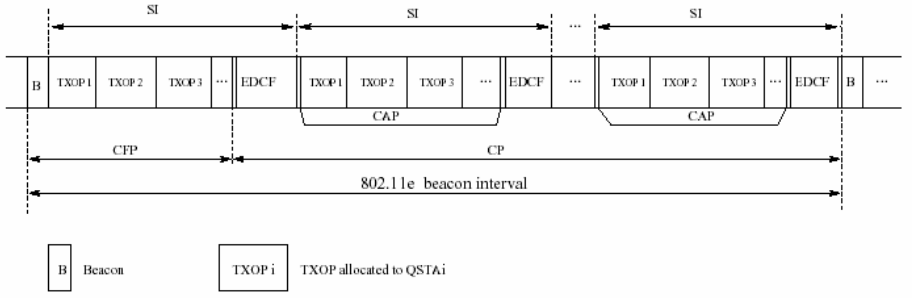


Fig. 5. Structure of the 802.11e beacon interval

The TXOP is determined by the equation 2 and 3 where R is the physical transmission rate and M is the size of maximum MSDU (2304 bytes) and O determines the overhead in time units.

$$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right) \tag{2}$$

$$N_i = \left\lceil \frac{SI \times P_i}{L_i} \right\rceil \tag{3}$$

3 Distributed Scheduling Algorithm (DSA)

A well known problem in WLANs, called hidden node, happens when two stations that communicate with a station, could not hear each other and as a result their packets collide. IEEE 802.11 has added two special frames called request to send (RTS) and clear to send (CTS) to address this problem. In this method, a station sends a RTS frame to the receiver and transmission only starts when a CTS frame is replied by the receiver. This local hand shaking between transmitter and the receiver provides an excellent opportunity to guarantee QoS. To achieve this, the proposed DSA algorithm adds several fields in RTS/CTS frames to obtain distributed guarantee for QoS in one hop WLANs.

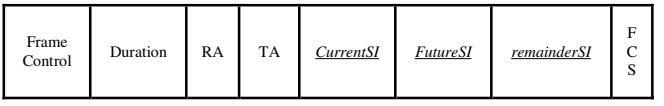


Fig. 6. New RTS/CTS frame structure

Three fields i.e. currentSI, FutureSI and remainderSI, shown in Figure 6, are added to both RTS/CTS. Therefore, two timers are needed in each QSTA: a duration timer and a service interval (SI) timer. A method can guarantee QoS parameters like delay for a traffic stream when it can allocate a channel to this specific traffic stream (TS) for a special duration in a specific SI. Duration for i^{th} TS in the j^{th} QSTA can be obtained by

$$D_i^j = N_i^j * \left(\frac{M_i}{R} + 2SIFS + ACK_{time} \right) + RTSNumber * (RTS_{time} + CTS_{time} + 2SIFS) - (RTS_{time}) \quad (4)$$

where, D_i^j is the time required to transmit N_i^j packets with length, M_i with the physical rate, *R* plus the time required to transmit RTSNumber RTS/CTS. Here, N_i^j is the number of packets in the queue of i^{th} TS in the j^{th} QSTA at the beginning of D_i^j.

The question here is how to guarantee D_i^j repeats every SI seconds for the i^{th} TS in the j^{th} QSTA, without disturbing other QSTAs with different TSs requirements?

Suppose i^{th} TS in the j^{th} QSTA is the first one that starts the transmission in the WLAN with using EDCF method. It calculates D_i^j and puts it in the duration field of RTS. Then it sets the CurrentSI and FutureSI with maximum service interval for i^{th} TS. The j^{th} QSTA transmit RTS frame and wait for receiving CTS. Destination receives the RTS and calculates the duration field for CTS by using equation 5.

$$Duration_{CTS} = Duration_{RTS} - (SIFS + CTS_{time}) \quad (5)$$

Then, it transmits the CTS frame to the j^{th} QSTA. All the QSTAs that receive the RTS or CTS understand a new transmission will be started with the length of duration time and will be repeated with CurrentSI period. Therefore, they reserve this time for

station j by setting and starting their duration timers with the duration field of RTS/CTS and the same for SI timers with the CurrentSI field of RTS/CTS. They also save FutureSI field of RTS/CTS for the next SI timer restart. When source receives the CTS frame it transmit data frame. This process repeats RTSNumber times to assure that all the active stations in the communication range have received at least one RTS or CTS. After which only data frames will be transmitted. The RTSNumber depends on the BER of the channel and increases with increasing the BER. In our simulation RTSNumber is set to 2. The duration field of RTS is updated by the value of the duration timer in the source station. Moreover, remainderSI is always filled with the value of SI timer in both source and the destination. After this, all stations have reserved the allocated time for the j^{th} station and they remain silent during this time. The exact duration is announced in the duration field of RTS/CTS by the j^{th} station so other stations do not need to save this value.

DSA can operate efficiently for VBR traffics because each station can change its duration at the start of transmission and announce it by RTS/CTS frames.

After finishing D_i^j all QSTAs start to compete for accessing to the channel based on EDCF method. Suppose k^{th} QSTA gets the channel for its l^{th} TS. It fills the duration field by using equation 5, sets the CurrentSI with the value of CurrentSI of j^{th} QSTA. However, it sets the FutureSI field with maximum service interval that l^{th} TS required if it is equal or less than previous FutureSI (related to i^{th} TS in the j^{th} QSTA). So, all the QSTAs that receive the new RTS understand that they must initialize their SI timer with FutureSI field of new RTS at the end of current SI. So after a number of SIs whole the network works with the sufficient SI that is the minimum of maximum service intervals for all TSs.

It must be mentioned that the stations only compete for accessing the channel in the unreserved periods. The remainderSI field is used by a new QSTA that enters a WLAN to wait before starting the channel access process.

The mentioned channel access process in DSA eliminates the need for a point coordinator and though each wireless station can act as an AP when it is connected to the wired network. This enhances the survivability of WLANs in case of an AP failure.

4 Simulation Results

DSA is implemented by the ns-2 simulator and compared with three schemes of the previously reported works for the distributed [4-8] and centralized [3, 9-11] channel access mechanism. Both distributed (EDCF) and centralized methods (HCCA) of 802.11e [2] are selected because they are widely used in the literature for comparisons. The fair HCF (FHCF) proposed in [3] is also selected as the third scheme to compare with our results. Two kinds of simulation scenario have been used. The first one contains 18 sources and one destination. The second contains 6 sources and one destination. In both scenarios the destination is QAP that contains a PC to satisfy the requirements for HCCA and FHCF but it is an ordinary QSTA for our proposed method.

4.1 Scenario 1

In scenario 1, 6 QSTAs send a high priority on-off audio traffic (64 kbps) each, another 6 QSTAs send a VBR video traffic (200 kbps of average sending rate) with medium priority each and, 6 QSTAs send a CBR MPEG4 video traffic (3.2 Mbps) with low priority each. Table 1 summarizes the different traffics used for this simulation. We model the audio flow by on-off source with parameters corresponding to a typical phone conversation [12]. UDP is used as transport protocol.

Table 1. Scenario 1 nodes and traffic flows

Node	Application	Arrival period (ms)	Packet size (bytes)	Sending rate (kb/s)
1→6	Audio	4.7	160	64
7→12	VBR Video	≈26	≈660	≈200
13→18	MPEG4 Video	2	800	3200

Table 2 demonstrates the characteristics of the selected traffics. The different VBR flows have been obtained with VIC video conferencing tool using the H.261 coding and QCIF format for typical "head and shoulder" video sequence. The PHY and MAC layer parameters used in the simulation are also summarized in Table 3.

Table 2. Traffic specification

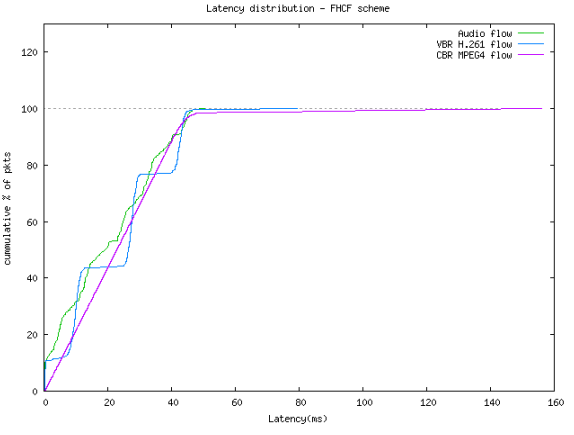
Traffic Type	Priority	CW_{Min}	CW_{Max}	Max Delay (ms)
Voice	6	7	15	50
VBR Video	5	15	31	100
CBR Video	4	15	31	100

Table 3. The PHY and MAC layer parameters

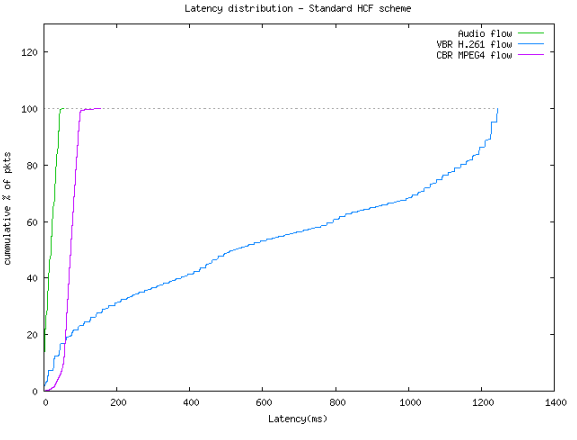
SIFS	16μs	CCA Time	4 μs
DIFS	34 μs	MAC Header	38 Bytes
ACK Size	14 Bytes	PLCP Header Length	4 bits
PHY Rate	36 Mb/s	Preamble Length	20 bits
Minimum Bandwidth	6 Mb/s	RTS Length	28 Bytes
Slot Time	9 μs	CTS Length	28 Bytes

Figure 7 demonstrates the latency distribution of the simulated methods. It shows that in the DSA all the traffic streams have the maximum latency under the maximum that they can tolerate (Table 2). In contrast, VBR traffic latency in HCCA is uncontrollable and for the EDCF it exceeds the limit. Maximum VBR latency for the proposed method is 80ms but for FHCF is 50ms. This might seem to be an advantage for FHCF but it is a centralized method that needs PC. However, DSA is a distributed algorithm that does not need any PC and still its latency is lower than the tolerable latency for VBR video.

There are two methods to increase the channel load: increasing the node number and increasing the packet size. The latter is selected for increasing the channel load in this simulation. The packet size of CBR MPEG4 video has been increased from 600 to 1000 bytes to achieve 96% channel load.

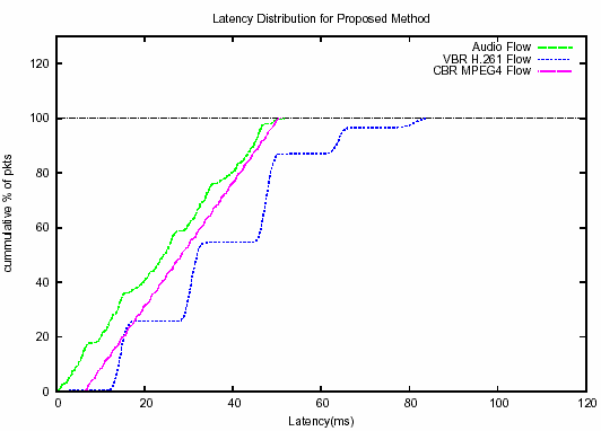


7-a)FHCF

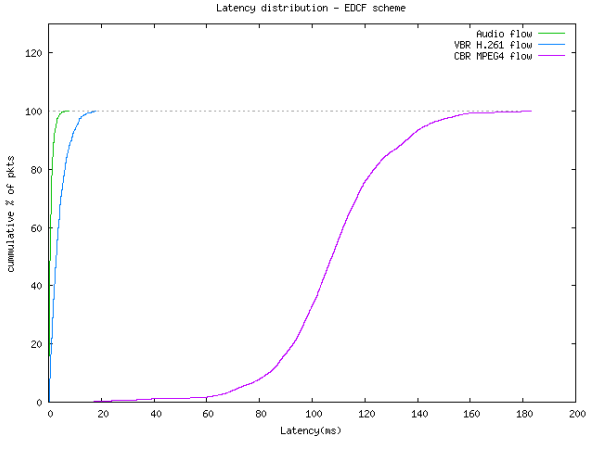


7-b) Standard HCF

Fig. 7. Latency Distribution for different Schemes



7-c) AFDSA



7-d) EDCF

Fig. 7. (*continued*)

Figure 8 shows fairness for VBR and CBR video traffic streams when the load increases up to 96%.

In order to compare the fairness of the different schemes for the same kind of traffic, Jain's fairness index has been employed [13]:

$$J = \frac{\left(\sum_{i=1}^{n} d_i\right)^2}{n \sum_{i=1}^{n} d_i^2}. \tag{6}$$

Where d_i is the mean delay of the flow i and n is the number of flows. Figure 8 indicates that FHCF and DSA are fairer than HCCA.

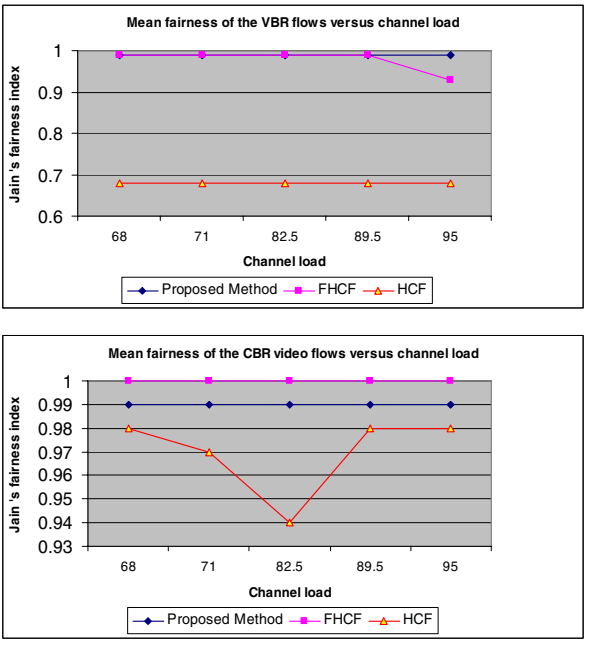


Fig. 8. Mean fairness for VBR and CBR flows

4.2 Scenario 2

In scenario 2 (see Table 4) there are 7 nodes. Six of which are sources and one node is destination, each QSTA has three different traffic flows (audio, VBR H.261 video and CBR MPEG4 video flows) simultaneously through three different MAC layer priority classes. We increase the channel load by increasing the packet size of CBR MPEG4 traffic from 600 bytes (2.4 Mbps) to 1000 bytes (4Mbps) using a 100 bytes increment and keeping the same inter-arrival period of 2ms. Figure 9 and 10 show the mean delay and fairness of several types of flows, obtained with the various schemes, for different loads of network respectively.

Audio and VBR H.261 Video Flows

Figure 8 shows that the delay is almost constant for the FHCF and the DSA with increase in load which indicates it does not strongly depend on the network load. In HCCA VBR traffic has a high value of delay (300 ms) that exceeds the limit for this kind of traffic. In EDCF mean latency is very low for audio and VBR video traffic streams because of the high priority that had been assigned for these streams. This increases the delay of CBR video traffic and it linearly increases with increase in

traffic load. Figure 9 shows that Jain index for all four methods that are the same for audio traffic. It is apparent that DSA and FHCF are better than EDCF and HCCA for VBR video traffic.

CBR MPEG4 Video Flows

In our simulation, CBR streams are responsible for increasing the traffic load. As we can see in Figure 8, latency is almost constant for HCCA, FHCF and DSA but increases with the load increment for EDCF such that for loads more than 79% it exceeds the limit for CBR traffic (100ms). Figure 9 shows that Jain's index for all methods is high with minor differences for CBR video traffic.

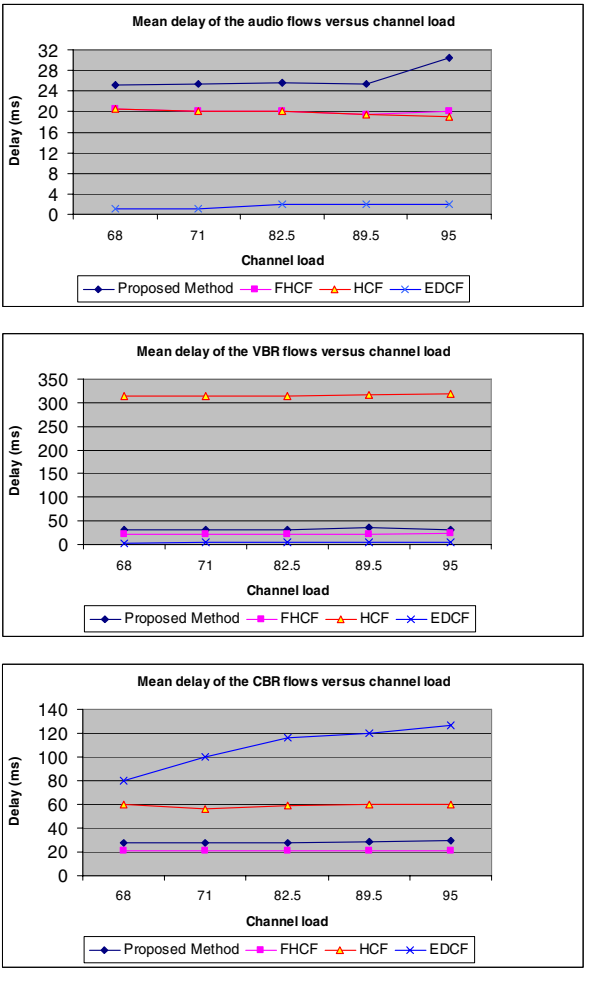


Fig. 9. Mean latency for different flows when channel load increased

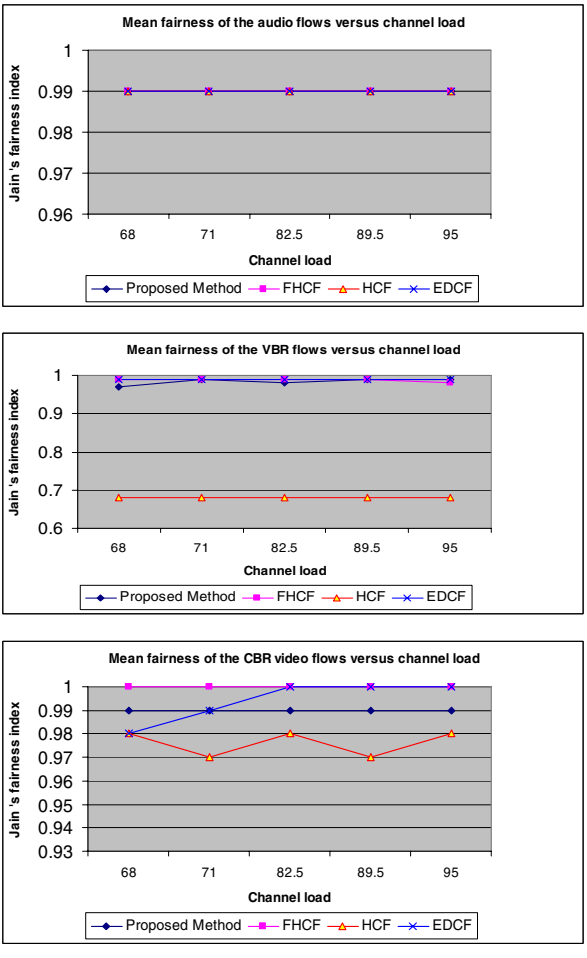


Fig. 10. Mean fairness of different flows when channel load increased

5 Conclusion

A new distributed MAC scheduling algorithm (DSA) for upcoming 802.11e standard is proposed and evaluated. The mechanism introduces three additional fields to the RTS/CTS frame to guarantee QoS. The EDCF method of 802.11e is used to access the channel for the first time. When time duration is reserved for a station, the rest of the stations only compete for accessing the channel in the unreserved periods. It is shown through extensive simulation that the DSA can guarantee QoS for both CBR and VBR traffic. It does not need any point coordinator and each node can play an access point role if it is connected to the backbone.

References

[1] IEEE 802.11 WG: "IEEE Std 802.11-1999, Part 11: Wireless LAN MAC and physical layer specifications", Reference number ISO/IEC 8802-11:1999 (E), (1999).

[2] IEEE 802.11 WG: "IEEE 802.11e/D8, Wireless MAC and physical layer specifications: MAC enhancements. for QoS" (2004).

[3] P. Ansel, Q. Ni, and T Turletti. "FHCF: An Efficient Scheduling Scheme for IEEE 802.11e". ACM/Kluwer Journal on Mobile Networks and Applications (MONET), Special Issue on Modeling and Optimization in Wireless and Mobile Networks, 2005.

[4] M. Malli, Q. Ni, T. Turletti and C. Barakat "Adaptive Fair Channel Allocation for QoS Enhancement in IEEE 802.11 Wireless LANs" IEEE ICC 2004 (International Conference on Communications), Paris, France, June 2004

[5] J. Zhao, Z. Guo, Q. Zhang and W. Zhu "Distributed MAC Adaptation for WLAN QoS Diffrerentiation" IEEE GLOBECOM 2003

[6] L. Romdhani, Q. Ni and T. Turletti. "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks", IEEE WCNC'03 (Wireless Communications and Networking Conference), New Orleans, Louisiana, USA, March 16-20, 2003.

[7] G. W. Wong and R. W. Donaldson "Improving the QoS Performance of EDCF in IEEE 802.11e Wireless LANs" IEEE 2003.

[8] W. P. Atikon, S. Banerjee and P. Krishnamurthy "A-DRAFT: An Adaptive QoS Mechanism to Support Absolute and Relative Throughput in 802.11 Wireless LANs" MSWiM 04 October 4-6 Venezia, Italy ACM 2004.

[9] A. Grilo, M. Macedo, and M. Nunes "A Scheduling Algorithm for QoS Support in IEEE802.11e Networks" IEEE Wireless Communication June 2003 pp36-43.

[10] G. Boggai, P. Camarda, L.A. Grieco, and S. Mascolo "Feedback-based bandwidth allocation with call admission control for providing delay guarantees in IEEE 802.11e networks" Computer Communications Elsevier 2004.

[11] B. A. Venkatakrishnan, and S. Selvakennedy "An Enhanced HCF for IEEE 802.11e Wireless Networks" MSWiM 04 October 4-6 Venezia, Italy ACM 2004.

[12] P. M. Soni, A. Chockalingam:" Performance analysis of UDP with energy efficient link layer on Markov fading channels" IEEE Transactions on Wireless Communications, 1 2002.

[13] R. Jain: "The art of computer systems performance analysis". John Wiley & Sons, 1991.

The Soft QoS-Aware Call Admission Control Scheme for HCCA in IEEE 802.11e

Sang Hoon Jang and Yeong Min Jang

School of Electrical Engineering, Kookmin University,
861-1, Jeongneung-dong, Songbuk-gu, Seoul, Korea
{jangsang, yjang}@kookmin.ac.kr

Abstract. A soft QoS-aware call admission control scheme to support the QoS requirements of multimedia traffic in IEEE 802.11e HCCA (HCF Controlled Channel Access) is proposed. HCCA can be used to provide the IP quality of service guaranteed in an IEEE 802.11e infrastructure like WLAN. A resource allocation scheme based on the soft QoS scheme will be investigated. In order to evaluate its performance, the proposed scheme, under a transient fluid flow model, will be compared with a traditional scheme in terms of Frame Loss Rate.

1 Introduction

In the past few years, there has been an explosion in the deployment of Wireless LANs (WLANs) conforming to the IEEE 802.11 standard. The PHY of WLAN evolved from 802.11 to 802.11a so that WLAN can support high speeds. The MAC of WLAN, however, was not upgraded from 802.11, so WLAN had difficulties adapting to various wireless environments. The popularity of multimedia applications has increased dramatically, so the quality of service (QoS) support in communication networks has become more and more important. However, 802.11 is designed for best-effort services. The IEEE 802.11 Medium Access Control (MAC) employs a mandatory contention-based channel access function, called the distributed coordination function (DCF), and an optional centrally controlled channel access function, called the point coordination function (PCF). However, DCF is unsuitable for multimedia applications with QoS requirements even though PCF can provide some limited QoS support. The IEEE 802.11 Task Group e is developing MAC enhancements to support the QoS requirements of sensitive applications to enable better mobile user experiences and to enable a more efficient use of wireless channels. The IEEE 802.11e draft specification defines two channel access mechanisms. One is an EDCA (Enhanced Distributed Channel Access) using CSMA/CA (Carrier Sensing Multiple Access/Collision Avoidance). The other is HCCA, using a polling scheme. Although the 802.11e standard has defined the QoS-enabled MAC mechanism to different QoS issues, admission control is an important component in providing guaranteed QoS parameters. The purpose of the admission control is to limit the amount of traffic admitted into a particular service class so that, at the same time, medium resources

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 146 – 155, 2006.

can be maximally utilized [1]. Traditional schemes provide hard QoS guarantees, but the network environment does not provide hard QoS guarantees. The network must provide soft QoS guarantees that are constrained by a minimum channel quality [2], especially when the polling scheme in HCCA needs an admission control scheme for greater station (STA) acceptance. The admission control scheme for HCCA already exists, called the simple scheduler in draft standards. The simple scheduler is very simple, but it is an inefficient scheme. The simple scheduler can guarantee the maximum requirement of QoS, called the hard QoS scheme. The number of acceptable STAs, however, is limited. So an admission control scheme, using soft QoS, is proposed. The proposed scheme can accept more STAs.

The rest of the paper is organized as follows. Section 2 presents the soft QoS, using the Greedy Approach. The fluid model and the central limit approximation model of the proposed scheme are presented in Section 3. Finally, the conclusion is presented in Section 4.

2 Soft QoS-Aware CAC by Using the Greedy Approach

2.1 Soft-QoS Scheme

The soft QoS is the connection between the available resources of the network and the requirement of the application. If collision occurs, we can use a softness profile in resources allocation to which the QoS is guaranteed for the soft QoS scheme. The multimedia network requests the smallest bandwidth for application performance, and the smallest bandwidth has a dynamic range per user performance requirement, application's use mode, and application's tolerance. The smallest bandwidth can be described as the softness profile. Soft QoS guarantees QoS to furnish a feasible maximum resource to active connection. In the existing hard QoS, bandwidth requiring all connections is permitted, and if collision occurs, the connection is dropped. In wireless networks using soft QoS, if collision occurs, the active connection's bandwidth is reduced till the traffic's quality boundary and the remnant is proffered a new connection. It is the method that improves the utilization of the network and its performance. For soft QoS, the function of the bandwidth allocation of each traffic flow and the user's level of satisfaction must be determined. The Critical Bandwidth ratio (ξ) must also be defined. The value of ξ varies according to the kinds of traffic. ξ of the voice traffic ranges from 0.8 to 0.9, and ξ of the video traffic ranges from 0.6 to 0.8 [5].

2.2 Greedy Approach

In the knapsack problem, there is a knapsack that should be filled up to SI (W) with some N connections. Each connection has TXOP and profit. Given are N connections of TXOPs w_1, w_2, w_3,…., w_N and profit P_1, P_2, P_3…., P_N that are equal to the Nominal MSDU Size/TXOP. Connections are initially in order, and every connection has to be

examined in sequence in order to find most optimal knapsack, and to check if this connection belongs there. This 0-1 knapsack optimization algorithm is used in the resource allocation problem, using the soft QoS scheme.

The greedy approach is to grab the data items sequence, each time taking the one that is deemed best according to some criterion, without regard for the choices it has made before or will make in the future [6]. That is, a greedy approach is to steal connections with the largest profit per TXOP of the connection. However, a greedy approach would not work very well if the most profitable connection had a large TXOP in comparison with its profit. The knapsack problem can be formulated as follows: We are given a set D = {connection$_1$, connection$_2$, ….. , connection$_K$}, such that each connection$_k$ has the TXOP, and profit p_k and a bound on the SI/TXOP of the knapsack W. The problem is to determine a subset A of D such that $\sum_{cnnection_k \in A} \left(\frac{p_k}{w_k}\right)$ is maximized, subject to $\sum_{connection_k \in A} w_k \le W$. The problem can be formulated as follows:

$$P(D) = \max \sum_{connection_k \in A} \left(\frac{p_k}{w_k}\right) = \max \sum_{connection_k \in A} \left(\frac{L_K}{TXOP_K}\right) \tag{1}$$

With constraint $\sum_{connection_k \in A} w_k \le W$ ·

2.3 Proposed CAC Scheme

An algorithm that does not use enough TXOP is proposed. If the TXOP is enough, a reference scheme may be used, but if the TXOP is not enough, the soft QoS scheme is used. If available TXOP is smaller than the requested TXOP, QAP can accept new STA, using soft QoS. When QAP makes a polling list, and if many STAs request TXOP, QAP must select one STA for TXOP. In this case, QAP can use the User priority (Access Category). When high priority STA is selected, and if the User priority stays the same, QAP must use another method. The Greedy Approach scheme is proposed. If many STAs with the same User priority request TXOP, QAP calculates the profit of each call. QAP selects new calls with the highest profits. When using soft QoS, QAP benefits from the profit of each call. QAP reallocates TXOP from the call with the smallest profit to the call with the highest profit so QAP can reduce the loss of QoS. Fig. 1 shows the proposed algorithm.

Step 1. A new call is received (one or more).If there is only one new call, go to step 3.

Step 2. QAP checks the User priority. If the User priority of all calls is the same, QAP calculates the profit of each call, using Eq. (1). QAP then selects an accepted call.

Step 3. QAP calculates available TXOP, using Eq. (2)

$$t_{rem} = t_{total} - \sum_{m=1}^{M} \sum_{n=1}^{N_m} TXOP_{mn} \tag{2}$$

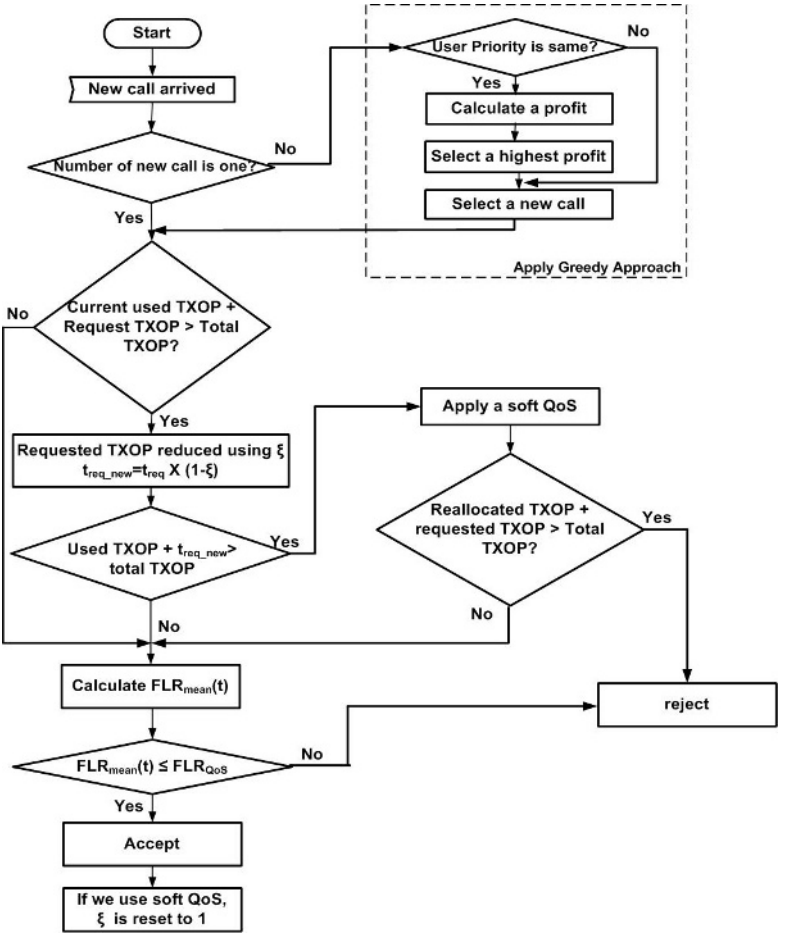


Fig. 1. The proposed soft QoS-aware CAC, using the Greedy approach

We assume that t_{req} is a requested TXOP by a new call, and t_{rem} is remained TXOP. If QAP has TXOP that is enough to accept new calls, a new call is accepted. But if TXOP is not enough to accept new calls, QAP sets up a critical TXOP ratio (ξ) for a new call. The equation becomes $t_{req\_new} = t_{req} \times (1-\xi)$, which is equal to the minimum TXOP requirement.

Step 4. If TXOP is satisfied Eq. (3), a new call is accepted. Otherwise, QAP uses the soft QoS algorithm.

$$\sum_{m=1}^{M}\sum_{n=1}^{N_m} TXOP_{mn} + t_{req\_new} \le TXOP_{total} \tag{3}$$

Step 5. QAP uses the soft QoS algorithm. First, QAP checks the User priority, and QAP calculates profit, using Eq. (4)

$$\frac{p_n}{w_n} = \frac{L_n}{TXOP_n} \tag{4}$$

where L is Nominal MSDU size. QAP reallocates TXOP from the call with the smallest values of p_n/w_n, using Eq. (7). Re-allocated TXOP can be calculated by the following. First, L_k and M are re-calculated, using Eq.(5)

$$L_{k,new} = L_k \cdot \xi_{mn} \quad , \quad M_{new} = M \cdot \xi_{mn} \tag{5}$$

So

$$TD_{new} = \max\left(\frac{N_k \times L_{k,new}}{R_k} + O, \frac{M_{new}}{R_k} + O \right) \tag{6}$$

Finally,

$$TXOP_{mn\_new} = \sum_{k=0}^{n} TD_{new} + SIFS + t_{poll} \tag{7}$$

If re-allocated TXOP is smaller than the requested TXOP, QAP re-allocates TXOP to the call with the next smallest profit. When the sum of the re-allocated TXOP is smaller than the requested TXOP, the new call is rejected.

Step 6. QAP calculates FLR_{mean} of soft QoS, and then compares FLR_{QoS}. FLR_{QoS} is requirement of QoS. If FLR_{mean} doesn't satisfied FLR_{QoS}, new call is rejected. In next section, we present process of calculating FLR_{mean}, and we show that our proposed scheme satisfies FLR_{QoS}.

Step 7. The critical TXOP ratio is reset to 1.

3 Calculating Frame Loss Rate by Using Central Limit Approximation

A series of packets arriving in the form of a continuous stream of bits or a fluid is assumed. We also assume that "OFF" and "ON" periods of the source both exponentially distributed with parameters λ_k and μ_k, respectively. The transitional flow rate from the "ON" state to the "OFF" state is μ_k and from "OFF" to "ON" is λ_k. In this traffic model, when a source is in the "ON" state, it generates packets with a constant inter-arrival time, $1/R_k$ seconds/bit. When the source is in the "OFF" state, it does not generate any packets [8]. Station m has one class-m source sharing an uplink are assumed to have the traffic parameters (λ_k, μ_k, R_k). Fig. 2 shows the fluid model.

A statistical bufferless fluid model to predict the FLR(t) at a future point in time t. Let $\Lambda_k(t) = (R_k Y_k(t))$ be the arrival rate from $Y_k(t)$ active source. In our model, $Y_k(t)$ is 0 or 1.We assume one STA has one traffic class, but traffic class of each STA is different. Taking into consideration the fact that N(=1) is given by an arbitrary

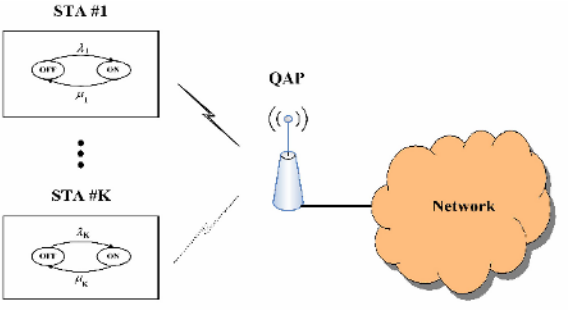


Fig. 2. Fluid model architecture

initial condition $Y_k(0)=I=[Y_k(0)=i_k]$, the conditional moment generating function of $\Lambda_k(t)$, s$\geq$0 can be obtained as follows:

$$G_{\Lambda_k(t)|Y_k(0)}(s)=E[e^{sR_kY_k(t)}\,|\,Y_k(0)=i_k]=[p_k(t)(e^{sR_k}-1)+1]^{1-i_k}[q_k(t)(e^{sR_k}-1)+1]^{i_k} \quad (8)$$

where $p_k(t)$ and $q_k(t)$ are defined as [7].

$$p_k(t)=\frac{\lambda_k}{\lambda_k+\mu_k}[1-e^{-(\lambda_k+\mu_k)t}] \quad , \quad q_k(t)=\frac{\lambda_k}{\lambda_k+\mu_k}+\frac{\mu_k}{\lambda_k+\mu_k}e^{-(\lambda_m+\mu_k)t}$$

where $p_k(t)$ is the transition probability that a class-m source is active at a future point in time t, given that the source is idle at time 0. $q_k(t)$ is the transition probability that a class-m source is active at a future point in time t, given that the source is active at time 0. Let $\Lambda_k(t)$ be the aggregate arrival rate from $Y_k(t)$ active source. C_k is the assumed allocated channel capacity by QAP, using the Maximum MSDU size and TXOP of a STA. When the overhead is not considered, C_k is

$$C_k=\frac{Maximum\_MSDU\_size_k}{TXOP_k-O} \quad \text{(For traditional scheme)} \quad (9a)$$

$$C_{k\_soft}=\frac{Maximum\_MSDU\_size_k\times\xi}{TXOP_{k\_new}-O} \quad \text{(For soft QoS-aware CAC)} \quad (9b)$$

Please recall that O is overhead time. In a bufferless system, packet losses occur when $\Lambda_k(t)$ exceeds the C_{k_soft}. The prediction of $FLR_k(t)$ is given by the radio of mean excess traffic ($OV_k(t)$) and mean traffic load ($A_k(t)$) at time t. The instantaneous frame loss rate, $FLR_{k\_soft}(t)$ is given by:

$$FLR_{k\_soft}(t)=\frac{OV_k(t)}{A_k(t)}=\frac{E[(\Lambda_k(t)-C_{k\_soft})^+]}{E[\Lambda_k(t)]}=\frac{E[(\Lambda_k(t)-C_{k\_soft})^+\,|\,Y_k(0)=i_k]}{E[E[\Lambda_k(t)\,|\,Y_k(0)=i_k]]} \quad (10)$$

In above equation, FLR of soft QoS is obtained by C_k to C_{k_soft}. As seen in this section, it is difficult to apply these results to a real-time call admission control scheme for wireless LANs. They need a matrix inversion and convolution that involve heavy computation. Approximation and bound approaches for a QoS measure, QoS_{FLR}, are also needed. The central limit approximation approach [8] is used to get following equation:

$$FLR_{k\_soft}(t) = (1 - \frac{A_k(t)}{C_{k\_soft}})Q(\frac{C_{k\_soft} - A_k(t)}{\sqrt{\sigma_k{}^2(t)}}) + \frac{1}{\sqrt{2\pi}}\frac{\sigma_k(t)}{A_k(t)}e^{[\frac{-(C_{k\_soft}-A_k(t))^2}{2\sigma_k{}^2(t)}]} \tag{11}$$

where Q(.) denotes the Q-Function. If the traffic classes of all STAs are the same, the mean FLR of all STAs is the following:

$$FLR_{mean}(t) = \frac{\sum_{k=1}^{K+1} FLR_k(t)}{K+1} \le FLR_{QoS} \tag{12}$$

where K means the number of all existing STAs, and K+1 is the new connection request. If $FLR_{mean}(t)$ is satisfied with eq. (12), QAP can accept new call. And using Eq. (12), the $FLR_{mean}(t)$ of the network can be observed. In the following section, the $FLR_{mean}(t)$ of the traditional scheme is compared with the proposed scheme.

4 Numerical Results

To evaluate the performance of the proposed scheme, voice and video traffics are considered. And the sizes of QoS-ACK and QoS-CFPoll frames in the table include the sizes of the MAC Header and the CRC overhead only. According to the draft standard, the PLCP (Physical Layer Convergence Protocol) Preamble and header are transmitted at Minimum Physical rate to ensure that all STAs can listen to these transmissions regardless of their individual data rates. Thus, t_{PLCP} is constant at 192μs if minimum physical rate in the analysis is 2Mbps, and the t_{PLCP} is reduced to 96μs. The transmission time for different headers and per-packet overhead is summarized in Table 1,2,3.

Table 1. PHY and MAC Parameters

Parameter	
MAC Header size	32bytes
CRC size	4bytes
QoS-ACK frame size	16bytes
QoS-CFPoll frame size	36bytes
PLCP Header length	4bytes
PLCP Preamble length	20bytes
PHY rate (R)	11Mbps
Minimum PHY rate (R_{min})	2Mbps
SIFS	10μs

Table 2. Transmission Time for Different header and per-pascket overhead

Parameter	
PLCP Preamble and Header(t_{PLCP})	96μs
Data MAC Header(t_{HDR})	23.2727μs
Data CRC(t_{CRC})	2.90909μs
ACK frame (t_{ACK})	107.63636μs
QoS-CFPoll (t_{Poll})	122.1818μs
Per-Packet Overhead (O)	249.81818μs

Table 3. TXOP and profit of each call

Parameter	Voice	Video
Mean Data rate	24Kbps	1000Kbps
Nominal MSDU Size	60bytes	1250bytes
TXOP	0.824	11.962
Profit	72.82	104.4976
TXOP'	0.7788(ξ=0.9)	8.485(ξ=0.7)
Profit'	71.9	103.1232

The FLR of soft QoS will be compared with the FLR of a simple scheduler, using the transient fluid model.

The parameters in Tables 1, 2, and 3 are used, assuming that the Frame Loss Rate requirement is 0.01[4]. All traffic is either voice or video. Each STA has the same value of λ and μ of the traffic class. The λ of voice is assumed as 0.5 and the μ of

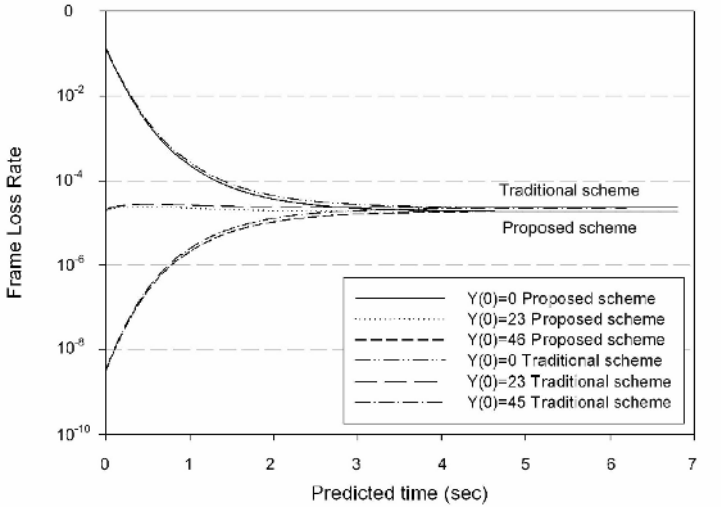


Fig. 3. Predicted FLRmean(t) (When all traffics are voice and video, and the new call is voice)

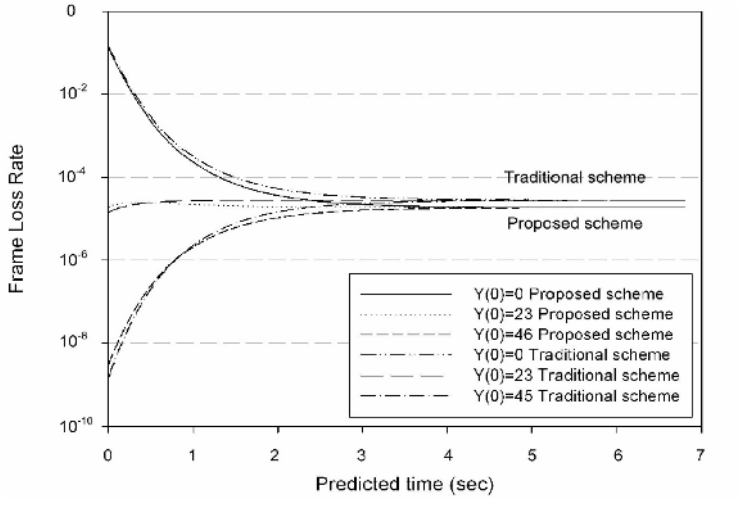


Fig. 4. Predicted FLRmean(t) (when all traffics are voice and video, and the new call is video)

voice is assumed as 0.833, while the λ of video is assumed as 0.5 and the μ of video is assumed as 0.335. The following figures present the numerical results.

In Fig. 3, 4, the predicted $FLR_{mean}(t)$ as a function of the prediction time for the various values of the initial conditions, Y(0), are shown. After observing for approximately seven (7) seconds, the predicted $FLR_{mean}(t)$ will converge to the steady state value, $FLR_{mean}(\infty)$. The different initial conditions of each traffic class were observed. In Fig. 3, 4, there are 44 voice traffics and one video traffic. In results, two schemes are satisfied FLR_{QoS} (0.01). But FLR of proposed scheme is smaller than $FLR_{mean}(t)$ of traditional scheme. Because C_k is lager than C_{k_soft} from eq. (9a), (9b). In Fig. 3, a new call is voice traffic, and the ξ of the voice traffic is 0.9. In Fig. 4, a new call is video traffic, and the ξ of the video traffic is 0.7. So the C_{k_soft} of the voice traffic is larger than the C_{k_soft} of the video traffic. Therefore, the decrement of the $FLR_{mean}(t)$ of video traffic is larger than the $FLR_{mean}(t)$ of voice traffic. Therefore, proposed scheme is excellent candidate for real time call admission control in IEEE 802.11e. Especially for voice traffic, performance of proposed scheme is excellent.

5 Conclusion

A new admission control scheme for HCCA is proposed. The proposed scheme uses the soft QoS-based Greedy Approach, so QAP can accept more STAs. To evaluate the performance of the proposed scheme, the transient Frame Loss Rate, one of QoS measures, was considered. The numerical results showed that the performance of the proposed scheme is better than the traditional scheme. Hence, the proposed admission control scheme is an excellent candidate for real-time call controls in WLAN.

Acknowledgment

This work was supported by the KOSEF, through grant no. R08-2003-000-10922-0, and by the University IT Research Center (INHA UWB-ITRC), Korea.

References

1. Deyun Gao, Jianfei Cai, "Admission Control in IEEE 802.11e Wireless LANs," *IEEE Networks*, July 2005.
2. Antonio Grilo, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks," *IEEE Wireless Communications*, June 2003.
3. IEEE 802.11e/D11.0, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements," October 2004.
4. Wing Fai Fan, "Admission Control for Variable Bit Rate Traffic in IEEE 802.11e WLANs," *10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications, IEEE*, 2004.
5. Daniel Reininger, "Soft QoS Control in the WATMnet Broadband Wireless System," *IEEE Personal Communications*, February 1999.
6. Richard Neapolitan, "Foundations of Algorithms: Using C++ pseudo-code, 2nd Edition," *Jones and Bartlett*, 1998.
7. Yeong M. Jang, "Estimation and Prediction-Based Connection Admission Control in Broadband Satellite Systems," *ETRI Journal*, December 2000.
8. Yeong M. Jang, "Soft QoS-based Vertical Handover Between cdma2000 and WLAN Using Transient Fluid Flow Model," *ICOIN2005, LNCS3391*, February 2005.

A Distributed Mechanism for Trust Propagation and Consolidation in Ad Hoc Networks

Christiane Marie Schweitzer[1,2], Tereza Cristina Carvalho[2], and Wilson Ruggiero[2]

[1] Karlstad University, Computer Science Department, SE-651 88, Karlstad, Sweden
[2] University of São Paulo, PCS - LARC, C1-46, 05508-900, São Paulo, SP, Brazil
{chrism, carvalho, wilson}@larc.usp.br

Abstract. Trust is a person's knowledge of and confidence in another's behavior and reputation. This concept can be applied to ad hoc networks, whose entities interact based on some kinds of relationships of trust in each other. This paper presents a trust model for ad hoc networks, using a specific metric to establish and associate the consolidated trust rating to their entities. A distributed mechanism is defined to propagate and consolidate these trust ratings, enabling the entities of an ad hoc network to decide whether or not to trust other entities with which they intend to interact.

1 Introduction

Wireless networks have been used increasingly in every type of environment thanks to the mobility of their nodes, providing their users with fast and easy access to their data and services anytime and anywhere. As in any other kind of network or communications system, the services and data provided require protection.

Wireless and wired networks have very similar security problems. However, wireless network devices can display an ad hoc behavior, communicating among each other and so presenting new security challenges due to the absence of a central element which is usually responsible for the implementation of most security services. In this case, the network members must protect themselves against security threats and be responsible for enforcing their own security mechanisms and for making decisions about with which nodes they can communicate or interact with. In this context, the trust concept can be used to identify trustworthy transaction partners. The network entities can make decisions about providing or requesting services based on the preestablished relationship of trust.

This paper describes a distributed mechanism for trust propagation and consolidation in ad hoc networks, building trust relationships and making their entities autonomous, thereby enabling them to make decisions independently of a central network element.

This paper is divided into four sections, including the introduction. Section 2 presents the related work in the field of trust computation, while section 3 presents the trust model and proposes and describes a distributed mechanism for trust consolidation and propagation. This section also describes the entities' behavior during the

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 156 – 165, 2006.

execution of this mechanism, and the content and format of trust data to be propagated. Lastly, section 4 makes a final analysis of the proposed mechanisms of trust distribution and consolidation and suggestions for further functionalities and improvements.

2 Related and Prior Work

Security in ad hoc networks, trust and trust relationship concepts have been the object of earlier research and some solutions have been proposed.

A. Jøsang [4] contributed significantly to the definition and characterization of trust in e-commerce environments. His contributions are connected to the context of this work because they are based on the same principle of trust compositions described in Smets [8], and on the same combination rule proposed in Dempster [1] and Shafer [7], which are used to consolidate two pieces of evidence of an entity trust level. Jøsang described and analyzed the transitive and consensus compositions of trust ratings using subjective logic. These compositions were applied to digital certificate chains, and are also applied in this work.

Other papers were also analyzed, such as those of Fernandes et al. [3] and English et al. [2]. These articles presented trust models and trust relationships for network entities, focusing on how entities can protect themselves from security attacks.

In addition, a prior study, which was used as the basis for this one, involves the Security Model and Architecture developed by Martucci et al. [5], and defines the network environment as a service network composed of a communications infrastructure and its components. The network components are entities which can provide or request services. When an entity wishes to join the network, this entity must register at a Certification Authority (CA). This CA issues a digital certificate defining the privileges and service access rights associated to each new entity. The CA also defines how trustworthy the new entity is by giving it an initial trust rating. The certificates and trust ratings are associated with security data and are used to verify if the privileges granted to the entity entitle it to use/provide services in any network.

Entities can have ad hoc behavior, using and providing services to each other. When an entity needs to access a service, it searches for that service through a Lookup Service (LS). The LS is responsible for maintaining a list of available services and for informing all the network entities about the services they are authorized to use and provide. To establish communication between a user entity and a service provider entity, they must first authenticate each other (i.e., exchange certificates). Upon successful authentication, the service provider entity must verify if the user entity is trustworthy to authorize the service, i.e., if the trust rating complies with the trust level required to use the service. If it does, the service provider has three options: it can use the trust rating associated with the identification certificate (initial trust rating); if the user entity has already requested services on previous occasions from the current service provider entity, then it can use the local trust rating; or it can ask for updated trust ratings from other entities belonging to the current network. According to Martucci et al. [5], the CA is responsible for consolidating events, calculating a

new trust rating and distributing it to requesting entities. However, when entities interact ad hoc, they cannot be associated to a centralized entity. Therefore, the entities must find a distributed way to update and disseminate the new trust ratings, which is the main purpose of the current study.

3 Trust Model

The literature contains several reports on trust and on how it can be used in wireless environments. This idea, which we have applied here, is based on human behavior about trust. People often have to decide whether or not to trust someone, even though they may not have sufficient information to make such a decision. Trust usually depends on personal relationships, previous history or information that can be obtained on demand. In a family, different types of relationships involve different levels of trust. For example, one can state that a mother trusts her son, daughter and husband, and that she trusts her husband more than her children. Similarly, her husband trusts her more than his children, and the son trusts his parents than his sister. Starting from this basic knowledge, let us imagine the following situation.

Someone rings the doorbell and the mother opens the door. If she knows the person at the door, she will grant him privileges according to how much she trusts him. These privileges may be: to allow him to enter the hall, living room and/or kitchen, help himself to get a piece of fruit from refrigerator, and/or a variety of other privileges. However, if she does not know him, she may ask someone else in the house if they do and if she can trust the person. Based on the information she is given and on how much she trusts the person who provided that information, she decides to what extent she should trust the visitor. If nobody knows the person at door or she cannot ask anybody, she may decide not to trust him or not to invite him into the house. However, if the caller shows identification ensuring minimal trust, by proving, for example, that he is a gas company employee, she may accept it and authorize him to carry out the service with restricted privileges.

The privileges are associated with trust ratings; therefore, if a person proves to be unreliable, he loses trust and hence, his privileges. Because of the great complexity of human behavior, it is impossible to consider all nuances about trust, so we will stick to the basics of human relationships. Moreover, it is unfortunately common to encounter transgressions, the propagation of such transgressions, and propagation of lies in any environment in which communication takes place between two or more elements. Considering these premises, we have defined a model of trust and entity behavior – trust ratings, operations and the distribution mechanism.

3.1 Trust Ratings

Taking as basis the trust model developed by Martucci et al. [2004], the trust profile associated with each network entity is initially defined based on the type of service provided or used and on how critical the service is in terms of security requirements. In this model, trust ratings represent the trust between two entities and their

behavior. For example, entity A trusts entity B at a given trust level, which is a measurement of the trust A has in this relationship, and its value is between 0 and 1. Dempster [1] and Shafer [7] first came up with this measurement, defining a trust level as a pair (minC, minD), where minC represents the minimal confidence(**c**) and minD represents the minimal distrust (**d**) one entity places in another. This pair is composed of the confidence and distrust ratings, and uncertainty (**u**) is a complementary value embedded in the final trust rating ($c+d+u=1$). The initial values are preconfigured when an entity registers and obtains its identification certificates from a trustworthy entity, the Certification Authority (CA). When the entity registration process succed, the registration service issues a digital certificate which specifies the entity's trust level.

3.2 Trust Operation

The trust ratings can be updated and propagated through three basic operations: simple, transitive and consensus compositions. These operations were initially by Dempster [1] and Shafer [7], then detailed and applied by Martucci et al. [5] (Simple) and Jøsang [4] (Transitive and Consensus).

Simple Composition ($T_R(A) = T_R(A) \Phi T_S(A)$). The trust composition is represented by the symbol Φ and is defined by a combination rule. This composition results from the consolidation of two pieces of evidence about the trust level of the requesting entity. E1 evidence establishes that **R** (Certification Authority) trusts **A** (requesting entity), which is defined by $\Omega_{E1} = \{c_A^R, d_A^R, u_A^R\}$, where c_A^R is the confidence **R** places in **A**, while **R**'s distrust of **A** is defined by d_A^R , and the uncertainty is defined by u_A^R . Similarly, E2 evidence establishes that **S** (Service Provider) trusts **A**.

Transitive Composition ($T_R(A) = T_R(S) \otimes T_S(A)$). The transitive composition is represented by the symbol ($\otimes$). This composition represents the level of trust **R** places in the third entity (**S**) and the level of trust **S** places in **A.** This type of composition is employed when an entity transfers its trust opinion to another entity.

Consensus Composition ($T_{RS}(A) = T_R(A) \oplus T_S(A)$). The consensus operator ($\oplus$) is used to consolidate a single opinion from two possibly conflicting opinions. This composition is adapted to the present work, when entities **R** and **S** each have a trust rating for entity **A** and a single trust rating must be defined (composed) from these opinions, i.e., a consensus of opinions.

3.3 Trust Distribution

Trust ratings must be distributed among all the network's entities. The Security Model specified by Martucci et al. [5] defines the certification authority (CA) as responsible for creating the network and adding new entities, issuing certificates and defining initial trust ratings for these entities, and for updating these ratings.

For purposes of simplification, it is assumed that all the entities received an identification certificate from the same CA. However, ad hoc entities would not depend on a centralized element, not even a CA, for any task.

In such a scenario, an entity must decide how much it will trust another entity, based on local information or recommendations from other entities in the network. Thus, the consolidation of trust ratings and their distribution can be considered a decentralized solution.

Trust Rating Consolidation and Distribution. The main challenge of a decentralized solution is how to discover whether the requesting entity have misbehavior in relation to other entities. The requested entity must ask other entities to compose a trust rating about the requesting entity before communications with the latter are established.

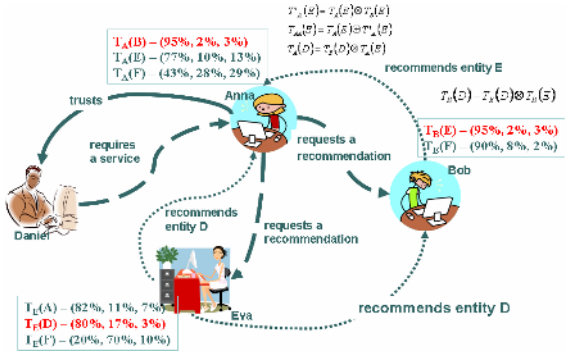


Fig. 1. Network of Client and Server Entities [6]

A new trust rating for an entity is composed by making a sequence of searches at a finite number of the most trustworthy entities. This procedure can be called **maximal trust in recommendations**. Figure 1 illustrates the steps entity **A** takes to compose a trust rating for entity **D**. To this end, entity **A** requests a trust rating of entity **D** from its most trustworthy entity. If the most trustworthy entity (entity **B**) has a trust rating, it returns this value to entity **A**. However, if entity **B** does not have an opinion about entity **D**, then it must recommend its most trustworthy entity – entity **E**, $R_B(E) = T_B(E)$, to entity A. Based on this recommendation, entity **A** will obtain a new opinion about entity **E**, using transitive composition $T'_A(E) = T_A(B) \otimes T_B(E)$. If entity **A** already has an opinion about entity **E**'s trustworthiness, the consensus composition can be used to consolidate both opinions – $T_{AA}(E) = T_A(E) \oplus T'_A(E)$. Thus, entity **A** will obtain a new opinion about entity **E** and will ask entity **E** for its trust rating of entity **D**. If entity **E** has a trust rating for entity **D**, it returns this trust rating to entity **A** and sends a calculated recommendation to both entities **A** and **B**, $R_E(D) = T_E(D)$. Now, entity **A** has an opinion about entity **D**'s trust rating of **E** – $T_A(D) = T_A(E) \otimes T_E(D)$, obtained by transitive composition. This process is repeated until a trust rating is found, or is not found if none of the requested entities have any information to provide. Figure 2 shows the recommendation chain built from the example in Figure 1.

All entities can make recommendations about other entities, preferentially about their most trustworthy entities. An entity making a recommendation must keep a log listing the entities it recommended and to which entities those recommendations were

sent. The logging process must be done to help react against attacks, because when an entity detects an abrupt change in the trust rating of another entity, the other entities can be warned.

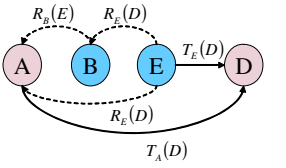


Fig. 2. Recommendation Chain [6]

Thus, summarizing the basic concept involved in the trust consolidation and distribution mechanism, if entity **X** is unable to make a recommendation about entity **Y**, then entity **Z**, which requested a recommendation from entity **X**, must request a recommendation from its next most trustworthy entity, which should not yet have participated in the recommendation process. These steps must be followed exhaustively, through the **maximal trust rating in recommendation**, until a trust rating is calculated, or is found to be impossible to calculate (i.e., no entity can make a recommendation for **Y**). This distribution mechanism converges to a single trust rating – the first trust rating that is found through the chain of most trustworthy entities. The positive or negative events generated (by others) are reported by the entities, which locally update their trust relationships through *Simple Composition* ($T_S'(A) = T_S(A) \Phi T_B(A)$) [5].

Formalization of the Mechanism. The mechanism can be formalized through the entities' identification, the trust data description and the entities' interaction.

The entities are defined by a $X = \{x_1, x_2, x_3, \ldots, x_n\}$ set, where x_i represents each entity of an ad hoc network, and n is a finite number of entities. Each entity in this network has a **Trust Table** ($\forall x_i \in X, \exists TT_i = \{T_1, T_2, T_3, \ldots, T_m\}$), on which the entry $T_j = \{c_{x_j}^{x_i}, d_{x_j}^{x_i}, i_{x_j}^{x_i}\}_{i \neq j}$ is the trust rating that entity x_i has at another entity x_j, and $m \leq (n-1)$ is the number of entities that have already established communications with x_i, which means entity x_i has an opinion about them. Each entity has a **Recommendation Table** ($\forall x_i \in X, \exists RT_i = \{r_1, r_2, r_3, \ldots, r_p\}$), in which the entry $r_k = (\{c_{x_k}^{x_i}, d_{x_k}^{x_i}, i_{x_k}^{x_i}\}, \{x_1, x_2, \ldots, x_m\})$ represents the set of trust recommendations made by entity x_i about entity x_k.

An entity must participate in the recommendation process only once in a trust update cycle. It is therefore necessary to know which entities have participated in the update. Theses entities are stored in a **General Recommendation History** table ($RH = \{x_1, x_2, x_3 \ldots, x_l\}$). Locally, each entity must have the control over the entities from which it has received recommendations, keeping a **Local Recommendation History** table ($\forall x_i \in X, \exists LH_i = \{x_1, x_2, x_3, \ldots, x_p\}$), where x_k, which represents an entity that participated in the recommendation process, must satisfy the relation $(x_k \in TT_i) \wedge (x_k \in RH)$. In order to a control the algorithm life cycle, we have defined i as the current entity that participates in the trust distribution process, where $i = 1,2,3,\ldots,n$. The algorithm uses the following nomenclature:

i. $x_i \rightarrow x_{i+1}(y)$ [i]. Entity x_i asks entity x_{i+1} for its trust rating of entity y.

ii. $(x_{i-1} \leftarrow x_i(x_{i+1})) = T_{x_{i-1}}(x_{i+1})$ [ii]. Entity x_i sends to entity x_{i-1} a recommendation regarding x_{i+1}. The final trust rating $T_{x_{i-1}}(x_{i+1})$ of this recommendation is determined through procedures defined as follows:

- $T_{x_{i-1}}(x_{i+1}) = T_{x_{i-1}}(x_i) \otimes T_{x_i}(x_{i+1})$, where entity x_{i-1}'s trust in entity x_{i+1} is composed by means of *Transitive Composition*, using as inputs entity x_{i-1}'s trust in entity x_i and entity x_i's trust in entity x_{i+1}.

If $T_{x_{i-1}}(x_{i+1}) \notin TT_{i-1}$, i.e., x_{i-1} does not yet have an opinion about entity x_{i+1}, the recommendation is the trust rating $T_{x_{i-1}}(x_{i+1}) = \{c^{x_{i-1}}_{x_{i+1}}, d^{x_{i-1}}_{x_{i+1}}, i^{x_{i-1}}_{x_{i+1}}\}$ and the trust table of entity x_{i-1} receives an entry about entity x_{i+1}, $TT_{i-1} = TT_{i-1} \cup T_{x_{i-1}}(x_{i+1})$.

However, if $T_{x_{i-1}}(x_{i+1}) \in TT_{i-1}$, i.e., x_{i-1} has an opinion about entity x_{i+1}, the previously calculated composed trust rating (by Transitive Composition) $T_{x_{i-1}}(x_{i+1})$ must be updated through a new composition which includes the trust rating $T'_{x_{i-1}}(x_{i+1})$ which x_{i-1} has given x_{i+1} (by *Consensus Composition*), $T_{x_{i-1}}(x_{i+1}) = T'_{x_{i-1}}(x_{i+1}) \otimes T_{x_{i-1}}(x_{i+1})$. Therefore, the recommendation is the final trust rating $T_{x_{i-1}}(x_{i+1}) = \{c^{x_{i-1}}_{x_{i+1}}, d^{x_{i-1}}_{x_{i+1}}, i^{x_{i-1}}_{x_{i+1}}\}$ and the trust table x_{i-1} receives an entry x_{i+1}, $TT_{i-1} = TT_{i-1} \cup T_{x_{i-1}}(x_{i+1})$.

iii. $x_{i+1} \Leftarrow \max Trust(x_i)$ [iii]**.** Entity x_{i+1} receives the x_i maximal trust element that has not yet participated in the recommendation process, i.e. $x_i \notin RH$; updating the **Local Recommendation History** $LH_i = LH_i \cup \{x_{i+1}\}$.

The entities behavior is defined basically according to three phases: **Local Request** (LR), **Chain Remote Request** (CR) and **Trust Table Request** (TR_T). Thus, if an entity $y \in X$ requests services from an entity $z \in X$, entity z must get the updated trust rating concerning y, following the procedures described below:

Initially, one checks if z contains a trust rating on y. If it does not, then $x_i \leftarrow z$ and entity x_i must research for a new trust rating.

If $y \notin TT_i$ **then**, $RH = RH \cup \{x_i\}$, x_i **is an RH entry**.

If at least one entity has not yet participated in the recommendation and no trust rating for entity y has been found, the mechanism continues.

1. **If** $X \not\subset RH$ **then**

In this step, the search for entity x_i's most trustworthy entity is done, x_{i+1}, and is added to the general and local recommendation history:

---

[i] $\rightarrow$: trust rating request.

[ii] $\leftarrow$: trust rating recommendation.

[iii] $\Leftarrow$: Attribution of the maximal trust element: the most trustworthy entity has the highest confidence value, independently of its uncertainty or distrust ratings. If both entities have the same confidence value, the most trustworthy entity has the lowest distrust rating.

a. $x_{i+1} \Leftarrow \max Trust(x_i)$, **where** $(x_{i+1} \in TT_i) \wedge (x_{i+1} \notin RH)$

b. $RH = RH \cup \{x_{i+1}\}$

First, this entity x_{i+1} receives a **Local Request** about trust in entity y. If x_{i+1} has a trust rating, this value is used to compose a recommendation for entity z.

2. **If** $y \in TT_{i+1}$ **then** $z \leftarrow x_{i+1}(y)$, $r = (z \leftarrow x_{i+1}(y))$, $RT_{i+1} = RT_{i+1} \cup r$, **end**.

If x_{i+1} does not have a trust rating for entity y, a chain of searches, called **Remote Request Chain**, is performed through other entities.

3. **If** $y \notin TT_{i+1}$ **then** $x_{i-1} \Leftarrow x_i$ and $x_i \Leftarrow x_{i+1}$

In this chain of requests, each new entity in the chain is the most trustworthy entity of the one that recommended it.

a. $x_{i+1} \Leftarrow \max Trust(x_i)$, **where** $(x_{i+1} \in TT_i) \wedge (x_{i+1} \notin RH)$

This new entity is added to the **Local Recommendation History**, in step 3, and must be added to the **General Recommendation History.**

b. $RH = RH + x_{i+1}$,

This entity x_{i+1} will be an entity x_i recommendation (**Trust Table Request**) for entity x_{i-1}, updating the trust table of x_{i-1}, and the recommendation table of x_i.

c. $x_{i-1} \rightarrow x_i(x_{i+1})$ **and** $x_{i-1} \leftarrow x_i(x_{i+1})$, $z \leftarrow x_i(x_{i+1})$,
$r = (x_{i-1} \leftarrow x_i(x_{i+1}))$, $RT_i = RT_i \cup \{r\}$.

If all the most trustworthy entities on x_i's trust table participated in the recommendation process, backtracking is used to form a new chain through the next most trustworthy x_{i-1} entity, and x_{i-1} becomes entity x_i.

d. **If** $LH_i \subseteq RH$ **then** $x_{i+1} \Leftarrow LH(l-1)$, $x_i \Leftarrow LH(l-2)$, $x_{i-1} \Leftarrow LH(l-3)$

If all the entities have participated in the recommendation process and no trust rating was found for y, the trust rating is defined as null.

e. **If** $X \subseteq RH$ **then** $z \leftarrow null$, **end**.

If any entity has not yet participated in the recommendation process, the process is repeated from step 2.

f. Return to step 2.

3.4 Proof of Concept

A prototype of the mechanism was implemented and some simulations were performed. For example, we simulated a network with 8 nodes in a local environment, in which each entity had a set of trustworthy entities and requested trust ratings for other entities. Table 2 presents the entities' behavior when entity **E** requested trust information on entity **G**. In this case, the **Trust Table** was updated with the information on the recommended and requested entities. The first column lists the network entities. Each entity has an initial **Trust Table** (**Before** column). The third column (**After**) identifies the entities that participated in the recommendation process and had their trust ratings updated after a trust request.

Other simulations showed that, after successive trust queries about other entities, new chains were formed and new recommendations were made. This process resulted in a new trust consolidation and updated the participating entities, converging toward homogeneous values. And, when numerous negative events occurred, reducing the trust ratings of an entity, that entity's privileges were reduced, it was blocked and its certificate was revoked, all without the participation or intervention of a centralized entity.

Table 1. E requests trust value about **G** [6]

	Before	**After**
B	H: (0.95, 0.02) F: (0.90, 0.08) C: (0.89, 0.03) D: (0.87, 0.08) E: (0.85, 0.1) A: (0.82, 0.10)	H: (0.95, 0.02) F: (0.9, 0.08) C: (0.89, 0.03) D: (0.87, 0.08) G: (0.855, 0.076) E: (0.85, 0.1) A: (0.82, ,0.1)
E	F: (0.82, 0.11) B: (0.80, 0.17) D: (0.20, 0.70)	H: (0.9025, 0.019) G: (0.855, 0.076) F: (0.82, 0.11) B: (0.63, 0.001) D: (0.2, 0.7)
G	C: (0.95, 0.02) H: (0.90, 0.08)	C: (0.95, 0.02) H: (0.9, 0.08)
F	E: (0.96, 0.02) B: (0.95, 0.02) D: (0.90, 0.08)	E: (0.96, 0.02) B: (0.95, 0.02) H: (0.9025, 0.019) D: (0.9, 0.08)
H	B: (0.98, 0.01) C: (0.95, 0.02) G: (0.90, 0.08)	B: (0.98, 0.01) C: (0.95, 0.02) G: (0.9, 0.08)

4 Final Evaluation

The entities in ad hoc networks must be responsible for their own security and their own decisions. To this end, these entities can use the trust concept and assess the trust relationships in the network. This work proposes a distributed mechanism to consolidate and propagate trust ratings in ad hoc networks. In the proposed solution, the entities run the proposed mechanism requesting trust information about another entity from the entities that belong to the network. This mechanism looks for the trust rating through successive searches at other trustworthy entities. The search begins from the most trustworthy entities of the requesting entity, and the trust ratings are consolidated and updated based on recommendations received from these entities. Therefore, the distributed and dynamic mechanism allows entities to keep their trust ratings and perceptions about other entities updated.

The main innovative contribution of this work is the proposal of a Distributed Mechanism for Trust Propagation and Consolidation in Ad Hoc Networks, which ensures that no data about meaningful events is lost and no events are treated twice. When an event occurs at one entity, it is consolidated locally by this entity. Once it is consolidated, if a trust rating changes, the new trust rating will be propagated throughout the network. The trust rating is consolidated consistently, independently of a centralized entity and regardless of the number of active nodes. This mechanism proceeds along a chain of the most trustworthy entities. Because every network has a finite number of entities and each entity trusts a finite number of entities, this search is performed within a finite number of chains. If no trust rating is found for the requested entity in any of the search chains, a null response is returned to the entity that initiated the process. In terms of scalability and convergence time**,** the most critical case in terms of convergence occurs when a new entity wishes to participate in the network but no other entity has any information on its trust rating. Considering that there are n entities in the network, the maximum number of search entities will be given by $(n-1)^n$. However, since each entity will provide only one recommendation

about the same entity during a search process, the number of search chains will be reduced to $(n-1)+(n-2)$.

Insofar as the mechanism's security is concerned**,** confidence in the authenticity of a recommendation is achieved because every recommendation must be signed using a certificate issued by a recognized and known Certification Authority (CA). Additionally, recommendation information can be sent through a secure encrypted channel, which ensures the confidentiality and integrity of the information transmitted. This work consisted of building a mechanism to consolidate and distribute trust ratings, which can be applied to any form of interaction among network entities, especially to entities with ad hoc behavior. It is important to point out that the trust model is only one security-enhancing component in wireless environments and that a complete security model must comprise additional security services.

Outlook. Additional simulations are planned for studies in the near future, aimed at assessing the behavior of ad hoc network entities in greater detail. This work can be extended, employing different methods and techniques to assign an initial trust rating to a network entity, and can also be applied to other network environments.

Acknowledgments. The authors thank Ericsson Research and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP-Brazil) for their funding of this work. We are also indebted to the researchers of the Laboratory of Networks Architecture (LARC-EPUSP), whose participation was fundamental to the results described herein.

References

1. Dempster, A. P. "Upper and Lower probabilities induced by a multivalued mapping", Annals of Mathematical Statistics, pp. 325-339, 1967.
2. English, C.; Terzis, S.; Nixon, P. Towards Self-Protecting Ubiquitous Systems: Monitoring Trust-based Interactions. System Support for Ubiquitous Computing Workshop (UbiComp), Journal, Nottingham, England, Set. 2004.
3. Fernandes, A. et al. Pinocchio: Incentives for honest participation in distributed trust management. Second Conference on Trust Management. Proceedings. UK, 2004.
4. Jøsang, A. E. Gray and M. Kinateder. Analysing Topologies of Transitive Trust. Workshop of Formal Aspects of Security and Trust (FAST). Proceedings. Pisa, Italy, Set. 2003.
5. Martucci, L. A.; Schweitzer, C. M.; Venturini, Y. R.; Carvalho, T.C.M.B.; Ruggiero, W. V. "A Trust-Based Security Architecture for Small and Medium Sized Ad Hoc Networks". The Annual Mediterranean Ad Hoc Networking Workshop. Turkey, Jun. 2004.
6. Schweitzer, C. M. "Mecanismo de Consolidação de Confiança Distribuída para Redes Ad Hoc". Doctoral Dissertation, Universidade de São Paulo (USP), São Paulo, Brazil, 2004.
7. Shafer, G. "A Mathematical Theory of Evidence", Princeton University Press, 1976.
8. Smets, P. "What is Dempster-Shafer's model?", Université Libre de Bruxelles, 1990.

A Quality of Relay-Based Routing Scheme in Multi-hop Cellular Networks

Ming-Hua Lin[1] and Kuen-Liang Sue[2]

[1] Department of Information Management, Shih Chien University,
70 Ta-Chih Street, Taipei 10462, Taiwan
mhlin@mail.usc.edu.tw
[2] Department of Information Management, National Central University,
300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan
klsue@mgt.ncu.edu.tw

Abstract. Discovering an available relaying path is a critical prerequisite for the success of the multi-hop cellular networks. Since forwarding data for others utilizes the resources of the mobile nodes such as battery energy, link bandwidth, buffer space and processing time, the mobile nodes may accept only a certain number of relaying requests. In this paper, we propose a Quality of Relay (QoR)-based routing scheme to select a routing path between a mobile node and the central base station based on the individual importance of each intermediate node contributing to hop-by-hop connections. The proposed routing scheme can retain more valuable resource for later relaying requests, thereby supporting more connections successfully. Simulation results indicate that the proposed routing scheme causes a lower new call blocking probability than the shortest-path routing scheme under a certain constraint on maximum relaying capacity of each mobile node.

1 Introduction

Multi-hop cellular networks that integrate the characteristics of both cellular and mobile ad hoc networks have received increasing attention in recent years. Several benefits have been investigated from this new family of networks [1,2,3,4]: (i) reducing the number of the fixed antennas; (ii) conserving the energy consumption of the mobile device; (iii) reducing the interference with other mobile nodes; (iv) enhancing the service coverage of the network; (v) increasing the capacity of the cell. Figure 1 indicates the scenario of general multi-hop cellular networks, the service area of the cellular networks can be extended by adopting hop-by-hop connections at the boundaries of the cell.

In the hybrid networks, the communication between the mobile node and the base station is relayed by a number of other mobile nodes. Therefore, discovering an available relaying path is a critical prerequisite for the success of the multi-hop cellular networks. Most of the multi-hop cellular networking models do not have a pre-constructed routing topology for forwarding packets. The mobile nodes find relaying paths when they desire the path to transmit data. Although the shortest path is the most simple and common metric used in the routing protocol, it may

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 166–174, 2006.

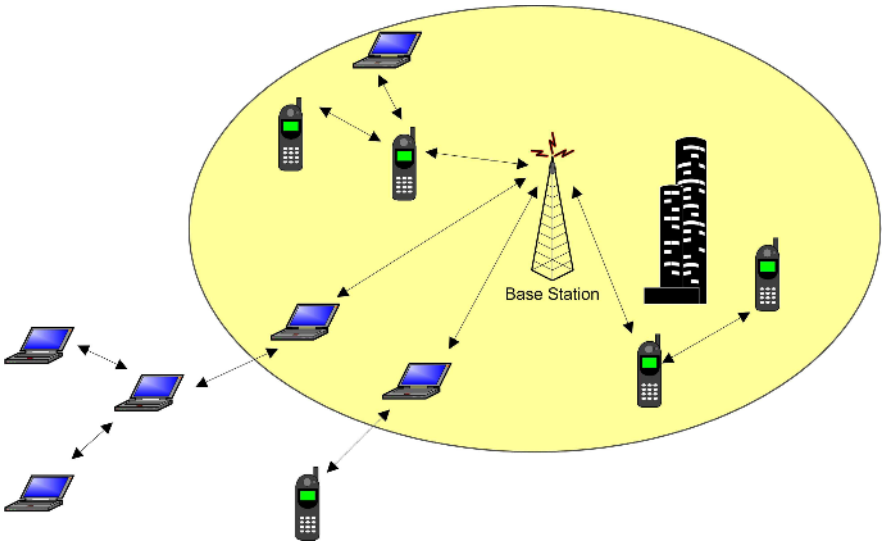


Fig. 1. Scenario of general multi-hop cellular networks

route almost packets over a few (shortest-distance) paths and result in network congestion and resource unavailable in hot spot [5].

Since forwarding data for others utilizes the resources of the mobile nodes such as battery energy, link bandwidth, buffer space and processing time, the mobile nodes may accept only a certain number of relaying requests. In this paper, we propose a Quality of Relay (QoR)-based routing scheme to select a routing path between a mobile node and the central base station according to the individual importance of each intermediate node supporting hop-by-hop connections. The importance of each mobile node is evaluated by a metric which represents the relaying capability of the node. The proposed routing scheme can retain more valuable resource for later relaying requests, thereby supporting more connections successfully. Simulation results indicate that the proposed QoR-based routing scheme causes a lower new call blocking probability than the shortest-path routing scheme under a certain constraint on maximum relaying capacity of each mobile node.

The rest of this paper is organized as follows. In section 2, we review the routing schemes in existing multi-hop cellular networking models. Section 3 describes the detail of the proposed QoR-based routing scheme. Section 4 presents the simulation results and discussions. Finally, concluding remarks are recommended in section 5.

2 Literature Review

Most of the multi-hop cellular networking models do not have a pre-constructed routing topology for relaying packets. The mobile nodes find relaying paths when

they desire the path to transmit data depending on different criteria, such as signal strength, path length and power consumption [6]. Some research has reviewed and investigated existing multi-hop routing protocols [7,8,9]. The routing scheme in multi-hop cellular networks is similar to the routing approach adopted in pure ad hoc networks.

Opportunity Driven Multiple Access (ODMA) is an ad hoc multi-hop protocol that the transmissions from mobile hosts to the base station are broken into multiple wireless hops, thereby reducing transmission power [1,10]. Previous work on ODMA uses path loss between terminals as the metric to determine the routing path. From the list of relaying routes available, the one with minimum aggregate transmit power along the route is selected.

Aggélou et al. describe an Ad Hoc GSM (A-GSM) system that presents a network layer platform to accommodate relaying capability in GSM cellular networks [11]. In A-GSM system, handover is initiated by a mobile node when a high possibility exists that the call will be lost or the quality of the ongoing connection seriously degraded. If multiple neighboring nodes are available for a mobile node to build a relaying path to the base station, the mobile node selects a relaying link with strongest signal to initiate handover.

Qiao et al. present a network model called iCAR that integrates the cellular infrastructure and ad-hoc relaying technologies [2]. In iCAR system, a number of specific stations called ARS's are deployed at strategic locations to relay data from the congested cell to the neighboring non-congested cells. Each ARS collects neighbor information and maintains a routing table containing one entry for every reachable Base Transmission Stations (BTS). Whenever a mobile node needs a relaying path to one or more BTS's which have free channel available, the mobile node chooses the best ARS with the shortest path or the lowest power consumption as the proxy ARS to relay data to the target BTS.

Wu et al. propose a scheme called Mobile-Assisted Data Forwarding (MADF) to add an ad-hoc overlay to the fixed cellular infrastructure and special channels are assigned to connect users in a hot cell to its neighboring cold cells [12]. In MADF approach, the user in a hot cell selects a forwarding agent according to the quality of the signal and the traffic load.

3 Proposed QoR-Based Routing Scheme

The common approach in most existing routing protocols is to consider the shortest-path routing. For simplicity, these protocols measure the distance of the path by the number of hops in the path. However, routing packets based on minimum hop count may take a considerable time to reach the destination because almost packets may be routed over a few (shortest-distance) paths in the networks [5]. Besides, forwarding data for others utilizes the resources of the mobile nodes such as battery energy, link bandwidth, buffer space and processing time. Consequently, each mobile node may accept only a certain number of relaying requests. For example, in A-GSM system [11], a protocol parameter

"relaying capacity" is used to tell neighboring nodes the number of calls a mobile node can simultaneously relay.

In this section, we present a routing scheme to select a relaying path based on the individual importance of each intermediate node contributing to hop-by-hop connections. The proposed scheme aims to retain the nodes with more valuable relaying capability for later relaying requests, thereby setting up more connections successfully in the networks. Herein we focus only on a single base-station cell as indicated in Fig. 2. The base station can enhance the service coverage by adopting relaying connections supported by the mobile nodes.

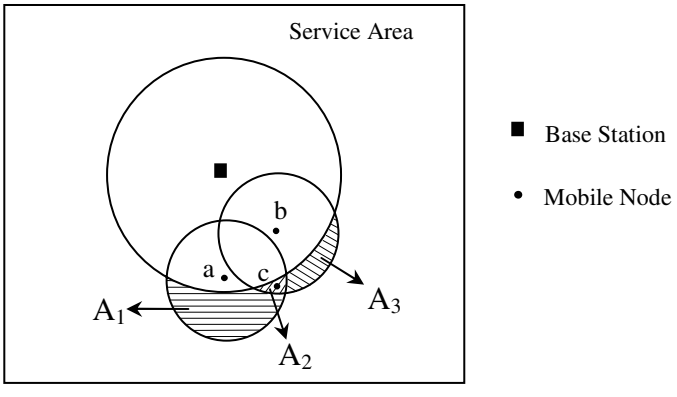


Fig. 2. An example of multi-hop cellular networks with a single base station

A node with higher importance means it can provide relaying services for larger area or it can provide relaying services for some nodes that other can not. In order to evaluate the degree of a mobile node contributing to relaying connections in the multi-hop cellular networks, we introduce a new metric called Quality of Relay (QoR) as follows:

$$QoR_v = \sum_{i \in C_v} \frac{1}{RI_i}. \tag{1}$$

C_v is the area inside the coverage of a node v where the mobile node requires hop-by-hop connections to reach the base station, RI_i is the relay index (RI) of position i that is defined to be the number of mobile nodes capable of relaying traffic for a mobile node staying in position i. As the example indicated in Fig. 2, $RI_{i \in A_1}$ is 1 because only node a can relay data for the mobile nodes reside in area A_1; $RI_{i \in A_2}$ is 2 because both node a and node b can relay data for the mobile nodes reside in area A_2. Since networking services provided by the base station are available when the mobile nodes can connect to the base station successfully, the service availability of the whole relaying networks is the probability that the mobile nodes can use relaying connections to reach the base station. The

degrees of node a and node b contributing to the service availability of networks are evaluated by their QoR values as follows:

$$QoR_a = \sum_{i \in C_a} \frac{1}{RI_i} = \sum_{i \in (A_1 \cup A_2)} \frac{1}{RI_i} = (A_1 * \frac{1}{RI_{i \in A_1}}) + (A_2 * \frac{1}{RI_{i \in A_2}})$$
$$= A_1 * \frac{1}{1} + A_2 * \frac{1}{2}, \tag{2}$$

$$QoR_b = \sum_{i \in C_b} \frac{1}{RI_i} = \sum_{i \in (A_2 \cup A_3)} \frac{1}{RI_i} = (A_2 * \frac{1}{RI_{i \in A_2}}) + (A_3 * \frac{1}{RI_{i \in A_3}})$$
$$= A_2 * \frac{1}{2} + A_3 * \frac{1}{1}. \tag{3}$$

In this study we use the inverse of relay index multiplied by the area because the contribution of a mobile node is based on not only the measure of area it can serve but the rarity of its resource. The value of the inverse of relay index increases as the number of mobile node capable of relaying data for this position decreases. From above equations, QoR_a is greater than QoR_b because the coverage of area A_1 is larger than that of area A_3. There are two conditions that node v has a higher QoR value:

- The node v has larger C_v, which means it can support larger coverage where mobile nodes necessitate hop-by-hop connections to reach the base station.
- The position inside C_v has lower RI value, which means the mobile nodes inside C_v can be supported by fewer nodes. That is, the node v can provide relaying services to the nodes that few nodes can also support.

Consequently, a node with a higher QoR value represents that its resource is more valuable and it has higher relaying capability. Let IM_r be the set of intermediate nodes (the nodes in the route except the source and the destination) in route r, $TQoR_r$ be the sum of the QoR values of all intermediate nodes on the route r, that is,

$$TQoR_r = \sum_{v \in IM_r} QoR_v. \tag{4}$$

Let R_i be the set of routes available from node i to the base station, we select the route with minimum $TQoR$ value as the relaying route to connect node i to the base station. That is, the routing criteria is

$$\min_{r \in R_i} \{TQoR_r\}. \tag{5}$$

A route with lower $TQoR$ value implies that the resource of the nodes along the route are relatively not scare, therefore occupying the resource will make a smaller impact on others. As the example illustrated in Fig. 2, node c has two

choices to route data to the base station: $c \rightarrow a \rightarrow BaseStation$ and $c \rightarrow b \rightarrow BaseStation$. The $TQoR$ values of these two routes, respectively, are as follows:

$$TQoR_{c \rightarrow a \rightarrow BaseStation} = QoR_a, \tag{6}$$

$$TQoR_{c \rightarrow b \rightarrow BaseStation} = QoR_b. \tag{7}$$

In our routing scheme, node c adopts the relaying path $c \rightarrow b \rightarrow BaseStation$ because QoR_b is less than QoR_a and

$$\min_{r \in \{c \rightarrow a \rightarrow BaseStation, c \rightarrow b \rightarrow BaseStation\}} \{TQoR_r\} = TQoR_{c \rightarrow b \rightarrow BaseStation}. \tag{8}$$

According to the definition of QoR in equation (1) and above discussions, a node with a higher QoR value can support larger coverage or provide relaying services to the nodes that few nodes can also support. Consequently, selecting a relaying path with minimum $TQoR$ value first makes more valuable resource retain for later relaying connections or for the mobile nodes that may not be served by others.

In [13], Ahmed et al. propose a method to determine the trajectory of the mobile gateway to serve the ad hoc group to which it belongs. A mobile gateway called Cross-Layer Communication Agent (CCA) provides connection between a group of ad hoc mobile nodes and a range extension network consisting of airborne relay nodes or low earth orbit/geostationart satellite. Each cluster with a CCA and a group of mobile nodes in [13] is an example of the multi-hop cellular networking models mentioned in this paper. Ahmed et al. define the CCA trajectory based on the location, traffic load, etc. Since the position of each node is required for computing the optimal location of the CCA, each node is equipped with a GPS device that enables the node to determine its position. For measuring the QoR value of a node proposed in this paper, each mobile node is also equipped with a GPS device to obtain its position. The mobile node can report which position it can support based on its location and signal strength. The RI value of each position depends on the number of mobile nodes capable of providing relaying services for it. Suppose all nodes participating in routing are able to reach the base station by hop-to-hop connections. After each mobile node transmits the positions it can serve to the central base station, the central base station can compute an optimal route for each request from the received information according to the proposed scheme.

4 Simulation Results and Discussions

The simulation environment is a rectangular region of size 400 units by 400 units with a single base station located in the central point. The radius of the base station is 150 units and the radius of each mobile node is 100 units. Each simulation runs for 100 seconds. The mobile nodes move according to a random waypoint mobility model [5]. Each mobile node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0-12 units/s). Once the destination is reached, another random destination

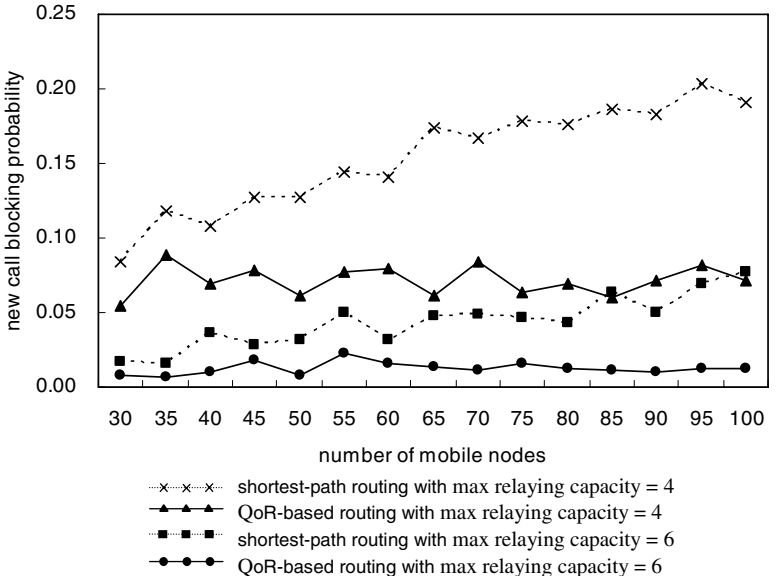


Fig. 3. Comparison of new call blocking probability by QoR-based routing and shortest-path routing under different number of mobile nodes

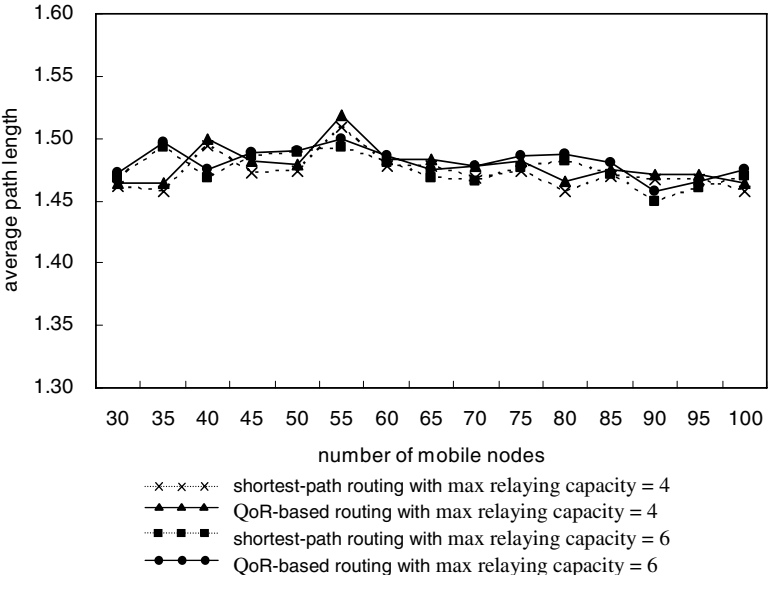


Fig. 4. Comparison of average path length by QoR-based routing and shortest-path routing under different number of mobile nodes

is targeted after a 10-second pause time. The arrival of new data transmission requests initiating in each mobile node forms a Poisson process with rate $\lambda = 0.2$ calls/second and the data transmission times are exponentially distributed with mean 10 seconds.

We compare the proposed QoR-based routing scheme with the shortest-path routing scheme in new call blocking probability. In our simulation, the shortest path is determined by the relaying path with minimum hop count. In order to observe the performance of different routing schemes under limited relaying resource, each mobile node only accepts a certain number of relaying requests. Herein the maximum relaying capacity for each mobile node is set as 4 or 6. If a mobile node desires to transmit data to the central base station but no relaying path is available or some node in the relaying path has accepted maximum relaying requests, then the new call blocking probability is 1.

In Fig. 3, we observe that the QoR-based routing scheme provides a lower new call blocking probability than the shortest-path routing scheme under various number of mobile nodes and different maximum relaying capacity. Since the mobile nodes do not accept more relaying requests than its limited capacity, the new call blocking probability is higher under the maximum relaying capacity of 4 than the maximum relaying capacity of 6. Moreover, we can find the new call blocking probability increases as the number of mobile nodes increases by adopting the shortest-path routing scheme in Fig. 3. This implies that routing based on minimum hop count may cause packets to route over a few paths, therefore resulting in network congestion and resource unavailable in hot paths. The condition is more significant when numerous nodes exist in the networks. In Fig. 4, we find that the average path length in the QoR-based routing scheme is longer than that in the shortest-path routing scheme since the shortest-path routing scheme has minimum hops to reach the base station. However, the difference is not obvious.

5 Conclusions

In this paper, we propose a QoR-based routing scheme to select a routing path between a mobile node and the central base station based on the individual importance of each intermediate node contributing to hop-by-hop connections. The proposed routing scheme can retain more valuable resource for later relaying requests, thereby supporting more relaying connections successfully. Simulation results indicate that the proposed QoR-based routing scheme causes a lower new call blocking probability than the shortest-path routing scheme under a certain constraint on maximum relaying capacity of each mobile node.

Acknowledgment

This research was supported by the National Science Council, Taiwan, R.O.C., under the contract NSC94-2416-H-158-003.

References

1. Rouse, T., Band, I., McLaughlin, S.: Capacity and Power Investigation of Opportunity Driven Multiple Access (ODMA) Networks in TDD-CDMA Based Systems, Proc. of IEEE ICC, pp.3202-3206, Apr. 2002.
2. Qiao, C., Wu, H.: iCAR: an Intelligent Cellular and Ad-hoc Relay System, Proc. of IEEE IC3N, pp.154-161, Oct. 2000.
3. Jakobsson, M., Hubaux, J.P., Buttyán, L.: A micropayment scheme encouraging collaboration in multi-hop cellular networks, Proc. of Financial Crypto 2003.
4. Ben Salem, N., Buttyán, L., Hubaux, J.P., Jakobsson, M.: A Charging and Rewarding Scheme for Packet Forwarding in Multi-Hop Cellular Networks, Proc. of ACM MOBIHOC, pp.13-24, Sep. 2003.
5. Sheu, S.T., Chen, J.H.: Novel Delay-Oriented Shortest Path Routing Protocol for Mobile Ad Hoc Networks, Proc. of IEEE ICC, pp.1930-1934, Jun. 2001.
6. Singh, S., Woo, M., Raghavendra, C.S.: Power-Aware Routing in Mobile Ad Hoc Networks, Proc. of ACM/IEEE International Conference on Mobile Computing and Networking, pp.181-190, Oct. 1998.
7. Royer, E.M., Toh, C.K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks, IEEE Personal Communications, pp.46-55, Apr. 1999.
8. Safwat, A., Hassanein, H.: Infrastructure-based Routing in Wireless Mobile Ad Hoc Networks, Journal of Computer Communications 25, pp.210-224, 2002.
9. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proc. of ACM MOBICOM, pp.85-97, Oct. 1998.
10. 3G TR 25.924 V 1.0.0. 3GPP TSG-RAN; Opportunity Driven Multiple Access, Dec. 1999.
11. Aggélou, G.N., Tafazolli, R.: On the Relaying Capacity of Next-Generation GSM Cellular Networks, IEEE Personal Communications, pp.40-47, Feb. 2001.
12. Wu, X., Chan, S.H., Mukherjee, B.: MADF: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure, Proc. of IEEE WCNC, pp.549-554, Sep. 2000.
13. Ahmed, M., Krishnamurthy, S., Katz, R., Dao S.: Trajectory control of mobile gateways for range extension in ad hoc networks, Computer Networks 39(6), pp.809-825, 2002.

Ad Hoc Sensor Networks

Optimal Transmission Range for Topology Management in Wireless Sensor Networks*

Jongmin Shin, Miae Chin, and Cheeha Kim

Department of Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH)
San 31 HyoJa-Dong, Nam-Gu, Pohang, 790-784, Korea
{sjm0621, zambang, chkim}@postech.ac.kr

Abstract. Topology management protocols aim to minimize the number of nodes in densely distributed sensor networks that constitute the topology. Most protocols construct the topology with a fixed transmission range (i.e. maximum transmission power). With respect to the power consumed by data transmission, it is more efficient to communicate with shorter transmission range, even with the number of nodes in the topology increased. In the sense of minimizing the total power consumed by the topology management as well as data transmissions, it is necessary to find an optimal transmission range. Moreover, data rate is also an important factor to determine the total transmission energy. In this paper, we obtain the optimal transmission range for a given data rate and maximum transmission power to minimize the total power consumption.

1 Introduction

In recent years, the availability of micro-sensors and low-power wireless communications has enabled the deployment of densely distributed sensor networks for a wide range of applications [1]. Sensor networks normally consist of a large number of distributed nodes that organize themselves into a multi-hop wireless network. It is often impossible to change or recharge batteries for these nodes, since sensor nodes are commonly small and distributed in a large area [8]. Therefore, energy conservation is crucial in extending the lifetime of a sensor network. Research efforts on energy conservation can be classified into three dominant approaches.

The first approach uses data reduction, by which correlations in data are exploited to reduce the size of data and unnecessary transmissions. All sensing data from every node are not necessarily delivered to users [23]. The second approach concerns finding an energy efficient routing path for forwarding the data packet to the sink, which reduces control overhead and maintaining cost of the route [3,4]. The last approach utilizes the low-power sleep mode supported by

* "This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)" (IITA-2005-C1090-0501-0018).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 177–185, 2006.

the wireless devices. The sleep power is usually one to four orders of magnitude less than the power required in listening state [17]. Energy conservation can be achieved if a sensor node is allowed to sleep as much as possible when it is not engaged in communication. This approach can exploit the sleep mode property either in time dimension or in density dimension. The former is embedded in the MAC layer through a fine grained control to switch the wireless radio on and off. Sensor nodes only need to be awake only for forwarding. IEEE 802.11 based power saving protocols [7], S-MAC [8], STEM [9], and T-MAC [10] are some examples. The latter, called topology management, focuses on constructing a backbone topology with minimum connected dominating set of the network [11]. If all deployed nodes involved in forwarding, the system will expend unnecessary energy and nodes may interfere with one another by congesting the channel [1]. By topology management, an energy efficient topology is employed in multi-hop wireless communication [15]. GAF [12], SPAN [13], and ASCENT [1] are examples. Existing protocols construct the topology with a fixed transmission range (i.e. maximum transmission power) for each node to minimize the number of nodes that constitute the topology. The nodes in the topology should be awake for the connectivity of the network, and the other nodes are in sleep state to reduce energy consumption.

With respect to the power consumed by data transmission, it is more efficient to communicate with shorter transmission range, even with the number of nodes in the topology increased. In other words, energy savings in data transmission can be achieved through the use of multiple short range transmissions as opposed to one large range transmission. However, this property may require the significant overhead of topology management. In the sense of minimizing the total power consumed by the topology management as well as data transmissions, it is necessary to find an optimal transmission range. Moreover, data rate generated by sensing nodes is also an important factor to determine the total transmission energy. In this paper, we obtain the optimal transmission range for a given data rate and maximum transmission power to minimize the total power consumption. Our analysis is based on random deployment since this deployment strategy is easy and cheap. Our analytical results will benefit the research on controlling the transmission range with respect to the data rate in sensor networks.

The rest of the paper is organized as follows. In section 2, we review principles of topology management from the energy saving point of view, discuss the energy consumption in multi-hop communication, and present the concept of data rate. In section 3, we compute the optimal transmission range for topology management with appropriate assumptions. Finally, concluding remarks are in Section 4.

2 Preliminaries

2.1 Energy Model

In this section, we define the energy model used in our analysis. Deployed sensor nodes use the energy in order to sense, collect, process and disseminate the

data to the sink in wireless sensor networks. In the sensor node architecture, the energy related subsystems are comprised of three parts [17]. First, the micro processor or microcontroller unit (MCU) is responsible for control of the whole sensor node and spends most of the energy in data processing and signal processing. It supports various operating modes, including Active, Idle, and Sleep modes, for power management. Second, the radio unit enables wireless communication with neighboring nodes. In general, radios can operate in one of four distinct modes; Transmit, Receive, Idle, and Sleep. Third, the sensing unit measures environmental parameters such as temperature, light intensity, sound, magnetic fields, etc. We assume that the power mode of MCU and the sensing unit is always on. That is, only radio energy consumption is controlled. The energy consumption in an operating mode is defined as follows:

- $P_{sleep}(mW)$: the power consumption in radio-off state or in Sleep mode. It is the microcontroller power consumed in the active mode plus the sensing power.
- $P_{idle}(mW)$: the power consumption in radio idle mode. A node in Idle mode is ready to receive or transmit data at any time. The amount is obtained by adding the needed power to keep the radio on to P_{sleep}.
- $P_{Rx}(k)(mW)$: the power consumed by a node receiving data in k bits per second. $P_{Rx}(k) = P_{idle} + E_{Rx-elec} \cdot k$, where $E_{Rx-elec}$ is assumed to be equal to E_{elec} [18].
- $P_{Tx}(k, d, \gamma)(mW)$: the power consumed by a node sending in k bits per second. It is related to the transmission distance d. $P_{Tx}(k, d, \gamma) = P_{idle} + E_{Tx-elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^{\gamma}$, where $E_{Tx-elec}$ is assumed to be equal to E_{elec}, ε_{amp} is the transmit amplifier power [18]. Typically we consider γ to be 2, 3, or 4 in wireless communication.

2.2 The Number of Active Nodes in Topology Management

In graph theory, the *minimum connected dominating set* (MCDS) problem best describes the topology management schemes. A subset D of vertices in a graph $G = (V, E)$ is called a *dominating set* if any vertex in V either belongs to D or has a neighbor in D. If the sub-graph induced by D is connected, then D is called a *connected dominating set* (CDS). Sensor networks can be modeled using *random geometric graphs* as follows [16]. Nodes are represented by vertices in the corresponding random geometric graph, where the Euclidian distance corresponds to the transmission range of sensor nodes.

N nodes are randomly (uniform distribution) and densely deployed in a two dimensional region, and they cover the entire sensing domain A with the sensing range R_s. We suppose N_{active} nodes are distributed as Poisson point process, and the set of N_{active} nodes is a CDS when the following properties are satisfied. First, the active nodes are connected by the transmission range R which is also large enough so that all nodes are covered by one of Voronoi cells in Voronoi tessellation [19] that is organized with the N_{active} nodes. The *cover ratio* is a performance metric of the CDS problem in random geometric graphs. We define

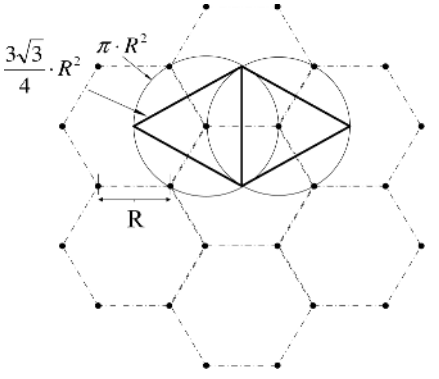


Fig. 1. Triangular tessellation

cover ratio α as the ratio of the average Voronoi cell size (A/N_{active}) to the area size of the neighbor nodes (πR^2) [5]. In the above case, α is on the order of $\frac{1}{\log(N_{active})}$ [6]. The problem of finding a MCDS of a graph is polynomially equivalent to finding a *maximum leaf spanning tree* of the graph [20]. A spanning tree with a set of leaves of large cardinality is obtained when the non-leaf nodes are uniformly distributed with high α value. The optimal α in an ideal case is a constant value. In Fig. 1, the ideal case is shown and the optimal α is $\frac{3\sqrt{3}}{4\pi}$. Heuristic algorithms for computing approximate solutions have various cover ratios with different properties [1,2,11,13]. It is beyond the scope of this paper to derive the cover ratio α for each algorithm.

We define the area A as a square area, where $A = L^2$. Generally, the average number of active nodes in a topology management scheme can be calculated as

$$N_{active} = \frac{L^2}{\alpha\pi R^2} \ , \quad if \quad 2 \cdot R_s \leq R \leq \sqrt{\frac{L^2}{\alpha\pi}} \ . \tag{1}$$

The minimum boundary of R is motivated by the geometric analysis that a sensing-covered network is always connected if it has the above property [22]. Also $N_{active} \geq 1$ limits the maximum boundary of R.

2.3 Average Idle Power in Topology Management

Most topology management protocols try to rotate the role of active node in fairness basis among all nodes of the network. So, all nodes consume their battery at the same rate and have the same lifetime in long time scale. We define β as the duty cycle of power save mode MAC protocols [8]. Then, the *average idle power* for one node can be derived from (1), as

$$\begin{aligned} \overline{P_{idle}} &= \frac{(N - N_{active})}{N} \cdot P_{sleep} + \left(\frac{N_{active}}{N} \cdot (\beta P_{idle} + (1-\beta) P_{sleep}) \right) \\ &= \frac{L^2 \beta (P_{idle} - P_{sleep})}{N \alpha \pi R^2} + P_{sleep} \ . \end{aligned} \tag{2}$$

2.4 Energy Consumption in Data Transmission

The energy to deliver a one bit message from one node to another within transmission range R is the transmission energy to transmit a bit at the sending node plus the energy needed to receive a bit at the receiving node. Then,

$$E_{1-hop} = E_{Rx-elec} + (E_{Tx-elec} + \varepsilon_{amp} \cdot R^{\gamma}) = 2 \cdot E_{elec} + \varepsilon_{amp} \cdot R^{\gamma} \ . \quad (3)$$

In the multi-hop communication model, it is assumed that considerable energy savings can be achieved by the use of multiple short range transmissions as opposed to one large hop transmission. When d is the distance from the source to the destination in communication, and R is the transmission range for the multi-hop communication, the required consumption energy can be expressed as

$$E = (2 \cdot E_{elec} + \varepsilon_{amp} \cdot R^{\gamma}) \left(\frac{d}{R} \right) \ . \quad (4)$$

To minimize the energy consumption, the optimal R_{opt} results in,

$$R_{opt} = \left(\frac{2 \cdot E_{elec}}{\varepsilon_{amp}} \right)^{\frac{1}{\gamma}} \ . \quad (5)$$

Accordingly, multi-hop short range communication is more energy efficient when R is greater than R_{opt}.

2.5 Total Data Rate

We define the *total data rate* λ of an entire sensor network as the sum of every individual data rate λ_i generated by node i. Then,

$$\lambda = \sum_{i=1}^{N} \lambda_i \ . \quad (6)$$

The average number of bits in one message and the frequency how often a message is generated by each sensor node is dependent on the traffic model for an application. First, we consider a static traffic model. This models for instance that the sensor nodes measure periodically various environment parameters (e.g., temperature, and pressure) and send the data back to the sink for further analysis. In this case we can assume that there is a steady flow of data from sensors to the sink, and as a result the total data rate is a constant. In contrast, the data rate generated by a sensor node may vary in event driven applications. Event generation may have a characterizable distribution in time and space. For simplicity, we assume the behavior of an event that is detected by node i, is a Poisson process with an average event rate given by μ_i. Node i generates an M bits data message whenever it senses an event, so λ_i is $M \cdot \mu_i$. Let $P_i(t)$ be the probability that at least one event occurs during time t at node i. Then,

$$P_i(t) = 1 - e^{-\mu_i \cdot t} \ . \quad (7)$$

That is, the inter-data generation time at each node is an exponential distribution with average event rate μ_i. In query based applications, the sink determines the data rates of each node directly in a real time manner. Therefore, we can easily compute the total data rate λ at each time interval t.

3 Optimal Transmission Range

3.1 Assumptions

1. N sensors are distributed according to a homogeneous spatial Poisson process in a square area A, where $A = L^2$. We may have two types of end-to-end communication models according to the location of the sink. In the first model, the sink is located at the center of the square area A. Then random variable D denotes the length of the segment from a sensor at random point (x, y) to the origin of the coordinate system. The expected value of D is equal to

$$E[D] = \int_A \sqrt{x^2 + y^2} \left(\frac{4}{L^2} \right) \mathrm{d}A = \frac{\sqrt{2} + \ln(1 + \sqrt{2})}{3} \cdot \frac{L}{2} \approx 0.3825 \cdot L \ . \quad (8)$$

   In the second model, the sink is not stationary but mobile, so it sends a query from all locations in the area and sensors reply to the sink. In this model, D denotes the length of the segment from a random point (x_1, y_1) to another (x_2, y_2). Then, by [21],

$$E[D] = \int_{A_1} \int_{A_2} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \left(\frac{1}{L^4} \right) \mathrm{d}A_2 \ \mathrm{d}A_1 \approx 0.521 \cdot L \ . \quad (9)$$

   The communication model is determined by an application. In this paper we consider only the second model.
2. The number of hops to communicate with two points at the distance D is equivalent to $\lceil \frac{D}{R} \rceil$, assuming that the routing strategy is intelligent. The boundary is given by

$$\frac{D}{R} \leq \lceil \frac{D}{R} \rceil < \frac{D}{R} + 1 \ . \quad (10)$$

   By equations (9) and (10), the average number of hops in two random point selections can be derived as

$$E\left[\lceil \frac{D}{R} \rceil \right] = \frac{D}{R} + 0.5 = \frac{0.521 \cdot L}{R} + 0.5 \ . \quad (11)$$

3. Data aggregation techniques that reduce the total volume of data to transmit are not considered here.
4. The wireless communication environment is contention free and error free, so that retransmission is not necessary. Energy consumptions derived by overhearing problem are not considered [8].

3.2 Computation of the Optimal Transmission Range

Based on these assumptions, the expected lifetime of the entire sensor network that uses the topology management mechanism can be expressed as

$$F_{lifetime}(R,\lambda) = \frac{E_{init} \cdot N}{\overline{P}(R,\lambda)} \ , \tag{12}$$

where E_{init} is the initial total battery energy for every node. The denominator of equation (12) represents the total power dissipation for all nodes during the lifetime and is given by

$$\begin{aligned}\overline{P}(R,\lambda) = &\left(\frac{L^2\beta(P_{idle}-P_{sleep})}{\alpha\pi R^2} + NP_{sleep}\right) \\ &+ \left(\left(\frac{0.521\cdot L}{R}+0.5\right)\left(2\cdot E_{elec}+\varepsilon_{amp}\cdot R^{\gamma}\right)\lambda\right) \ .\end{aligned} \tag{13}$$

The boundary of the transmission range R can be defined as by equation (1) and (5). Then,

$$R_{min} = \max\left[\sqrt{\frac{2\cdot E_{elec}}{\varepsilon_{amp}}} \ , \ 2\cdot R_s\right] \ , \ R_{max} = \min\left[R_{MAX} \ , \ \frac{L}{\sqrt{\alpha\pi}}\right] \ , \tag{14}$$

where R_{MAX} is the physically maximum transmission range of a sensor node. For calculating the optimal transmission range that makes the minimum power consumption in topology management, we must define the bound of λ. The conditions that make the optimal range in the boundary as equation (14) are given by

$$\frac{\partial}{\partial R}\overline{P}(R_{min},\lambda) < 0 \ , \ \frac{\partial}{\partial R}\overline{P}(R_{max},\lambda) > 0 \ . \tag{15}$$

Then, the boundary is

$$\omega_{min} < \lambda < \omega_{max} \ , \tag{16}$$

where ω_{min} and ω_{max} are

$$\begin{aligned}\omega_{min} = &\frac{-2L^2\beta(P_{idle}-P_{sleep})}{\alpha\pi R_{max}^3} + \left(-\frac{0.521\cdot L}{R_{max}^2}\right)\left(2\cdot E_{elec}+\varepsilon_{amp}\cdot R_{max}^{\gamma}\right)\lambda \\ &+ \left(\frac{0.521\cdot L}{R_{max}}+0.5\right)\left(\gamma\cdot\varepsilon_{amp}\cdot R_{max}^{\gamma-1}\right)\lambda\end{aligned} \tag{17}$$

and

$$\begin{aligned}\omega_{max} = &\frac{-2L^2\beta(P_{idle}-P_{sleep})}{\alpha\pi R_{min}^3} + \left(-\frac{0.521\cdot L}{R_{min}^2}\right)\left(2\cdot E_{elec}+\varepsilon_{amp}\cdot R_{min}^{\gamma}\right)\lambda \\ &+ \left(\frac{0.521\cdot L}{R_{min}}+0.5\right)\left(\gamma\cdot\varepsilon_{amp}\cdot R_{min}^{\gamma-1}\right)\lambda \ .\end{aligned} \tag{18}$$

Finally, the optimal value of the transmission range in topology management can be classified into three categories according to the data rate. Thus,

$$R_{opt} = \begin{cases} R_{max} & \text{if } \lambda < \omega_{min} \\ \left(\frac{\partial}{\partial R}\overline{P}\right)^{-1}(0) & \text{for given } \lambda_0 \text{ if } \omega_{min} \le \lambda < \omega_{max} \\ R_{min} & \text{if } \lambda > \omega_{max} \end{cases} . \quad (19)$$

The boundary values ω_{min} and ω_{max} of λ are important parameters that determine the optimal transmission range. In the case where λ is smaller than ω_{min}, maximum range is employed. As in existing protocols [12,13], selecting the active nodes within the smallest number is essential in such circumstances. If λ has a greater value than ω_{max}, the topology management may lose energy efficient advantage, so topology management is not a solution to save energy. However, the data rate is not always constant during the lifetime of the network and varies according to the characteristics of the data traffic generated. It is necessary to estimate data rate for a given interval regularly to compute the optimal transmission range and execute the topology management scheme with the optimal value.

4 Concluding Remarks

In this paper, we analyze power consumption in multi-hop communication environments and calculate the optimal transmission range for a given data rate which helps to extend the network lifetime. The result shows that the data rate within a certain range allow the system to save energy by adopting the multi-hop communication with the optimal transmission range. Topology management must be applied in a manner that the transmission range can be adjusted dynamically according to its optimal value for an interval if data rate is variable. We believe that the computation interval is one of implementation parameters.

References

1. A. Cerpa, D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies," in Proc. of IEEE INFOCOM 2002, Vol. 3, pp.1278 - 1287, June 2002.
2. S. Guha, S. Khuller, "Approximation algorithms for connected dominating sets," Algorithmica, 20, No. 4, pp. 374-87, April 1998, Springer-Verlag.
3. S. Singh, M. Woo, C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp.181 - 190, 1998.
4. B. Karp, H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in Proceedings of the 6th annual international conference on Mobile computing and networking, pp.243 - 254, 2000.
5. S. G. Foss and S. A. Zuyev, "On a Voronoi Aggregative Process Related to a Bivariate Poisson Process," Advances in Applied Probability, Vol. 28, no. 4, pp. 965-981, 1996.

6. S. Shakkottai, R. Srikant, N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," in Proc. of IEEE INFOCOM 2003, pp.1073 - 1083, March-April 2003.
7. Y. C. Tseng, C.S. Hsu, T.Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in Proc. IEEE INFOCOM, pp.200-209, June 2002.
8. W. Ye, J. Heidemann, D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Transaction on Networking, Vol. 12, No. 3, pp. 493-506, June 2004.
9. C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, "Topology Management for Sensor Networks: Exploiting Latency and Density," in Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing, pp.135-145, June 2002.
10. T. van Dam, K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in the First ACM Conference on Embedded Networked Sensor Systems(Sensys03), pp.171-180, November 2003.
11. L. Bao, J.J. Garcia-Luna-Aceves, "Topology Management in Ad Hoc Networks," in Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing, pp.129-140, 2003.
12. Y. Xu, J. Heidemann, D. Estrin, "Geography-informed energy conservation for Ad Hoc routing," Proceedings of the 7th annual international conference on Mobile computing and networking, pp.70-84, July 2001.
13. B. Chen, K. Jamieson, K. Balakrishnan, R. Morris, "SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless networks," Wireless Networks, Vol. 8, Issue 5, pp.481-494, 2002.
14. A.D. Amis, R. Prakash, T. Vuong, D.T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," in Proc. IEEE INFOCOM, pp.32-41, March 2000.
15. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, 38(4), pp.393-422, 2002.
16. C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," in Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing, pp.80 - 91, June 2002.
17. V. Raghunathan, C. Schurgers, S. Park, M. Srivastava, "Energy-Aware Wireless Microsensor Networks," IEEE Signal Processing Magazine, Vol. 19, Issue 2, pp.40-50, March 2002.
18. W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in Proc. Hawaii Int. Conf. Systems Sciences, pp.3005-3014, January 2000.
19. A. Okabe, B. Boots, K. Sugihara, "Spatial Tessellation Concepts and Applications of Voronoi Diagrams," New York, NY: John Wiley, 1992.
20. Y. Caro, D. B. West, R. Yuster, "Connected domination and spanning trees with many leaves," SIAM J. Discrete Math, Vol. 13, No. 2, pp.202-211, April 2000.
21. J. Borwein, D. Bailey, "Mathematics by Experiment: Plausible Reasoning in the 21st Century," Natick, MA: A.K. Peters, 2003.
22. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C.D. Gill, "Integrated coverage nd connectivity configuration in wireless sensor networks," in the First ACM Conference on Embedded Networked Sensor Systems(Sensys03), pp.28-39, November 2003.
23. D. Ganesan, D. Estrin, and J. Heidemann, "DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?," in Proc. of the ACM Workshop on Hot Topics in Networks, pp.143-148, Princeton, USA, October 2002.

Service Discovery in Mobile Ad Hoc Networks*

Hua-Wen Tsai[1], Tzung-Shi Chen[2], and Chih-Ping Chu[1]

[1] Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan 701, Taiwan
Tel.: 886-6-2757575 ext. 62527, Fax: 886-6-2747076
chucp@csie.ncku.edu.tw
[2] Department of Information and Learning Technology
National University of Tainan, Tainan 700, Taiwan
chents@mail.nutn.edu.tw

Abstract. Service Discovery is expected to become a crucial application of mobile ad hoc networks (MANETs). A number of existing service discovery protocols is mainly aimed at infrastructure-based and/or 1-hop ad hoc wireless networks. In this paper, we propose a novel mechanism using hierarchical grid architectures for service discovery in MANET. The geographic area of MANET is partitioned into 2D logical hierarchy grid. In each grid, we select a directory node to cache the available service descriptions that are registered by the providers for replying the inquiry of requestors. For enhancing the efficiency of registration and discovery, we propose service registration and discovery transmission patterns. The shared service description is registered in the grids of registration pattern. When a seeker needs a service, it obtains the descriptions along the discovery pattern. Finally, the performance shows that our protocol can work efficiently in any density and different scales of networks.

1 Introduction

Recently, the research of application has focused on service discovery. Service/location discovery [5][6] is one of applications in MANET to achieve standalone and self-configurable communication networks. The wide application range of service are used in providing host's location, printing documents, processing database queries, lending of computational power, delivering documents, using technical equipment [5], and so forth. A service discovery protocol lets the service providers to advertise their services, and allows users to automatically discover the services. Another application of service discovery is location discovery that offers the information of node's geographical position.

The major structural difference between the existing service discovery architectures is whether the architecture utilizes a central directory or not. Sun's Jini [1] utilizes a central directory to register and to search the available service. The central directory keeps the service descriptions that are registered by the providers. A requestor queries the central directory while the requestor needs a service. The central directory replies the service descriptions to the requestor. This method can reduce the

* This work is supported by National Science Council under the grant NSC-94-2213-E-024 - 002, Taiwan.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 186 – 195, 2006.

cost of search efficiently. But it is suitable for wireless infrastructure-based networks. Another is decentralized resource discovery protocols that are more suitable to MANET, e.g. SLP [3]. The known decentralized resource discovery protocols are pull-based and push-based discovery. The pull-based discovery is that a requestor broadcast a query packet to search a service provider. The push-based discovery is that a provider broadcast a registration packet to register its service. But the broadcast strategy incurs more overhead in traffic.

Recently, a geographic approach is proposed to register service descriptions. Geography-based Content Location Protocol (GCLP) [6] uses the geographic location information to reduce the overhead in traffic. When a provider wants to share its service, it designates four farthest neighbors to advertise the registration packet. These nodes are located at the provider's east, west, south and north respectively. And then the registration packet is forwarded along the four geographic trajectories. When a node receives the registration packet, it has to keep the service description. If this node is a designated node at the trajectory, it continues to advertise the packet along the trajectory. When a requestor wants to search a service, it sends a query packet along the geographic trajectories that are similar to that of registration process. Eventually, one of query trajectories is intersected the registration trajectories. This method reduces the cost of discovery and registration efficiently. And it is also suitable for the scalability network. However, there are two major drawbacks in this solution. First, it only is suitable for the high density MANET. When the density of whole or part network is sparse, this method can not guarantee to implement very well. Another drawback is that a large number of nodes must cache the service descriptions. In this method, all nodes close to the geographic trajectory have to cache the service descriptions. Hence, it incurs more resource consumption.

The motivation of this paper is to improve the shortcomings of the existing method. Here, we combine the advantages of directory and geographic information. Though the node must equip the device that supports geographical location (e.g., GPS or other location devices), this technology will be popularized in the near future. We can achieve the goal of saving resource consumption through geographical information. We propose a service discovery protocol in Grid that is proposed in [4] and we name SGrid. We use a geographic area of MANET that is partitioned into 2D logical hierarchical grids. In each grid, a head node is selected and it also acts a directory that caches the service description. We propose registration and discovery patterns for service registering and discovering, respectively. SGrid can reduce the resource consumption and is suitable for the large network scale. In addition, SGrid can also discover the service at the low density network efficiently.

The rest of this paper is organized as follows. Section 2 presents the network model. Our protocol is developed in Section 3. Section 4 shows the experimental results. The conclusions from this work are presented in Section 5.

2 Network Model and Notations

In this paper, we assume that each node acquires its current location from GPS or other positioning devices. Our protocol is designed for a designated zone, e.g. colloquium, zoo, botanical garden, disaster area, downtown, great exhibition and metropolis. The service is provided in a designated zone. The detail description of grid model

is in [4]. The geographic area of MANET is partitioned into 2D logical grids. We assume that each grid is a square and its size is $d \times d$. The area of network is constituted by $N \times N$ grids. $grid_{(x, y)}$ is the grid-coordinates. A node can transfer the physical-coordinates to the grid-coordinates via a predefined mapping while it acquires the physical location.

The *provider* represents that a node has a service and it wants to provide for others. The *requestor* represents that a node needs a service. The *seeker* represents that a requestor does not have the service and it needs to get the service from MANET. Three kinds of node are *head*, *gateway* and *normal*. A node will be selected as *head* in a grid. The responsibility of *head* includes: (i) acting the directory to record the service descriptions within the grid, (ii) forwarding the service discovery/service registration packets to its neighboring grids, and (iii) maintaining the routes between the grids. The *gateway* node is only responsible for (ii). The *normal* node is not responsible for these jobs unless it is a service provider/seeker. A service provider is responsible for (ii) and (iii). A service seeker is responsible for (iii).

Let R be the transmission range and d be the side length of grid. For mapping the physical-coordinates to grid-coordinates, we must predefine the relationship of d and R. We assume that the transmission range of directory must be able to cover the whole grid. So, the minimum d is $d=R/\sqrt{2}$. In this relationship of d and R, a directory can communicate with any node in its grid. But the directory maybe cannot communicate with that of neighboring grids directly. The communication must be via a relay node (gateway).

For registering and discovering the service, we maintain three tables that are neighbor, service and route table in a node. Neighbor table records the neighbor node information that includes *xy*-coordinates, neighbor's node state and the node state of its neighbor grids. Service table records the service descriptions that are registered from the providers. The routing information is recorded in the route table. We define some notations that are used throughout this paper. $grid_{(x, y)}$ is the grid coordinate, while x is grid x-coordinate and y is grid y-coordinate. $G_n.x$ and $G_n.y$ are the grid-coordinates of node n in $grid_{(x, y)}$. $G_n.xlv$ and $G_n.ylv$ are the grid level of $G_n.x$ and $G_n.y$ (to be discussed in Sect. 3.1). $G_n.lv$ is the grid level of $grid_{(x, y)}$. $Reg_n.td$ is the transmitting direction of register packet that is issued by provider n. The values of $Reg_n.td$, 0 and 1, represent that the register packet is transmitted toward the horizontal and vertical direction respectively.

3 Service Discovery Protocol

In this section, we describe our proposed protocol SGrid that includes the hierarchical grid architecture, directory selection, service registration, and service discovery phase.

3.1 Hierarchical Grid Architecture

In this subsection, we propose hierarchical grid architecture for registering and discovering service. Two terms, *grid level* (L) and *zone size* (Z), are used to form hierarchical grid architecture. The level of grid is from 0 to L. The grid area of level 0 is also named *zone* and the side length of zone is Z grids. The network is $N \times N$ grids and N is equal to $(1+Z) \times 2^L$-1.

Fig. 1 illustrates an example for different zone sizes, while L is 2 and Z is 1 and 2 at **Fig. 1** (a) and **Fig. 1** (b), respectively. The proposed hierarchical grid architecture can easily adapt to any scale of network. MANET is used in a designated area for a special purpose. So the scale of MANET can be predefined and every node knows this information. Here, we assume that that each node knows the values of L, Z and the physical-coordinates of $grid_{(0, 0)}$. When a node acquires its or its neighbor's physical-coordinates, it can obtain the grid-coordinates via the predefined mapping. A node can also compute the grid level while it knows the grid-coordinates. Each node utilizes Algorithm 1 that is shown in **Fig. 2** to obtain the grid level. We assume $L = 2$ and $Z = 2$ shown in **Fig. 1** (b). When node n knows that it is in $grid_{(6, 4)}$, $G_n.xlv = 2$, $G_n.ylv = 0$ and $G_n.lv = 2$ are obtained by Algorithm 1.

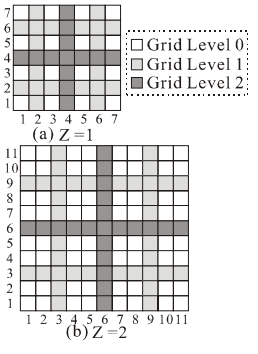


Fig. 1. An example for different zone size (Z) of hierarchical grid architecture

```
Algorithm 1: Computing-grid-level ( )
{Assuming Gn.x and Gn.y are gotten from
 grid(x, y), Z is the parameters of grid
 architecture}
1  lv←1, run←true, Gn.xlv←0, Gn.ylv←0
2  while run=true
3    do run←false
4       if Gn.x mod (1+k)*2^lv - 1=0
5         then run←true, Gn.xlv←lv
6       If Gn.y mod (1+Z)*2^lv - 1=0
7         then run←true, Gn.ylv←lv
8       If run=true
9         then lv←lv+1
10 if Gn.xlv>=Gn.ylv
11   then Gn.lv←Gn.xlv else Gn.lv←Gn.ylv
```

Fig. 2. Algorithm 1 - Computing node n's grid level

3.2 Directory Selection Phase

In this subsection, we discuss the directory selection. We select a head as directory to keep the shared service descriptions in each grid. First, each node needs to know the information of its neighbors. For keeping the latest information of its neighbors, each node broadcasts Hello message to its neighbors periodically. When a node receives a Hello message from its neighbor, it records the information of Hello message into its neighbor table. The Hello message includes the node location and its node state. The format of Hello message is ***Hello***(*id*, *x*, *y*, *state*, <*NSL*>), while *id* is the unique ID of broadcasting node; *x*, *y* are the physical-coordinates; *state* is node state that are three states: *Head* (*H*), *Normal* (*N*) and *NULL*; *NSL* is a node state list that includes eight node states in *id*'s neighbor grids. The information of *NSL* is used to detect whether a gap grid exists or not. *Gap grid* means that there is no node in a grid. We can not forward any packets to the gap grid. In addition, we cannot guarantee that a head can communicate with another head in its neighbor grid directly in $d=R/\sqrt{2}$. If the communication distance is greater than 1 hop, the communication has to pass a relay node (gateway). The gateway information is also obtained from the *NSL* of its neighbors'

Hello messages. In other words, when a head detects that it can not directly communicate with another head, it select a gateway from its neighbor table.

When a node *n* broadcast its Hello message, the eight node states of *NSL* are got from *n*'s neighbor table and they are arranged in *NSL* according to the positions (directions) of grid as <northwest, north, northeast, west, east, southwest, south, and southeast>. If there are more than two nodes in a neighbor grid, the priority of *head* is high than that of *normal*. If no node is in a neighbor grid, the node state in this grid is set *NULL*. An example shown in **Fig. 3**(a), node *A* broadcast ***Hello***(*A*, *x*, *y*, *H*, *NSL*<*NULL*, *H*, *N*, *H*, *N*, *NULL*, *H*, *N*>) to its neighbors. When node *A* wants to communicate with nodes *B* and *C*, it has to forward the packets via nodes *D* and *E*, respectively.

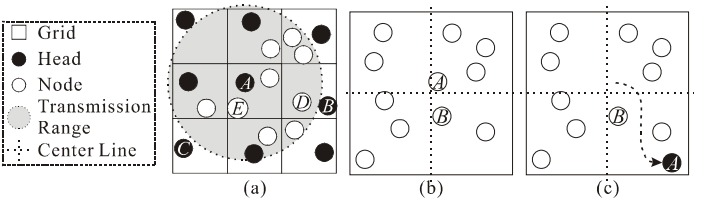


Fig. 3. (a) An example of MANET with grid, (b) and (c) an example for head selection

The nodes within the same grid can receive Hello messages each other while $d=R/\sqrt{2}$. So a node knows the information of all nodes that are in the same grid. The head selection is described in below. When no head was in a grid, a node that is in the center of grid selects itself to act head. And then it utilizes next Hello message to inform its neighbors "I am the head node". In **Fig. 3** (b), node *A* is the node that is the closest the center of grid and there is no head existed. So, node *A* selects itself to act head. Each node may be a mobility node in MANET. Other node has to act head while the original head has gone away the grid. In order to reduce the overhead of changing head, a new head is selected if the original head had left the grid. In addition, when the original head had no enough resources to support acting a head, a new head is also selected. The original head forwards all cached service descriptions to the new head while it has left the grid or it does not act a head. This procedure can guarantee that the cached service descriptions will not be lost in the grid. In **Fig. 3** (c), we assume that node *A* has left the center of grid and it is still in the same grid, so node *B* is still a *normal* node. If node *A* has left the grid, node *B* is the new head.

3.3 Service Registration Phase

In this subsection, we discuss the provider how to register its services into network. In the following discussion, the communication is focused on the head layer. We do not expatiate the routing between two grids. In order to not rely on broadcasting method, we propose a predefined transmission pattern, named *registration pattern*, to register services. The service descriptions are registered in the directories of *backbone grids*. The *backbone grids* are on the *registration pattern* and the level of backbone grids is greater than level 0. Before we discuss the registration pattern, we predefine the concept of *region*. Grid level *lv* divides the area into 4 regions, i.e. regions *A*, *B*, *C*, and *D*. The location of region *A*, *B*, *C*, and *D* are at the northwest, northeast, southeast and

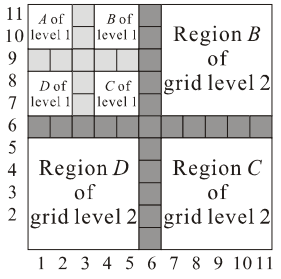


Fig. 4. An example for the regions of grid level *lv*

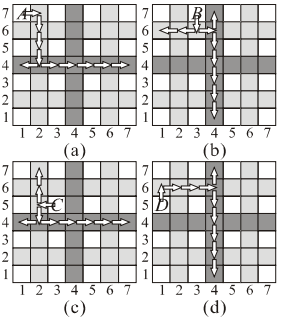


Fig. 5. The registration pattern of each region

southwest of level *lv*, respectively. Grid level 0 is located at the region *A*, *B*, *C* or *D* of grid level 1. An example for the region of grid *lv* is shown in **Fig. 4**, while the value of *L* and *Z* are both 2.

Next, we discuss the registration pattern in detail. When a provider *n* wants to share its services into MANET, *n* forwards a registration packet along the registration pattern to register service. The registration packet includes the service descriptions, the transmission direction ($Reg_n.td$), and the designated nodes that must receive this packet. We discuss the registration pattern at grid level 0 and other level separately.

At grid level 0, the registering direction is in accordance with the region of grid. Two basic transmission directions are vertical and horizontal. The registration pattern at grid level 0 has four types that are shown in **Fig. 5**. A provider *n*, which is in grid level 0 and at the region *A* or *C* of grid level 1, transmits the service descriptions toward horizontal direction ($Reg_n.td = 0$). The provider *n*, which is in grid level 0 and at region *B* or *D* of grid level 1, transmits the service descriptions toward vertical direction ($Reg_n.td = 1$). The transmission direction of registration packet is kept constant at the same level. When a head on the registration pattern receives the registration packet, it caches the service descriptions in its service table and continues to forward this packet. Others nodes discards the received packet except the gateway nodes. When the packet is transmitted from grid level *lv* to grid level *lv*+1, the head of grid level *lv*+1 varies the transmission direction. This process is repeated until the service is registered to the maximal grid level. In **Fig. 5**(a), $grid_{(1, 7)}$ is in grid level 0 and at the region *A* of grid level 1, so the first transmission direction of $grid_{(1, 7)}$ is the horizontal. The designated head is the head in $grid_{(2, 7)}$. When the head in $grid_{(2, 7)}$ (i.e. grid level 1) receives the service registration packet, it varies the transmission direction from the horizontal to vertical. When the registration packet is transmitted to $grid_{(2, 4)}$ (i.e. grid level 2), the transmission direction is varied from vertical to horizontal direction.

At other grid level (except grid level 0), the first registering direction is along the direction of same level. For example, a provider is in $grid_{(2, 1)}$ (i.e. grid level 1), so the first transmission direction is the vertical. When a node in $grid_{(2, 4)}$ (i.e. grid level 2) receives the registration packet, it varies the transmission direction from the vertical to horizontal. A grid is an intersection grid while $grid_{(x, y)}.xlv$ is equal to $grid_{(x, y)}.ylv$ (e.g. $grid_{(4, 4)}$ or $grid_{(2, 6)}$). When a provider is in an intersection grid, the transmission

direction of provider can be selected the horizontal or vertical. We define that the transmission direction is vertical while a provider is in an intersection grid.

Next, we discuss the multiple same services provided in the network. A provider updates its service description periodically and the regular update period is $T_{regular}$. A service description is kept in the service table of directory a predefined time T_{keep}. We assume that the current time is $T_{current}$, the cached time of service description is T_{cached}, the update time is $T_{current}$-T_{cached} and the threshold of update time is T_{update}. The service description is removed while $T_{current}$-$T_{cached} \geq T_{keep}$. When a directory receives a registration packet from a provider at $T_{current}$, the directory checks whether the registration pattern has registered or not. If yes, the directory checks the later cached time $Later(T_{cached})$ of the same service description whether $T_{current}$-$Later(T_{cached}) \geq T_{update}$ or not. If yes, the directory updates the service description and it continues to forward the packet. If $T_{current}$-$Later(T_{cached}) < T_{update}$, the directory discards this packet. If the registration pattern has not registered, it caches the service description and it continues to forward the packet. In other words, the same service description does not need to update while the update time is lower than T_{update}. This way can reduce the overhead of register. We assume $T_{keep} > T_{regular} > T_{update}$.

3.4 Service Discovery Phase

When a requestor can not find a special service in its service table, it has to execute discovery process. We also name this requestor as *seeker* that issues a discovery packet to query the directories. The seeker designates the heads in its neighbor grids to receive and to convey the discovery packet along the *discovery pattern*. The directories in maximum grid level have the service descriptions, so the direction of discovery is towards the maximum grid level. When a head receives this query, the head checks whether it has the special service description in its service table or not. If yes, it replies the information that includes the provider's address and position to the seeker. If no, it checks whether it is the designated node. If it is the designated node, it forwards the discovery packet along the discovery pattern to the maximum grid level. If no, it just discards the packet. We assume that each node knows the size of network, so it can compute and know the position of the maximum grid level. When the head in the maximum grid level receives the discovery packet, the heads will stop forwarding the discovery packet. The range of discovery is limited in a region of the maximum grid level. An example for service discovery is shown in **Fig. 6**. There are two providers and two seekers in this network. S_1 can get the provider information at $grid_{(4, 2)}$ and $grid_{(6, 4)}$. S_2 can get the provider information at $grid_{(4, 7)}$ and $grid_{(5, 4)}$.

3.5 Improved Registration Phase

Our service discovery algorithm SGrid has better efficiency while it is executed in the high node density network. SGrid employs the head of grid to act directory for reducing the number of directory nodes. This is because that only one node needs to cache the registered service in a grid. When SGrid is executed in a sparse network, many grids maybe have no nodes due to hollow regions. *Gap grid* means that no node exists in a grid. The gap grid will stop the transmission for registration and discovery. **Fig. 7** shows an example for gap grids and three gap grids are in $grid_{(4,2)}$, $grid_{(3,3)}$ and

$grid_{(4,3)}$. Inasmuch as $grid_{(4,2)}$ and $grid_{(4,3)}$ are in the registration pattern, they influence the transmission of registering. In order to solve this problem, we improve the registration phase and propose *alternate grid*. We use *alternate grids* to prevent the hollow region problem. When a node receives all Hello messages from its neighbors, the node utilizes the *NSL* of Hello messages to check whether its neighbor grid is a gap grid or not. If a directory detects that one of its neighbor grids is a gap grid, the directory set its grid as *alternate grid*. The directory of alternate grid has to cache the service descriptions and forward the registration packet to its neighbor grids while it has received the packet. In other words, a set of alternate grids takes the place of a gap grid. All neighbor grids of gap grids are alternate grids shown in **Fig. 7**. Provider *P* registers its service along the registration pattern. When the directory of $grid_{(4,4)}$ can not forward the registration packet to $grid_{(4,1)}$ via $grid_{(4,3)}$, it utilizes its neighbor alternate grids to forward this packet to the directory of $grid_{(4,1)}$. If we do not utilize the alternate grid to forward the packets, the registration pattern will be broken. Seeker *S* can find the service description from the directories in alternate grids. In improved registration phase, we patch the gap grids using a set of alternate grids. This way increases the success ratio for discovery effectively, though it will incur the number of directories. Therefore, the seeker can seek the service at the low density network via the alternate grids.

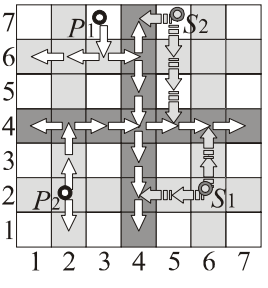


Fig. 6. An example for service discovery

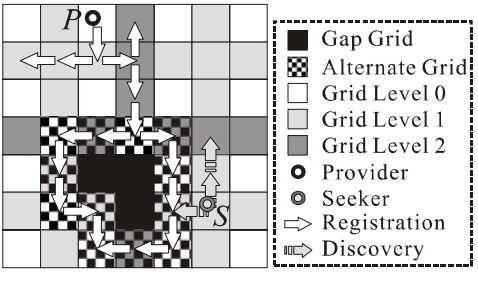


Fig. 7. An example for improved SGrid

4 Experimental Results

This section presents the experimental results of our service discovery algorithm SGrid compared with GCLP's [6] using ns2 [2] simulator. We use four metrics for work: number of service directories, success ratio, discovery overhead, and average delay time. Number of service directories is used for estimating that how many nodes must to store one type of service. The success ratio is equal that the number of found the service requestors divided by total requestors. The discovery overhead is the packet quantity that is used in the protocol including Hello messages. The average delay represents the necessary time of found service. When a requestor can find the needed service in its service table, the delay time is set 0.

Each radio's range is approximately a disc with a 250 meter radius. The simulations use 2 Megabit per second radios. Each simulation runs for 300 simulated seconds. The nodes are uniformly random placed in a square area. We vary the number of mobile

nodes from 100 to 500 nodes. One side of the square area is of 3344 m. The side of grid (*d*) is 176 m. The values of *L* and Z are 2 and 4, respectively. The movement model of node is used a "random waypoint" model. The maximum speed is from 10 to 50 m/s and the pause time is set zero. In our scenario, we simulate 10 service types and a service type has 3 providers. Any node may be a requestor except the providers. Each requestor selects a random service type at a random time. And then a requestor will not inquire the same service twice in a simulation run. SGrid-10 and SGrid-50 represent that the maximum speed of mobile nodes are 10 and 50 m/s, respectively.

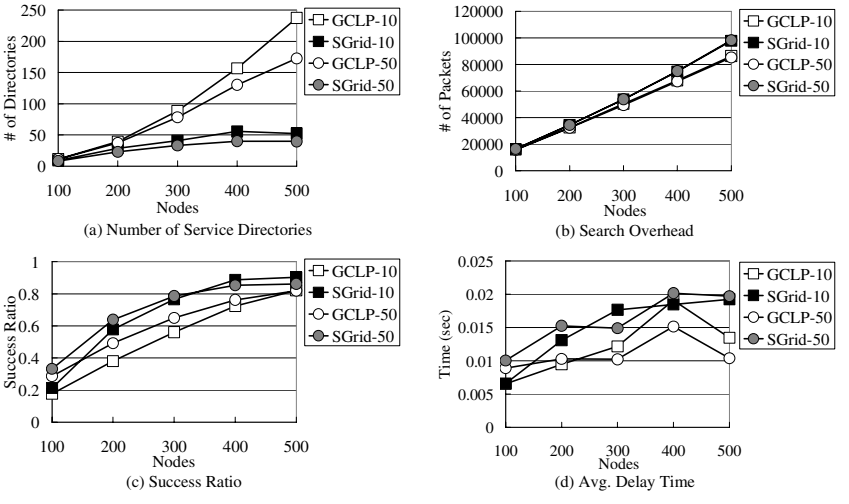


Fig. 8. The performance of simulation

Fig. 8(a) shows the number of directories. When the number of nodes is increased, the number of directories in GCLP will be followed to increase. This is because that all nodes close to the geographic trajectory are directories for GCLP. In other words, when the density of the network becomes higher, GCLP has more directories. SGrid does not need more directories to cache the service descriptions while the density of the network becomes higher. This is because that we use the grid structure and the head of backbone grids to act the directories. This way can steady the quantity of directories effectively, e.g. GCLP-10 needs 237 nodes to cache a service description, but SGrid-10 only needs 52 nodes. No matter how many nodes are in a grid, SGrid only needs a directory to cache the service descriptions. Therefore, the quantity of directories is increased slightly when the number of nodes increases. The performance of discovery overhead is illustrated in **Fig. 8**(b). SGrid needs a little more discovery overheads than GCLP. This is because that SGrid uses fewer directories to cache the service descriptions. When a requestor can not find the need service information in its service table, it must issue a packet to discover the service. The seeker/requestor ratio of SGrid is higher than GLCP's due to SGrid utilizes the fewer directories at the higher density network. E.g. in 500 nodes and 10 m/s, GCLP and SGrid have 51.81% and 69.40% requestors to seek the services, respectively. The performance of success

ratio is shown in **Fig. 8**(c). The success ratio of SGrid is better than GCLP's, e.g. in 400 nodes, 10 m/s, the success ratio of GCLP and SGrid are 72.42% and 88.67%, respectively. The cause is that the hollow regions exist in the network though the network connects each other. If the geographic trajectory in GCLP meets a hollow region, the transmission will break down and it cannot be forwarded along the predefined direction. SGisd can overcome this shortcoming in the low node density, because the alternate grids are used to prevent the hollow region problem. We compare the performance of delay time in **Fig. 8**(d). The delay time is shown the transmission time for discovery. The average delay time of GCLP and our algorithm's are similar.

From the experimental results, we can observe that SGrid has better efficiency than GCLP in any density network. GCLP only is suitable for the high density network. In the middle or low density network, the efficiency of GCLP is not good. This is because that the hollow regions have existed in network though the networks have connection. The alternate grid of SGrid is used for solving the hollow region problem. We use the different registration patterns for each region at the same level on the Grid. The registering load is distributed to different registration patterns for reducing the overhead of backbone grids.

5 Conclusions

We propose a service discovery protocol for MANET. Our algorithm is based on a hierarchical grid structure. When a provider wants to register its service into MANET, it transmits the registration packet along a predefined registration pattern to the backbone grids. In addition, we only need to discover a quarter of network area. So we can reduce the cost of register and discovery. To prevent a gap grid at the registration pattern, we use a set of alternate grids to patch a gap grid. In simulation analysis, SGrid got better success ratio and the number of directories than GCLP. Summarizing the result of experiments, our algorithm can implement efficiently in any density and different scale of networks.

References

1. K. Edwards and T. Rodden, *Jini Example by Example*, Prentice Hall PTR, Jun, 2001.
2. K. Fall and K. Varadhan, ns notes and documentation. Technical report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Nov. 1997.
3. E. Guttman, J. Veizades, C. Perkins, and M. Day, REC 2608: Service Location Protocol, version 2, Jun. 1999.
4. W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," in *Telecommunication Systems*, Vol. 18, No. 1, pp. 37-60, 2001.
5. F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," in *Proceedings of the* 3rd *IEEE International Conference on Pervasive Computing and Communication* (*PerCom* 2005), Sheraton-Kauai Resort, Kauai, Hawaii, USA, pp. 235-244, Mar. 2005.
6. J. B. Tchakarov and N. H. Vaidya "Efficient Content Location in Wireless Ad Hoc Networks," in *Proceedings of the* 2004 *IEEE International Conference on Mobile Data Management*, Berkeley, California, USA, pp. 74-85, Jan. 2004.

Service Oriented Networks – Dynamic Distributed QoS Routing Framework

See-Hwan Yoo and Chuck Yoo

Department of Computer Science and Engineering, Korea University
{shyoo, hxy}@os.korea.ac.kr

Abstract. Routing algorithms have been using a single metric such as hop count in route calculation. The selected route is optimal only in the context of the metric used. In order to support diverse service requirements from users, there is a growing need where multiple metrics need to be taken into account in routing, especially in ad-hoc routing. This paper addresses ad-hoc routing with multiple metrics. We present a new framework called service oriented network in which users can specify routing metrics and the metrics are used in route calculation. Our approach is implemented in Linux kernel and the compared with AODV. Simulation study shows that using our framework, we can improve the network lifetime by 200%.

1 Introduction

This paper is to introduce the notion of 'service' to ad-hoc routing. A service is characterized by its requirements. For example, a user may want a secure communication service even though the delay is endurably long. Another user might want low delay jitters to watch a movie on his mobile devices. Obviously, user requirements are very diverse, depending on people's preference, and also requirements may change dynamically. Some of service characteristics such as delay are network-related, and to meet the service requirements needs cooperation of the underlining routing algorithm.

However, current ad-hoc routing algorithms do not consider service requirements. For example, Ad-hoc On-Demand Distance Vector [1] (AODV) is one of the most famous ad-hoc routing protocols. It makes use of the Distance Vector algorithm, and it achieves endurable response time with low routing overhead. In other words, AODV considers only response time in route calculation. Other ad-hoc routing algorithms are similar in that a single criterion is used in route calculation.

To the best of our knowledge, this paper is one of the first attempts to incorporate the service notion into ad-hoc routing protocol. Specifically, we attempt to make ad-hoc routing flexible enough to handle user requirements so that ad-hoc routing can adapt dynamically to different characteristics of various services. This paper proposes an ad-hoc routing protocol construction framework that takes multiple criteria in route calculation to meet various

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 196–206, 2006.

service requirements including QoS parameters. Furthermore, our framework provides a programmable interface using simple script language to express service requirements.

This paper is organized as follows. Background and related work is described in Section 2. Section 3 highlights the core concept of the service oriented networks, the framework which constructs a service-specific network, and Section 4 presents the route construction and maintenance mechanisms in the service oriented network framework. Finally, we conclude this paper with the results in Section 5.

2 Background and Related Work

We briefly describe how ad-hoc network works. In an ad-hoc network, nodes configure themselves to communicate with their neighboring nodes. Also, any node can choose to join or leave the network, which is different from existing networks. Each node should provide its own resources to guarantee the connectivity with other nodes. Therefore, every node works as an end terminal as well as an intermediate router. In a given route, nodes included in the route ('members') should process packets. That is, all the members should involve in the route construction process, and they should maintain network information in order to provide the connectivity among themselves.

Routing protocol in an ad-hoc network provides connectivity among the nodes without preexisting infrastructure. Many routing protocols have been proposed for the ad-hoc network, and AODV and DSR are the most popular routing protocols. AODV and DSR [2] are called on-demand routing protocols because they do not keep the past network information such as topology. AODV construct a route to a designated destination based on the node's responsiveness. It selects the shortest delay path in the probing time. AODV does not maintain the route to arbitrary node in advance; instead, it probes the whole network to find a route when a new session starts. Every node in AODV network keeps routing table and it contains the next hop to destination in distributed manner. On the other hands, in DSR, only the source keeps explicit node list on the path. To keep the recent neighbor's information, AODV optionally implements a scheme that keeps sending hello messages periodically.

One of the important metric in ad-hoc routing protocol is network lifetime. Network lifetime is highly dependent on the remaining energy of the member nodes in the network. Network data transmission is one of the dominant energy consuming parts in a system, and many strategies for energy efficient management are proposed. Specifically, for longer network lifetime, several routing protocols are proposed [3]. One of them is based on total transmission power. In the scheme, route is selected based on the battery consumption of all the nodes on the path when a packet is sent. Namely, the total energy consumption for a packet on each path is calculated. If we assume that every node's energy consumption for a packet is the same, then the minimum hop route is naturally selected. However, this scheme does not consider the network lifetime. So the

lowest battery node should not use its energy on forwarding. Therefore, Toh[4] suggested a routing scheme that can avoid the lowest residual battery node. In CMMBCR[4], among all the nodes on each path, the minimum battery capacity remaining node is selected. Then, a path that has the maximum lowest energy capacity is selected as the best route. More distributed power control algorithms are proposed and power efficient protocol related work is summarized in [5].

Security issue in ad-hoc network is serious because all the nodes in the network take part in network management. For example, every node maintains a routing table and exchanges routing messages. Moreover, temporal communication failures occur often in ad-hoc networks, and it is very difficult to distinguish them from the network attack. Several studies reveal the exploitation for the routing protocol, and propose defensive schemes against security risks for ad-hoc routing protocols [6], [7].

QoS routing protocol is proposed for satisfying the QoS-related requirements [8], [9]. For realtime applications, network delay for the communication should be pre-determined. However, finding a route with delay-bound is difficult in two folds. At first, the network dynamics influences the network delay, and secondly it is proven that finding the lowest delay route to arbitrary destination is an NP-complete problem. Therefore, heuristics and approximation algorithms have been developed. Current QoS routing can be categorized as follows: 1) source routing, 2) distributed routing, 3) hierarchical routing. Various routing objectives lead different routing strategies such as bandwidth-based route selection, delay-bound route selection, etc. Most of the QoS routing protocol does not consider dynamics in the network environment and requirement change, and they solve only routing problem in static environment. Wide meaning of QoS covers various criteria such as user response or security level, which are service-specific criteria. Current QoS routing study focus on network issues such as delay bound, available bandwidth, or jitters, and the algorithms for approximation. This work enlarges the area of QoS routing in more general meaning of QoS with flexible framework.

Active network is a research area for adapting network systems to dynamically changing environment. Several approaches are proposed for supporting flexible and programmable network construction [10], [11]. Programmable network provides flexibility in network management. Because the network management is a complex task, network administrator wants it to be done automatically. However, some research work shows a possibility of active network with real experiments using Tunneling protocol [12].

3 Concept of Service Oriented Networks

The core concept of the service oriented network is presented in figure 1. The service oriented network is to support services with different network requirements so that the services should be provided in the timely manner and can be changed easily or migrated to other environment. Different services such as security or multimedia streaming service have their own requirements. In service oriented network, the middle layer, called collaboration layer, gathers the requirements

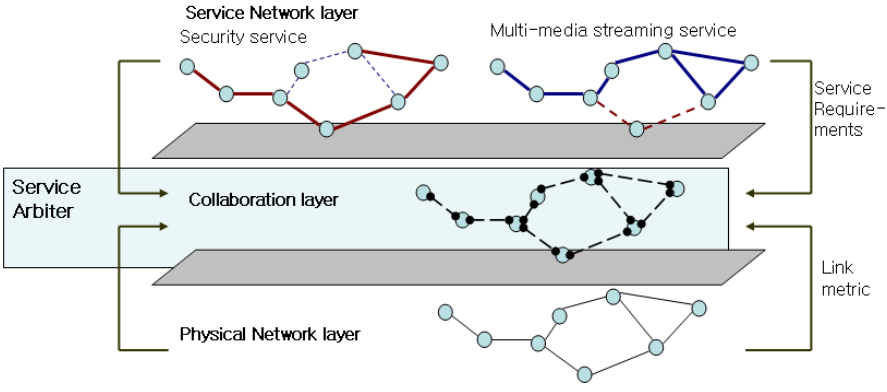


Fig. 1. Conceptual view of service oriented network

from the upper layer (user services) and constructs a service-specific network topology. The collaboration layer also gets link information, called link metric, from physical network layer. In the collaboration layer, there is a service arbiter that constructs a service specific network with the link metric information.

For example, we can imagine a following scenario for the service oriented network. A user wants to watch a movie. In the middle of the process, user authentication is required because multimedia service is a high-cost service and only valid user should access the service. Once authentication is completed, the codec for the media contents is required because codec influences the quality of the media streaming display, and it is highly dependant on the device processing power and media data size. If codec is not found in the local display device, the codec can be downloaded from the codec database. Finally, the multimedia application contents are streamed through a network. Streaming service requires realtime processing of contents and delay jitter is very sensitive factor in realtime data process.

Collaboration layer receives metrics using a simple script language. User or application service specifies his requirements, and the requirement specification is processed via service arbiter. Therefore, we can provide the logical view of the network which can support specific application service we targets on.

Although the language approach provides good flexibility, this method has a weakness. If the language is an interpretation one, we need a virtual machine or interpreter to execute the instruction and the performance will be quiet poor. If it is a compilation language, we need to compile before setting the configuration.

In both cases, there should be a structure for checking the input grammar, which is called parser in the compilation language. We develop intermediate structure for the metric calculation is developed in order to reduce the grammar check overhead. Using Lex and Yacc, we implemented software that is an intermediate code generator from the metric definition language.

Once the routing metric is specified, the service oriented network infrastructure prepares a code generator for the metric definition. Specifically, the input

string of the cost definition is parsed and makes a stack structure. When we evaluate the routing cost for the specific route, metric factors come from the symbol table. Each metric factor is previously calculated and stored in the symbol table. By interpreting the cost function, the structure has flexibility and the performance problem can be resolved with the help of calculator which is created when we parse the cost function.

4 Route Construction and Management

4.1 Routing Metrics

In service oriented network, we need a route construction rule for each routing metric. The reason why an existing ad-hoc network cannot provide the different characteristics is that all the nodes have only one routing metric. Consequently, we have to define new routing metrics that reflect service characteristics and distribute them to all the nodes in the network. In service oriented network, any node can create a new routing metric, and all the nodes should understand and calculate the proper cost function of the metric.

Service oriented network uses two features. One is dynamic routing metric processing. Application service requirement is expressed in simple script language, and it is processed via service arbiter to calculate cost for a route. It allows complex and dynamic metric calculation so that the calculation of multiple metrics involves quiet a large overhead such as parsing the metric definition and processing overhead. Using the intermediate code, it significantly reduces the overhead such as run-time parsing of requirement specification.

The second feature is two combination rules for multiple metrics. The combination rules unify multiple metrics to a new single routing metric. These rules are essentially a normalization scheme for multiple routing metrics. The first combination rule is to deal with different scales of metrics. For example, the hop count metric has a range of integer value, but remaining battery metric has a range of real number within 0 and 1. If we add two metrics, the integer (hop count) will dominate the combined metric, and the remaining battery metric is ignored. Combining multiple metrics gets more difficult when the range of a metric is not known. Two examples of hop count and remaining battery metrics are normalized as follows.

- Hop count metric
  Hop count is one of the most popular metrics. According to the combination rule, the hop count is normalized as the ratio of minimum number of hops between the route that includes the specific path and the optimal route (minimum number of hops to destination). By taking the relative value of number of hops, we can easily set the range of the metric value. Hop count metric function f can be expressed as in Eq.(1).

  $H_{dmin}(i)$ = (Minimum number of hops from node i to destination node d).
  $H_d(i,j)$ = (Minimum number of hops from node i to destination node d, including path from node i to node j).

$$F_d(i,j) = \frac{H_{dmin}(i)}{H_d(i,j)}, \tag{1}$$

where j is a neighboring node of i.

So, the combination rule gets a value between 0 and 1 for the hop count metric, and a lower value means a better route.

– Remaining battery metric
This metric selects the maximum value among remaining batteries of each node in a route. Then it is normalized as in Eq. (2), and a bigger value means a better route.

$B_d(i,j)$ = (Minimum battery remaining on the route
from node i to node d including path (i,j))
,where j is a neighboring node of i.
$B_{dmax}(i) = \underset{k}{Max}(B_d(i,k))$

$$F_d(i,j) = \frac{B_d(i,j)}{B_{dmax}(i)}, \tag{2}$$

where j is a neighboring node of i.

After metrics are normalized, we need another combination rule simple combination of the normalized metrics does not result in an optimal route. The reason is that for some metrics such as remaining battery and security level, a larger value means a better route whereas in metrics like number of hops and delay jitter, a smaller value is better.

So the second combination rule is: for the metrics where the larger value is better, Eq. (3) is used to generate the unified metric.

$$Metric\_unified_{da}(i,j) = \frac{Metric_{da}(i,j)}{Max(Metric_{da}(i,k))}, \tag{3}$$

where k is a neighboring node of i.

In Eq. (3), $metric\_unified_{da}(i,j)$ means normalized metric that returns the goodness of a route from the node i to destination node d through the path (i, j).

For the metrics where a smaller value is better, the unified metric is calculated below:

$$Metric\_unified_{da}(i,j) = \frac{Min(Metric_{da}(i,k))}{Metric_{da}(i,j)}, \tag{4}$$

where k is a neighboring node of i.

Through two combination rules, $metric_unified$ of Eq. (3) and (4) is used for route calculation with multiple metrics.

4.2 Comparison with AODV

Route Construction and Management. Although AODV constructs a network on-demand, the node always chooses the route that has the least delay time. To avoid loop in a network topology, AODV does not process more than twice for packets that have same broadcast ID. Namely, AODV processes only the first routing packet for the same broadcast, and this means that all the packets arrived later are silently ignored.

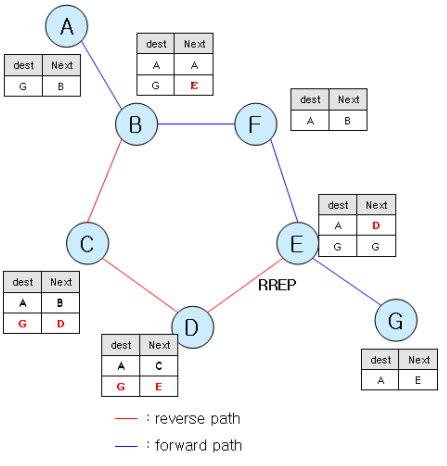


Fig. 2. Routing table update by RREP propagation

On the other hands, service oriented network should not ignore the rest of the packets. Constructing a network that takes service characteristics into acount, we have to gather the information such as security measure or remaining battery as well as delay or hop count. In AODV implementation, we consider the metric processing whenever it receives a routing message. However, we do not forward route request (RREQ) message because this message causes infinite message forwarding and loop of network topology.

To calculate each of the metric, RREQ packet carries metric definition. When a node receives a RREQ packet, the packet is forwarded as soon as possible to the next node. When the packet is being forwarded, it also carries updated metric value for each metric. In the meanwhile, intermediate code is prepared and routing metric is evaluated as to the metric value.

As a matter of fact, AODV construct a reverse path which is destined to the source node in route construction stage. Forwarded RREQ makes a reverse path and the route evaluation can be done after that all of the information is gathered. Actually, the route to the designated destination is confirmed by the RREP packet. Once the RREQ packet reaches the destination node, RREP packet is generated and returned on the reverse path which is

constructed previously. Therefore, the reverse path can be chosen after getting all of neighbor's information.

For example, as in the figure 2, let A is a source node, and we try to find a route to G, then at node D and F the RREQ packet is broadcasted. The first-comer to the E node, D or F node can create a reverse route to node A, and the RREQ from left one (F or D) is ignored. The figure presents that E chose F as next hop to the reverse route to A. RREQ broadcast from D is silently discarded and there is no table update.

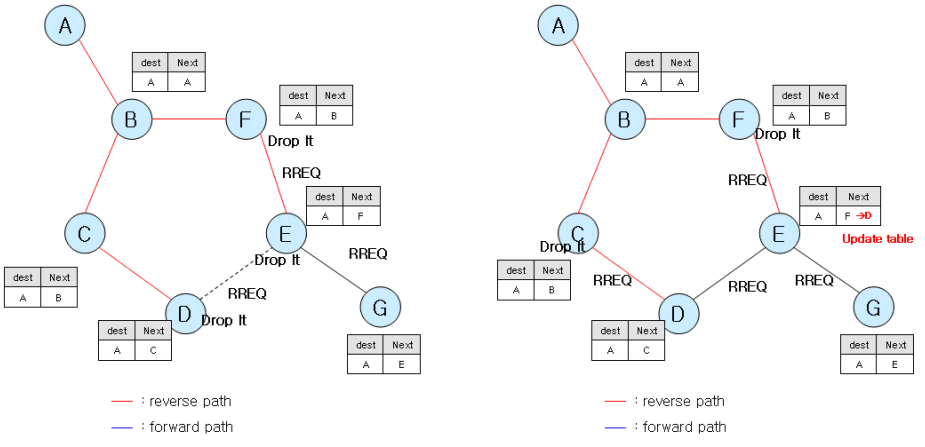


Fig. 3. Original AODV operation and AODV of service oriented network

Modified AODV is also presented in figure 3. All the packets to the node E are processed and routing table is updated. Next hop to the destination is changed to node D and in the RREP forwarding phase, E sends RREP to node D and the forwarding route is constructed as figure 3.

4.3 Simulation Results

For the performance evaluation, we have implemented the service oriented network framework on the NS-2 simulator. Simulation topology used is presented in figure 4. The following two services are used.

Metric alteration is done manually, and the metric used in simulation has chosen from the following scenarios.

- Service 1: service that has a requirement of minimum delay.
  (node 2 to node 4)
- Service 2: service to maximize the network lifetime by saving battery of nodes.
  (node 1 to node 4).

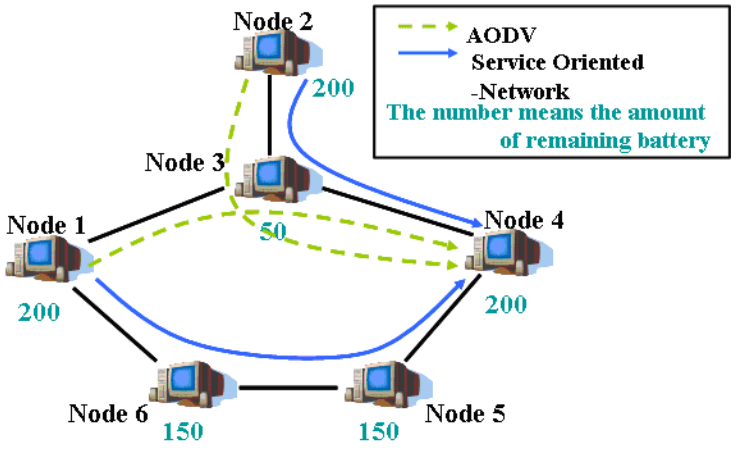


Fig. 4. Simulation scenario of the service oriented networks

Table 1. Comparison of the network life times and throughputs on session

	Network life-time(s)	Throughput on session
AODV	44	185Kbps
Our framework	127	186Kbps

For service 1, AODV calculates route: node 2 → 3 → 4, and for service 2, node 1 → 3 → 4. But service oriented network framework, because the remaining battery metric is used, the route for service 2 is node 1 → 6 → 5 → 4. The new route chosen by service oriented network framework saves battery in node 3, which extend the network lifetime. The detailed result of network lifetime is presented in table 1. The result presents that service oriented routing has similar throughput as AODV but has extended network lifetime 200% longer.

5 Conclusion

There is a growing need in ad-hoc networks for handling various and dynamic services. However, ad-hoc network construction protocols such as AODV do not consider the service characteristics. This paper develops a flexible and efficient framework for handling the dynamically changing service requirements.

Our framework has distinct features: 1) language approach to gives flexibility to specify routing metrics; 2) combination rules to unify different metrics. The simulation study reveals that incorporating service characteristics in routing is more useful for ad-hoc networks.

Acknowledgement

This research is supported by the Ubiquitous Autonomic Computing and Network project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea and partially supported by No.R01-2004-10588-0 from the Basic Research Program of the Korea Science & Engineering Foundation, and also partially supported by Korea university Grant.

References

1. C. Perkins, E. Belding-Royer and S. Das: AODV RFC 3561, Internet Engineering Task Force(IETF), July 2003.
2. David A., Maltz David B., Johnson and Yih-Chun Hu, The Dynamic Source Routing Protocol, experimental edition, July 2004. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt.
3. S. Singh, M. Woo, and C. Raghavendra, Power-aware routing in mobile ad hoc networks, In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom '98), pages 181-190, October, 1998.
4. C. Toh, H. Cobb, and D. Scott, Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks. In Proceedings of IEEE International Conference on Communications (ICC '01), June 2001.
5. Christine E. Jones , Krishna M. Sivalingam , Prathima Agrawal , Jyh Cheng Chen, A Survey of Energy Efficient Network Protocols for Wireless Networks, Wireless Networks, v.7 n.4, p.343-358, 2001.
6. Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth Belding-Royer, A Secure Routing Protocol for Ad hoc Networks, In Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP '02), November 2002.
7. Manel Guerrero Zapata, Secure Ad hoc On-Demand Distance Vector(SAODV) Routing, Mobile Computing & Communications review, v. 6 n. 3, p. 106-107, 2002.
8. Shigang Chen and Klara Nahrstedt, An Overview of Quality-of-Service Routing for the Next Generation HighSpeed Networks: Problems and Solutions, IEEE Network, Special Issue on Transmission and Distribution of Digital Video, v. 12, n. 6, p. 64-79, 1998.
9. Z. Whang and J. Crowcroft, Quality-of-Service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications, v. 14, n. 7, p. 1228-1234, 1996.
10. David J. Wetherall, John Guttag and David L. Tennenhouse, ANTS: Network Services Without the Red Tape, IEEE Computer, v. 3, n. 4, p. 42-48, 1999.
11. Christine E. Jones , Krishna M. A. B. Kulkarni, G. J. Minden, R. Hill,Y. Wijata, A. Gopinath, S. Sheth, F.,Wahhab, H. Pindi and A. Nagarajan, Implementation of a Prototype Active Network, In Proceedings of the 1st IEEE Conference on Open Architectures and Network Programming (OPENARCH 98), April 1998.
12. Sanjai Narain, Thanh Cheng, Brian Coan, Vikram Kau, Kirthika Parmeswaran, William Stephens. Building Autonomic Systems Via Configuration, In Proceedings of 5th IEEE Annual International Workshop on Active Middleware Services (AMS '03), p. 77, June 2003.

Appendix 1: Lex and Yacc Grammar for Cost Definition Language

Grammar definition in yacc.y

```
statement_list: statement '\n'
    | statement_list statement '\n' ;
name_list: NAME
        | name_list ',' NAME ;
statement: NAME '=' expression
    | expression
    | DEF_METRIC name_list
    | DEF_COST_FUNC '(' DEF_METRIC name_list ')'
      '{' expression '}'
    | eval_metric ;
metric: NAME '=' expression ; metric_list: metric
    | metric_list metric
    | metric_list ',' metric ;
eval_metric: DEF_METRIC '{' metric_list '}' ;
expression:
    expression '+' expression
    | expression '-' expression
    | expression '*' expression
    | expression '/' expression
    | '-' expression %prec UMINUS
    | '(' expression ')'
    | NUMBER
    | NAME
    | FUNC '(' expression ')'
    | '\n' ;
```

Lexical Analysis rule in lex.l

```
METRIC { return DEF_METRIC; } COST_FUNC {
    cost_function_index = 0;
    return DEF_COST_FUNC;
    }
([0-9]+|([0-9]*\.[0-9]+)([eE][-+]?[0-9]+)?) {
    yylval.dval = (double)atof(yytext);
    return NUMBER;
    }
[ \t] ; [A-Za-z][A-Za-z0-9]* {
    struct symtab *sp = symlook(yytext);
    yylval.symp = sp;
    if (sp->funcptr)
        return FUNC;
    else
        return NAME;
    }
"$" {return 0;}
\n | . return yytext[0];
```

Load Balancing Mechanisms in the MANET with Multiple Internet Gateways*,**

Youngmin Kim[1], Yujin Lim[2], Sanghyun Ahn[1,***],
Hyun Yu[1], Jaehwoon Lee[3], and Jongwon Choe[4]

[1] School of Computer Science
University of Seoul, Korea
{blhole, ahn, finalyu}@venus.uos.ac.kr
[2] Department of Information Media
University of Suwon, Korea
yujin@suwon.ac.kr
[3] Department of Information and Communications Engineering
Dongguk University, Korea
jaehwoon@dongguk.edu
[4] Department of Computer Science
Sookmyung Women's University, Korea
choejn@sookmyung.ac.kr

Abstract. A mobile ad hoc network (MANET) is an infrastructureless wireless network that supports multi-hop communication. For the MANET nodes wishing to communicate with nodes in the wired Internet, the global Internet connectivity is required and this functionality can be achieved with the help of the Internet gateway. For the support of reliability and flexibility, multiple Internet gateways can be provisioned for a MANET. In this case, load-balancing becomes one of the important issues since the network performance such as the network throughput can be improved if the loads of the gateways are well-balanced. In this paper, we categorize the load-balancing mechanisms and propose a new metric for load-balancing. Simulation results show that our proposed mechanism using the hop distance and the number of routing table entries as a load-balancing metric enhances the overall network throughput.

Keyword: Internet Gateway, Load Balancing, Mobile Ad hoc Network.

1 Introduction

A mobile ad hoc network (MANET) is a multi-hop wireless network without an infrastructure or a base station. The MANET allows mobile nodes (MNs)

* This research was supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.
** This work was supported by the University IT Research Center Project.
*** Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 207–216, 2006.

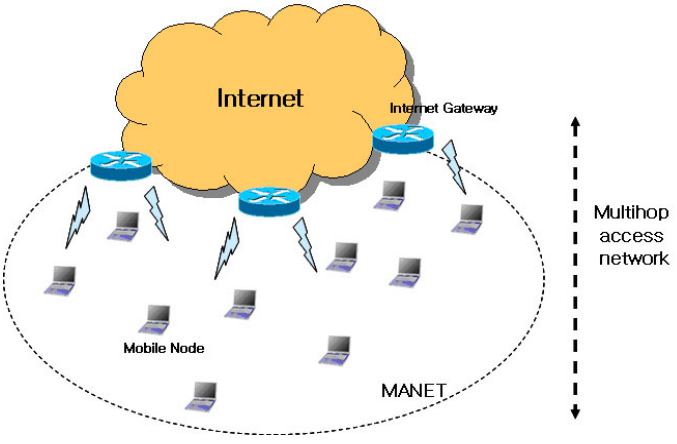


Fig. 1. Multi-hop access network

to establish a low cost, limited range network for the purpose of sharing data among them. The Internet gateway (IGW) in an access network can provide the global Internet connectivity for nodes in a MANET. The IGW belongs to both the wired Internet and the MANET and connects them. Figure 1 shows an example of the multi-hop wireless access network with multiple IGWs.

The most important issues for the support of the global Internet connectivity are handover and the Internet gateway discovery. A MN performs handover if it changes its IGW while communicating with a node in the Internet. In a traditional single-hop access network, the quality of the wireless link between a MN and an IGW determines when to handover from one to another. But, in a multi-hop access network, the situation becomes more complicated. Some nodes may be connected to an IGW via relay nodes due to the lack of direct wireless links to an IGW and, in this case, handover based on the link quality is not possible. Handover in a multi-hop access network may occur when a MN itself or any intermediate relay nodes moves. The multi-hop handover issue is out of the scope of this paper and we focus only on the Internet gateway discovery mechanism.

It is important for MNs to discover available IGWs for the Internet connectivity. There are two major approaches, reactive and proactive [1]:

- Reactive discovery
  A MN broadcasts a message which solicits the information on IGWs for the global connection within the MANET. Each IGW receiving the message replies the MN with its IP prefix address.
- Proactive discovery
  Each IGW periodically broadcasts its service and IP prefix information within the MANET. A MN receiving the message decides an IGW to connect to the Internet.

These two approaches can be combined into a hybrid gateway discovery scheme [2].

Load-balancing is one of the important issues when MNs access the Internet via multiple gateways. The network performance can be improved if the load is balanced well among the gateways. In this paper, we propose Internet gateway discovery mechanisms to improve the network throughput with balancing the load among multiple gateways.

The paper is organized as follows. In section 2, we give a brief overview of the related work. Section 3 describes our categorization of gateway discovery mechanisms solving the load-balancing problem. The results of the performance evaluation are presented in section 4, and section 5 concludes the paper.

2 Related Work

[1] proposes the mechanism to obtain a globally routable address for the global Internet connectivity and to communicate with a node in the wired Internet. All IGWs disseminate their own information, such as the IP prefix, the prefix length, and the lifetime, to the MANET using the *IGW ADVERTISEMENT MESSAGE*. A MN discovers available IGWs by receiving IGW advertisements. Each IGW may disseminate the IGW advertisement proactively, or a MN can solicit the IGW advertisement using the *IGW SOLICITATION MESSAGE* when it needs a route to the Internet. For the IGW solicitation and advertisement, modifications to NDP (Neighbor Discovery Protocol) [3] and MANET routing protocols are proposed. For the Internet connectivity, each MN needs to generate an IPv6 global address. Once a MN receives an IGW advertisement, it generates a global IPv6 address by using the IGW prefix address. Using the global IPv6 address, the MN can communicate with a node in the Internet via the IGW. When the MN receives more than one IGW advertisement, it chooses the first one received.

If all MNs and IGWs are evenly distributed in a MANET, the first received advertisement may be the best choice. But, in the real environment in which nodes are randomly distributed, the traffic concentration on a single IGW may occur while other IGWs are idle. Thus, several approaches to solve the load-balancing problem have been proposed [4][5][6].

In [4], the load index (LI) is defined to share the bandwidth of IGWs. LI is the ratio of the traffic load to the bandwidth of the interface of an IGW. Reducing the difference between the maximum and the minimum value of LI (LBI, Load-Balance Index) implies better balanced situation. But choosing a long-distance IGW to reduce LBI may cause lower network throughput due to the inefficiency of the multi-hop communication of MANET.

In [6], the concept of the dynamic gateway is proposed. A dynamic gateway acts as either an IGW or a normal MN in an alternating way. A MN accessing the Internet would select the closest and the least loaded dynamic gateway for the Internet connectivity. The number of gateways can be variable, so they mainly focus on how many gateways are required to optimize the network performance.

3 Mechanisms to Provide Load Balancing in MANET with Multiple Internet Gateways

We assume that a MANET has fixed multiple gateways for the Internet connectivity and mobile nodes move within a limited area. Our goal is to balance the load of each IGW and to maximize the bandwidth utilization of IGWs. We adopt the IGW operation for the global connectivity of [1].

Before we get into the details of our proposed mechanism, we classify load-balancing mechanisms as shown in table 1. The factors considered for the classification are the chooser of the IGW for the Internet connectivity (the chooser can be a MN or an IGW) and whether the flooding of a control message is limited or not (i.e., the expanding ring search or the maximum TTL flooding). The expanding ring search scheme selects the locally optimal IGW among nearby IGWs and, on the other hand, the maximum TTL flooding scheme selects the globally optimal IGW among all IGWs.

Table 1. Classification of load-balancing mechanisms

Flooding scheme	Selection of an IGW	
	Selected by a MN (SMN)	Selected by an IGW (SIGW)
Expanding Ring Search (ERS)	SMN-ERS	SIGW-ERS
Maximum TTL Flooding (MTF)	SMN-MTF	SIGW-MTF

3.1 SMN

In order to provide a MN with the load-balancing information which can be used for the optimal IGW selection, additional information is included in the IGW advertisement. The IGW advertisement may be disseminated proactively or reactively.

A MN may receive more than one advertisement during a certain period (TIMER_MN). The MN selects the IGW with the minimum value of IGW_i^c computed according to equation 1. When two or more IGWs have the same value, the first received advertisement is chosen. Let $i \in G$ be any IGW. IGW_i^c is defined as:

$$IGW_i^c = k \cdot H + R \qquad i \in G \tag{1}$$

where H is the hop distance between the IGW and the MN and k is the weighting factor. R is defined as the number of valid routing table entries. Under the low-load condition, H dominates IGW_i^c. When traffic is concentrated on a certain IGW, R gets increased and becomes the dominating part of IGW_i^c. Thus, the MN newly accessing the Internet selects the IGW with light load, and traffic load can be distributed among IGWs.

Figure 2 shows the modification of the prefix information option of the IGW advertisement for the inclusion of H and R [1][3]. M flag indicates that the *Hop*

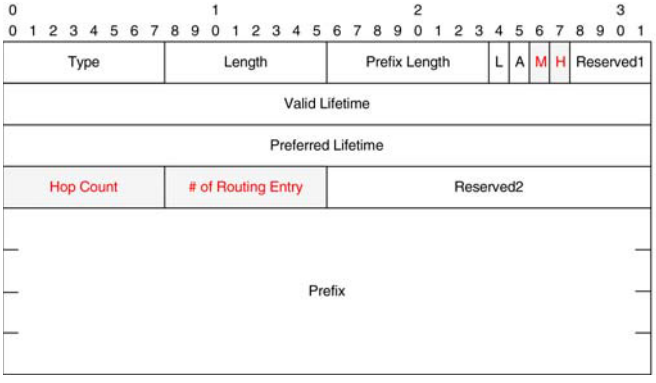


Fig. 2. Modification of the prefix information option of the IGW advertisement

Count and the *Number of Routing Entry* field are valid. *H* flag indicates that the *Hop Count* field needs to be increased by one whenever the message is processed by each intermediate MN on the path toward the MN. As explained previously, an IGW advertisement is sent out periodically or as a response to an IGW solicitation. In the case of a response to the solicitation, the *Hop Count* field has the number of hops that the IGW solicitation message has been forwarded. In the case of the periodic advertisement, the field is increased by one by each intermediate MN.

3.2 SIGW

In this mechanism, all IGWs in a MANET share the IGW_i^c information, and the only IGW with the minimum value can send an IGW advertisement message. To compute IGW_i^c, each IGW should receive an IGW solicitation message with the hop count information from the soliciting MN, as shown in figure 3. *H* flag indicates that the *Hop Count* field is valid and the *Hop Count* field need be increased by one whenever the message is processed by each intermediate MN on the path towards the IGW.

The IGW calculates IGW_i^c based on the number of its routing table entries and the hop count from the IGW solicitation, and shares the calculated information with other IGWs over the wired Internet. To share the information, the IGW can multicast using the ALL_MANET_GW_MULTICAST address [1] or other dedicated multicast address, or can unicast to each IGW in the MANET. How to share the IGW information over the wired Internet is out of the scope of this paper.

Figure 4 shows the format of the IGW signaling message newly defined to share the IGW_i^c information among all IGWs. Using the type-1 message (with $Type = 1$), each IGW disseminates its address to other IGWs. The source and the destination address of the message are the IGW address for the wireless network

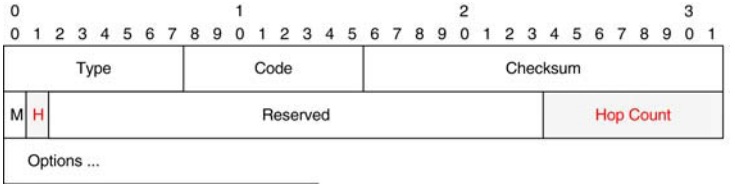


Fig. 3. Modification of the IGW solicitation

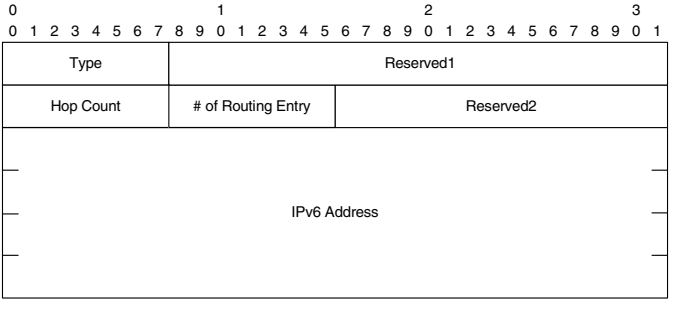


Fig. 4. Format of the IGW signaling message

interface and the ALL_MANET_GW_MULTICAST address, respectively. The *IPv6 Address* field has the IGW address for the wired network interface. By exchanging this message, each IGW can know the wireless and the wired interface address of other IGWs in the same MANET.

Once an IGW receives an IGW solicitation, the IGW sends to the wired Internet a type-2 IGW signaling message with setting valid values in the *Hop count* and the *Number of Routing Entry* field. After receiving an IGW solicitation, the IGW waits for the IGW signaling messages from other IGWs for a certain period (TIMER_IGW). After the certain period, if its IGW_i^c is smaller than the received IGW_i^cs the IGW sends an IGW advertisement to the soliciting MN. Otherwise, the IGW drops the solicitation without sending an IGW advertisement. If no IGW signaling message is received during the period, the IGW sends an advertisement to the soliciting MN.

When the IGW sends the IGW advertisement to the soliciting MN, it also sends a type-3 message to all IGWs through the Internet. The IGW receiving the type-3 message just drops the solicitation message. If the soliciting MN receives more than one advertisement, it selects the first one received.

The *IPv6 Address* field of a type-2 or type-3 IGW signaling message has the IPv6 address of the soliciting MN so that the IGW receiving the message can know for which MN this message is. The type-2 or type-3 IGW signaling message is sent through the wired Internet to the IGWs in the same MANET using some kind of wired Internet multicast mechanism or unicast. And this is out of the scope of this paper.

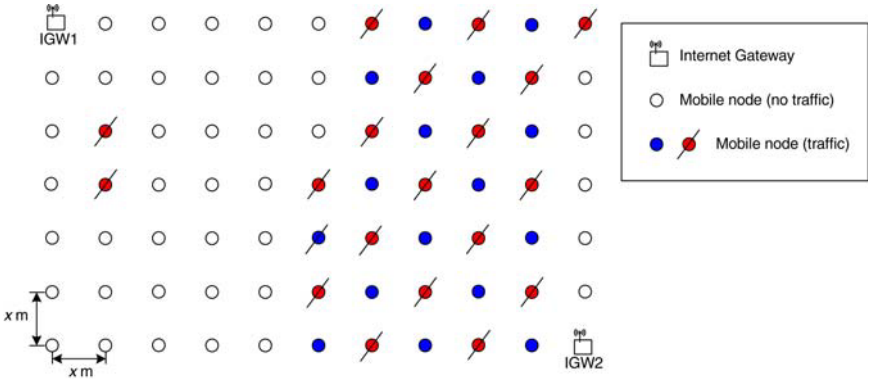


Fig. 5. An example network for the simulation

4 Performance Evaluation

In order to evaluate the performance of the classified four mechanisms, we have used the NS-2 simulator [7]. Simulations were performed for the 75-node network in figure 5. Each MN is deployed at the regular interval x (x = $100m$ or $150m$) and two IGWs can access the Internet. Most of the 19 (lined circles) or 35 (colored circles in the figure) sending MNs are deployed in the vicinity of IGW2. We assume all sending MNs want to access the Internet. The sending MNs communicate with correspondent nodes in the Internet at the constant bit rate (CBR). The size of a CBR packet is 210 bytes and the sending rate is $32kbps$. Each sending MN randomly selects the start time of data packet sending from the time interval between $1s$ and $10s$. The total simulation time is 50 seconds.

4.1 Weighting Factor *k*

To determine the optimal value of the weighting factor k in equation 1, we simulated the SIGW-ERS mechanism with varying the network size and the number of sending nodes. To adjust the network size and the number of sending nodes, we used $100m$ and $150m$ for x and 35 colored circles and 19 lined circles for the sending nodes in figure 5.

Figure 6 shows the effect of the weighting fact k on the throughput. The throughput is the sum of the received traffic on IGW1 and IGW2. In the figure, "100-19" means the throughput of the network with $x = 100m$ and 19 sending nodes, and "150-35" that of the network with $x = 150m$ and 35 sending nodes. In the network with $x = 100$m, k of 4 achieves the best throughput. The figure shows the best performance for the network with $x = 150m$ when k is 2 and 4. That means, because the average hop distance of the network with $x = 150m$ is longer than that with $x = 100m$, for the network with $x = 150m$ we can get higher throughput with smaller weighting factors. But there is no performance improvement when k is larger than 6. From these results, we can see that the

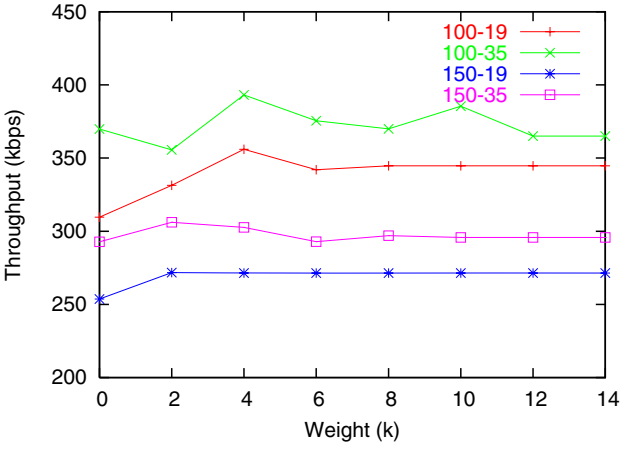


Fig. 6. Effect of the weighting factor k on throughput

weighting factor k of 4 is suitable for this environment. The optimal value of k can be different for networks with different network sizes and densities. In this paper, we choose the weighting factor k of 4 for the rest of our simulations.

4.2 Performance Comparison of the Classified Load-Balancing Mechanisms

To compare the classified load-balancing mechanisms using IGW_i^c in equation 1, we measured the throughput with varying timer values, TIMER_MN for SMN and TIMER_IGW for SIGW. We adopted the network with $x = 150m$, 35 sending nodes, and $k = 4$ for the simulation. If the timer is set to a larger value, more IGWs are considered for the Internet connectivity but it requires more time to collect the information from IGWs. On the other hand, if the timer is set to a smaller value, the waiting time is decreased but, since only the nearby IGWs are considered for the Internet connectivity, the chosen IGW may not be the optimal one.

In figure 7, SMN-ERS shows the lowest performance because it prefers closer IGWs to the IGW with the minimum IGW_i^c. SIGW-MTF shows the highest performance because the optimal IGW is chosen by sharing information among all IGWs in the MANET. From these results, we can know that the timer value of 0.1s is suitable for this environment and we choose the timer value of 0.1s for the rest of our simulations.

4.3 Performance Comparison with Load-Balancing Mechanisms using Different Metrics

In figure 8, SIGW-MTF is compared with other load-balancing mechanisms mentioned in the related work, SD (Shortest Distance) and MLI (Minimum Load

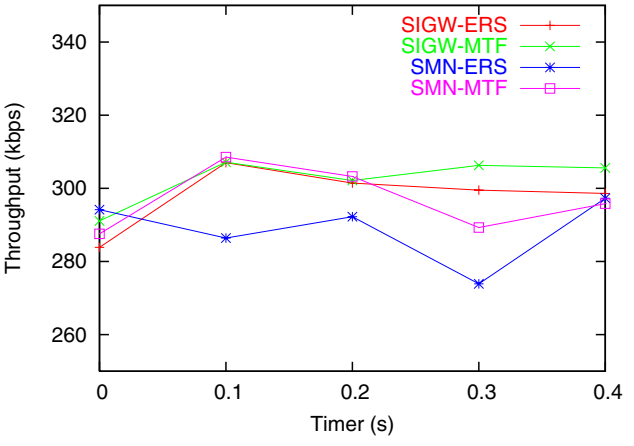


Fig. 7. Performance comparison of the classified load-balancing mechanisms

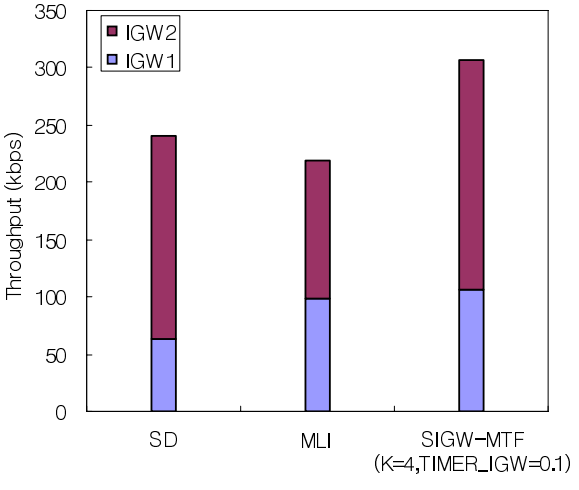


Fig. 8. Performance comparison with SD and MLI

Index) [4]. SD chooses the closest IGW without any consideration on load balancing. MLI chooses the lowest-loaded IGW for load balancing, which may lower the performance by choosing a long-distance IGW. As shown in figure 8, SIGW-MTF outperforms SD and MLI because it considers the hop distance and the number of routing table entries at the same time.

5 Conclusions

An Internet gateway (IGW) connecting a MANET and the Internet can provide the Internet connectivity for mobile nodes (MNs) in the MANET. Load-balancing

is one of the important issues when MNs access to the Internet using multiple gateways. The network performance can be improved when the load of the gateways are balanced well. In this paper, we have classified the load-balancing mechanisms into four categories and proposed a new metric to improve the network throughput by balancing the load among multiple gateways.

Our simulation results show that SIGW-MTF achieves the best performance than other three mechanisms. Also, we have simulated two other load-balancing mechanisms using different metrics and found out that our proposed metric outperforms in terms of the network throughput.

References

1. R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson and A. J. Tuominen, "Global connectivity for IPv6 mobile ad hoc networks", IETF Internet-draft, draft-wakikawa-manet-globalv6-04.txt, July 2005.
2. P. Ratanchandani and R. Kravets, "A hybrid approach to internet connectivity for mobile ad hoc networks", WCNC 2003, vol 3, pp. 1522-1527, March 2003.
3. T. Narten, E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6 (ipv6)", RFC 2461, IETF, Dec. 1998.
4. C. Huang, H. Lee and Y. Tseng, "A Two-Tier Heterogeneous Mobile Ad Hoc Network Architecture and Its Load-Balance Routing Problem", pp. 2163-2167, IEEE VTC, Oct. 2003.
5. Y. Hsu, Y. Tseng, C. Tseng, C. Huang, J. Fan and H. Wu, "Design and Implementation of Two-tier Ad Hoc Networks with Seamless Roaming and Load-balancing Routing Capability", pp. 52-58, IEEE QSHINE, Oct. 2004.
6. J. H. Zhao, X. Z. Yang and H. W. Liu, "Load-balancing Strategy of Multi-gateway for Ad hoc Internet Connectivity", pp. 592-596, IEEE ITCC, 2005.
7. The Network Simulator, NS-2, http://www.isi.edu/nsnam/ns.

Design of Modified CGA for Address Auto-configuration and Digital Signature in Hierarchical Mobile Ad-Hoc Network*

Hyewon K. Lee[1] and Youngsong Mun[2]

[1] Dept. of Computer Engineering, Daejin University, Pocheon, Korea
kerenlee@nate.com
[2] School of Computing, Soongsil University, Seoul, Korea
mun@computing.ssu.ac.kr

Abstract. The CGA (Cryptographically Generated Address) is designed to prevent address spoofing and stealing and to provide digital signature to users without certification authority or any other security infrastructures, but fake key generation and address collision appear in flat-tiered network. To solve these critical problems, CGA defines security parameter (SEC), which is set to high value when high security is required and vice versa. Although CGA with high SEC makes attackers be difficult to find fake key and to try address stealing, it brings an alarming increase in processing time to generate CGA. On the contrary, the probability to find a fake key is high if low SEC is applied to CGA. We propose modified CGA (MCGA), which is proper to mobile ad-hoc network. The proposed MCGA has shorter processing time than CGA and offers digital signature with no additional overheads. We have settled fake key and address collision problems by employing hierarchical network structure. The MCGA is applicable to as well public networks as ad-hoc networks. In this paper, we design mathematical model to analysis processing time for MCGA and CGA firstly and evaluate processing time via simulations, where processing time for MCGA is reduced down 3.3 times and 68,000 times, compared to CGA with SEC 0 and SEC 1, respectively. Further, we have proved that CGA is inappropriate for both ad-hoc networks and public networks when SEC is 3 or bigger than 3.

1 Introduction

Mobile ad-hoc network (MANET) is a multi-hop wireless network without any prepared base station. It is capable of building a mobile network automatically without any help from DHCP servers or routers to forward or to route messages. Significant difference from other wireless and wire-lined networks is continuous excessive changes of network topology without base station. Routing protocols such as DSR, AODV, TBRPF, etc. to find shortest or optimistic route have

* This work was supported by the Korea Research Foundation Grant. (KRF-2004-005-D00147).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 217–226, 2006.

been proposed, but these protocols assume that nodes have been pre-configured before building a network. To compensate these problems, MANETConf [1], automatic node configuration protocol [2] and prophet address allocation [3] have been proposed. [1] proposes address allocation and duplication avoidance in flat-tiered network, [2] proposes enhanced node auto-configuration protocol in hierarchical network and [3] suggests new IP address allocation algorithm, namely prophet allocation, which is expectable by initiating node; however, these protocols assume that address pool is already defined and they do not consider what kind of address is used in ad-hoc network.

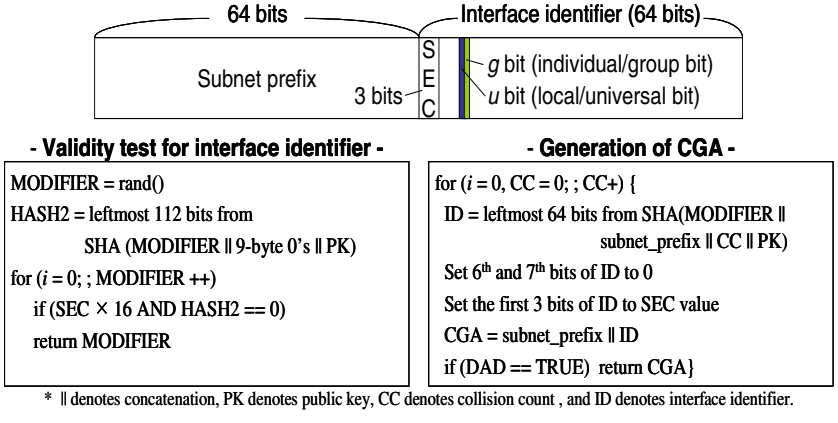


Fig. 1. CGA format

CGA (Cryptographically Generated Address) is designed to solve address spoofing and stealing attacks in IPv6. CGA offers digital signature without additional key opening process or help from certificate authority (CA), which is proper to mobile ad-hoc nodes that have low processing power and memory capacity. However, fake key generation and address collision appear in flat-tiered network due to 64-bit-taken operation from SHA's original output, as shown in Fig. 1. To solve this critical point, CGA defines 3-bit security parameter (SEC) field within IPv6 address and allows a node to generate address only when the specific condition[1] [4] in Fig. 1 is satisfied. When SEC is set to high value, it becomes more difficult for attackers to find fake key pair corresponding to origin key pair, but processing time to generate CGA increases incredibly, which eventually brings about high delay and defers communication. On the contrary, the probability to find a fake key is high when low SEC is applied to CGA. We propose modified CGA (MCGA), which is adjusted for mobile ad-hoc network. MCGA has shorter generation delay than CGA and offers digital signature with

[1] We call it as validity test for interface identifier in distinction from address validity test in section 2.4. The first test is done by an individual node on address generation, and the other is done by a receiver.

no additional overheads. We have settled fake key and address collision problems by employing two-tiered hierarchical ad-hoc network from [2], where network is divided into two levels; agent and consignors. One agent and more than one consignor organize a Mobile Unit (MUnit). An agent manages MUnit and performs duplication check on behalf of consignors, and consignor selects one of the nearest agents. All members in the same MUnit know each other, but only agent knows others in the other MUnits. This architecture ensures facility for fast and eligible duplication address detection (DAD). MCGA is applicable to both public networks and ad-hoc networks. In this paper, we first design mathematical model to analysis processing time for MCGA and CGA and evaluate processing time via simulations.

2 Proposed Modified CGA (MCGA)

Unlike CGA, modified CGA removes SEC field from IPv6 address format, which improves away validity test, shown in Fig. 1. To settle fake key and address collision problems, hierarchical network structure is employed from [2]. Besides, MCGA offers digital signature like CGA.

2.1 MCGA Format

MCGA is composed of 64-bit subnet prefix learned from network and 64-bit interface identifier generated by individual node. For interface identifier, a random number or NIC (Network Interface Card) address may be used. For subnet prefix, local-scoped prefix, FE80::/64 is used. When a node moves and gets different kinds of IEEE 802.11 [12] beacon message, then it may accepts new subnet prefix and will get connectivity to the outside. The MCGA format is similar to CGA format in Fig. 1 except the SEC field.

2.2 MCGA Generation

SHA is generally known to be much stronger than the other hash algorithms. The CGA only takes the leftmost 64 bits from SHA's output, and collision avoidance should be considered, rather than strong algorithm. MCGA employs MD5 algorithm [8] instead of SHA, because MD5 is simpler than SHA and has short processing time. To generate key pairs, RSA algorithm is employed. MCGA generation process is detailed as follows:

1. Build key pair using RSA algorithm.
2. Generate a random number or use NIC address for MODIFIER, and set collision count to 0.
3. Concatenate MODIFIER, collision count and user's public key, and put the concatenation into MD5.
4. Take the first 64 bits from 128 bits key value generated by MD5, and set interface identifier to them.
5. Set the u and g bits of interface identifier to 0.

6. Concatenate subnet prefix and interface identifier, and put the concatenation into MCGA
7. If DAD is done successfully, allocate the MCGA to interface. If the check goes wrong and if collision count is equal to 3, then go to step 2. Else, add 1 to collision count and go to step 3.

2.3 Considerations for Address Duplication

Once a node enters into a network, it generates MCGA as explained in section 2.2 and assigns MCGA to its interface unless no duplication is found in the network. For duplication check, ad-hoc node will request duplication check to the nearest agent, which lookups its resource table and gives appropriate answer to the requester. If requested address is not registered, the agent will give positive answer to the requester, and vice versa. In case of non-registered address, the agent will ask the remaining agents for duplication check [2]. If no duplication is found, the agent will get positive answers from others, and the agent will give final positive answer to the requester. When opti-DAD (Optimistic Duplication Address Detection) [9] is employed, the requester is able to start communication with other nodes after the first positive answer from the agent.

When there are n nodes in network, the probability that at least 1 fail (duplication) occurs in n MCGA generations can be expressed as (1). For instance, if 1000 MCGAs are generated, expected failure is 0. Especially, in hierarchical network, a node in logically higher position holds information about all address resources in network, so duplication check between two nodes in the different logical positions seems to be enough. In this paper, opti-DAD [9] is employed for DAD. In opti-DAD, a node is able to initiate communication before completion of DAD. Once duplication is detected, address generation and DAD should be resumed. Hence, [9] recommends that opti-DAD be prohibited in links where duplication ratio is relatively high.

$$1 - \frac{\binom{2^{64}}{n} n!}{2^{64n}} \tag{1}$$

An ad-hoc node is able to initiate communication with others using on-pending address that is still under DAD, which reduces delay due to long DAD process. Even though duplication ratio for address generation is very low, unallocated address may go to on-pending state concurrently by different nodes, and priority from arbitrary contention algorithm can be used. Besides, address stealing problem can be cropped up, and it is considered in section 2.6.

2.4 Validity Check for Received Message

When a node uses MCGA as its identifier, every message generated by this node should contain all parameters which have been used to generate MCGA, such as MODIFIER, public key, collision count, and etc. These parameters help other nodes to verify that MCGA is built properly and genuinely. When a node

receives any message, it should perform validity test for each message whether it is generated properly and genuinely, as shown in Fig. 2. Only verified message will be passed to the upper layer or forwarded to the next node. If non-agent node operates as a forwarder, then it may simply forwards the message to the next node without validity test.

Table 1. Information kept by each ad-hoc node for corresponding node

Field	Description
Address State	Allocated/on-pending/freed address
Address	Corresponding node's IP address
Physical address	Corresponding node's physical address
Registered time	Time to be allocated / on-pending / freed
Public key	Specify public key for corresponding node
Lifetime	Specify lifetime of this record
Num. of hop count	Specify the number of hop count

Each node keeps correspondent node's address information, such as {address state, address, physical address, registered time, public key, lifetime, number of hop count}, and each field is specified in Table 1. This information should be kept with valid life-time, and invalid record should be deleted. If a node receives a message whose parameters are not identical to its binding information, it should drop the message.

2.5 Digital Signature Using MCGA

When MCGA is used as a source address, user's public key is contained within every message. Consequently, no additional message exchange is necessary to open each node's public key to public. Now, message sent from MCGA can protected by attaching public key and parameters and by signing the message with the corresponding private key [4]. When a message is important and should be secured, it may be encrypted by receiver's public key and transmitted to network.

2.6 Considerations for Collision Problem

If there are a hash function ($h()$) and two different inputs (m_1, m_2), $h(m_1) \neq h(m_2)$ is true. Picking specific part from hash's output, though, provokes collision. When we think of CGA, 2^{96} cases are mapped to one 64-bit identifier, mathematically. The probability for collision is proved by 'birthday problem,' and it becomes 0.63 [10].

As specified above, collision appears with pretty high rate, attackers are easily able to build fake key pair corresponding to origin key pairs by brute-force. Once fake key pair is found, an origin node encounters with address spoofing or

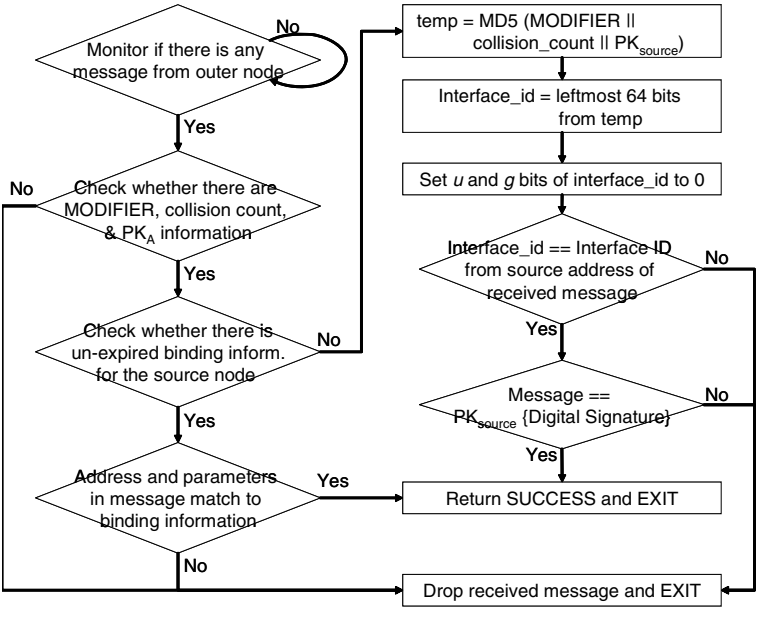


Fig. 2. Validity test for address of destination address field in a received packet

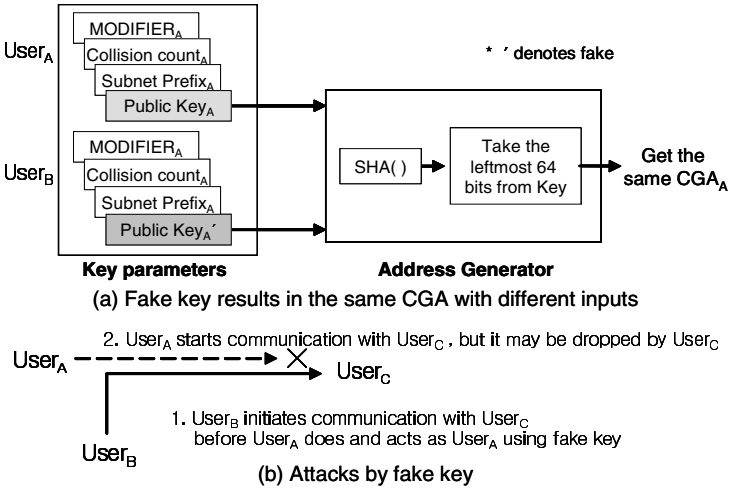


Fig. 3. Example: address collision and attack by fake key

stealing attack. An example is illustrated in Fig. 3. User$_A$ builds its key pair and generates its CGA$_A$. If an attacker, User$_B$ finds a fake key pair which yields the same CGA$_A$ and begins to send message to User$_C$, User$_C$ will perform validity test for received message from User$_B$ and identify User$_B$ as proper owner for

CGA_A. Unless $User_A$'s key is disclose, $User_B$ cannot mimic $User_A$'s signature nor can it decrypt any message encrypted with $User_A$'s public key, however, address stealing induces other nodes to wrong communication. To avoid this, [4] defines SEC field and validity test for interface identifier. Even if high SEC is applied, once fake key to generate the same CGA is found, the same problem is caused. Unless CGA is applied to hierarchical network, address collision problem cannot be avoided.

When MCGA is applied to hierarchical ad-hoc network, a node in logically higher position, agent, holds information about all address resources in network. If a stranger sends any message with different parameters for registered address, intermediate agent will notice malicious message transmission and drop the message. Let's back to the above example, where $User_A$ is real owner of $MCGA_A$, and $User_B$ finds fake key pair and starts to send message to $User_C$. If $User_A$ and $User_C$ locate at the same region (MUnit), $User_C$ notices that the message from $User_B$ is strange and drop it, because all nodes know their neighbors and parameters for MCGAs within the same MUnit. If $User_A$ and $User_C$ locate at different regions, any message from $User_B$ will be dropped by any intermediate agent between $User_B$ and $User_C$. Besides, to defend replay attack, timestamp can be used.

3 Performance Evaluation

For performance evaluation, we design mathematical model firstly and shows that MCGA has shorter processing time than CGA. Then, we compare and analysis processing time for MCGA and CGA through simulations.

3.1 Modeling

Processing time for CGA is the sum of requisition time of proper MODIFIER, generation time of interface identifier and delay due to duplication check. When we think of m duplications, the generation time of CGA (L_{CGA}) can be expressed as (2). Processing time for MCGA is the sum of generation time of random number for MODIFIER, generation time of interface identifier and delay due to duplication check. Unlike CGA, no validity test for interface identifier is required to generate MCGA. When we assume m duplications to build MCGA, the generation time of MCGA (L_{MCGA}) can be expressed as (3). Variables are explained in Table 2.

$$L_{CGA} = \lfloor \frac{m}{2} + 1 \rfloor l_{MOD} + (m+1)(l_{SHA} + l_{DAD}). \tag{2}$$

$$L_{MCGA} = \lfloor \frac{m}{2} + 1 \rfloor l_{RV} + (m+1)(l_{MD5} + l_{DAD}). \tag{3}$$

$$2l_d(d + \frac{8s}{b}) \leq l_{DAD} \leq 2l_d(d + r + \frac{8s}{b}). \tag{4}$$

Table 2. System parameters

Variables	Description
l_{DAD}	Delay to perform duplication address detection
l_{RV}	Processing time to generate random number
l_{MOD}	Processing time to get appropriate MODIFIER
l_{SHA}	Processing time of SHA function
l_{MD5}	Processing time of MD5 function
m	Number of duplication for address generation

From (1), we assume that the number duplication during address generation is 0. To find adequate MODIFER must take more time than to generate random number ($l_{MOD} \gg l_{RV}$). Especially, as SEC is set to high value, to find specific random number is more difficult [4]. It is known that to process SHA takes more time than to process MD5 ($l_{SHA} \gg l_{MD5}$). Besides, delay due to duplication check after address generation (l_{DAD}) is determined by the network size. From convergence time from [11], l_{DAD} can be expressed as (4). The longest path for data exchange is the maximum of shortest path in network or network (l_d). Transmission time on single link which transmits b bits per second for s-byte message is $8s/b$. Besides, processing delay before data transmission is assumed as d ms or $d + r(0 \leq r \leq 1)$ ms. When opti-DAD is employed, l_d becomes 1-hop distance.

3.2 Simulations

System for this simulation has following resources; CPU Pentium 4.3GHz and Memory 1GB. For operating system, Linux is employed, especially Kernel 2.4.

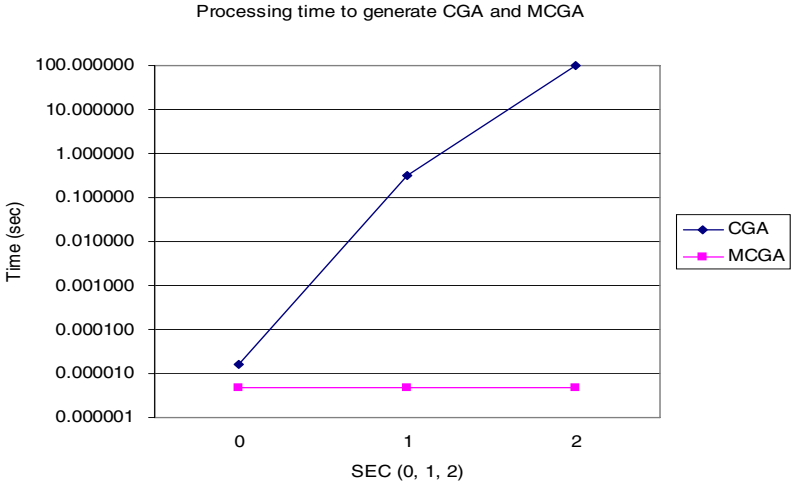


Fig. 4. Processing time for CGA and MCGA, respectively

Fig. 4 shows variation on generation time of 3000 CGAs and 3000 MCGAs. The SEC field is set to 0, 1 and 2, respectively.

The average execution time for MCGA is 4.77 μs while the average execution time for CGA when SEC is set to 0 is 15.57 μs. μs is very small, but address generation by ad-hoc nodes will need relatively more time. For example, 400 MHz ad-hoc node will perform the process 10 times slower than the above system. No address duplication is occurred in both CGA and MCGA generations. Propagation delay is not considered in this simulation.

Fig. 4 clearly proves that processing time for CGA increases dramatically when SEC increases, and processing time is strongly affected by SEC. When SEC is set to 3, it requires more than 200 hours. CGA with larger than 3 seems to be inappropriate for both public network and ad-hoc network.

4 Conclusions

The CGA (Cryptographically Generated Address) is designed to solve address spoofing and stealing in IPv6. The CGA offers digital signature without additional key opening process or help from certificate authority (CA), which is proper to mobile ad-hoc nodes that have low processing power and memory capacity. However, fake key generation and address collision appears in flat-tiered network, so CGA introduces security parameter (SEC) and uses high SEC when high security is required. Although CGA with high SEC makes attackers be difficult to find fake key and to attempt address stealing, it brings an alarming increase in processing time to generate CGA. On the contrary, the probability to find a fake key is high if low SEC is applied to CGA. We propose modified CGA (MCGA) which is proper to hierarchical ad-hoc network. The proposed MCGA has shorter processing time than CGA and offers digital signature with no additional overheads. To solve fake key and collision problems, we adopt hierarchical network structure. The MCGA is applicable to as well public networks as ad-hoc network. Simulations show that processing time for MCGA is reduced down 3.3 times and 68,000 times, compared to CGA with SEC 0 and SEC 1, respectively. Further, we have proved that CGA with SEC 3 is inappropriate for both ad-hoc and public networks.

References

1. Nesargi, S. and Prakash, R., "MANETconf: Configuration of Hosts in a Mobile ad Hoc Network," Twenty-First Annual Join Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE, 2002
2. Lee, H. and Mun, Y., "Node configuration Protocol based on Hierarchical Network Architecture for Mobile Ad-Hoc networks," ICOIN 2004, Lecture Notes in Computer Science 3090, 2004
3. Zhou, H., Ni, L. and Mutka, M., "Prophet Address Allocation for Large Scale MANET," Twenty Second Annual Join Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE, 2003

4. Aura, T., "Cryptographically Generated Address," RFC 3972, IETF, 2005
5. Vaidya, N., "Duplicate Address Detection in Mobile Ad Hoc Networks," MobiHoc'02, 2002
6. Misra, A., Das, S., McAuley, A. and Das, S., "Autoconfiguration, Registration, and Mobility Management for Pervasive Computing," IEEE Personal Communication, August, 2001
7. Eastlake, D., and Jones, P., "US Secure Hash Algorithm," RFC 3174, IETF, 2001
8. Rivest, R., "The MD5 Message-Digest Algorithm," RFC 1321, IETF, 1992
9. Moore, N., "Optimistic Duplicate Address Duplication for IPv6," work in progress, IETF, 2004
10. http://physics.harvard.edu/probweek/sol46.pdf, "the birth problem," Solution Week 46
11. Kulik, J., Heinzelman, W. and Balakrishnann, H., "Negotiation-Based Protocols for Disseminationg Information in Wireless Sensor Networks," 2002
12. Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirement- Part 11: IEEE Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANS/IEEE Std 802.11, 1999 Edition

A Power Control MAC Protocol Based on Fragmentation for 802.11 Multi-hop Networks

Dongkyun Kim[1], Eunsook Shim[1], and C.K. Toh[2]

[1] Department of Computer Engineering,
Kyungpook National University, Daegu, Korea
dongkyun@knu.ac.kr, esshim@monet.knu.ac.kr

[2] Department of Electronic Engineering,
University of London Queen Mary, UK
ck.ieee@doctor.com

Abstract. In order to reduce energy consumption at the 802.11 based MAC layer for MANETs (Mobile Ad Hoc Networks), there exists an approach to use the maximum power during RTS-CTS exchange and compute the required amount of power in order for DATA and ACK messages to reach the receiver and sender. However, it does not consider the existence of an interference range, which results in a collision at sender. Although another approach forces nodes located within a sender's interference range to defer their transmission trials in order to avoid collisions at the sender, it does not consider possible collisions at the receiver, which requires frequent retransmissions and hence greater energy consumption. In this paper, we propose an efficient protocol called F-PCM (Fragmentation-based Power Control MAC) which utilizes the fragmentation mechanism of the IEEE 802.11 MAC protocol to avoid collisions at both sender and receiver. Extensive simulations show it has better performance in terms of higher throughput and lower energy consumption, particularly in a dense network environment with high collision.

1 Introduction

Mobile ad hoc networks (MANETs) [1] are multi-hop networks in which mobile nodes cooperate to maintain network connectivity and perform routing functions. Particularly, since mobile nodes spend a great deal of energy in forwarding packets, efficient techniques to minimize energy consumption are definitely needed. Besides much research work to save a node's energy by switching the node into the "sleeping mode" whenever it does not participate in forwarding the packets [2] [3], the energy can be saved by dynamically adjusting the transmission range of nodes according to the distance between two communicating nodes over a given wireless link, instead of using fixed transmission ranges [4] [5]. This paper adopts the latter approach. The two approaches can be possibly combined in order to have more energy saving.

Although the BASIC (Basic Power Control MAC Protocol) scheme [4] tried to reduce energy consumption at nodes, it did not consider the existence of the

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 227–236, 2006.

interference range (or called carrier sensing zone)[1]. In PCM (A Power Control MAC Protocol) [5], although the interference range was considered to avoid collision at a sender, the possible collision at a receiver node was not addressed in the scheme.

We therefore aim to reduce the number of collision at both sender and receiver by taking the interference range into account. An efficient power control protocol, called F-PCM (Fragmentation-based Power Control MAC)is introduced by taking advantage of a fragmentation technique used at the IEEE 802.11 MAC layer. The rest of this paper is organized as follows; In Section 2, we describe shortly the IEEE 802.11 MAC protocol whose RTS-CTS-DATA-ACK exchange is used in F-PCM, PCM and BASIC. We discuss the disadvantages of BASIC and PCM in Section 3. Our F-PCM technique is given in Section 4. In Section 5, we evaluate the performance of F-PCM. Finally, some concluding remarks are presented along with a plan for future work in Section 6.

2 IEEE 802.11 MAC Protocol

Since carrier sensing used in the packet radio network is dependent on location of nodes, the well-known hidden terminal problem can occur in MANET, resulting in collision on data transmission. For the purpose of resolving the hidden terminal problem and providing reliable data transmission, IEEE 802.11 MAC protocol uses a four-way exchange, RTS (Request to Send)-CTS (Clear to Send)-DATA-ACK [7]. All nearby nodes receiving either an RTS or a CTS maintain Network Allocation Vector (NAV) which indicates the remaining time of the on-going transmission session. During their NAVs, they are not permitted to transmit their packets.

In particular, the existence of the interference range (also called carrier sensing zone) makes the MAC protocol more complex. Nodes in a node's interference range cannot decode a received data successfully because it is hard to decode the data. Therefore, nodes in the interference range of an RTS-sending or a CTS-sending node simply defer their transmissions with their own NAVs set to the EIFS (Extended Inter-Frame Space) period defined in IEEE 802.11 standard.

In addition, IEEE 802.11 allows a large DATA, called MSDU (MAC Service Data Unit), to be fragmented into small fragments in order to improve reliability since a large DATA is more susceptible to channel error than a short one.

3 Related Work

In the BASIC (Basic Power Control MAC Protocol) scheme [4], the RTS and CTS packets are transmitted with maximum power P_{max}. The RTS-CTS handshake is used to decide the transmission power for subsequent DATA and

[1] When node $n1$ is in the carrier-sensing zone of node $n2$, node $n1$ can sense the signal from node $n2$, but the received signal strength is not high enough to decode it correctly.

ACK packet. Although there are several possible methods, we describe one of them in this paper (see [4] for details). Sender node X transmits the RTS with maximum power P_{max}. This RTS is received at the receiver with signal level P_r. The receiver node Y can calculate the minimum required transmission power level $P_{desired}$ for the DATA packet, based on the received power level P_r, the transmitted power level P_{max}, and the noise level at receiver Y. Node Y then specifies this $P_{desired}$ in the CTS packet it transmits to node X. Node X transmits the DATA packet using power level $P_{desired}$. Similarly, the receiver uses the signal power of the received RTS packet to determine the power level to be used, $P_{desired}$, for the ACK packet. To sum up, the BA-

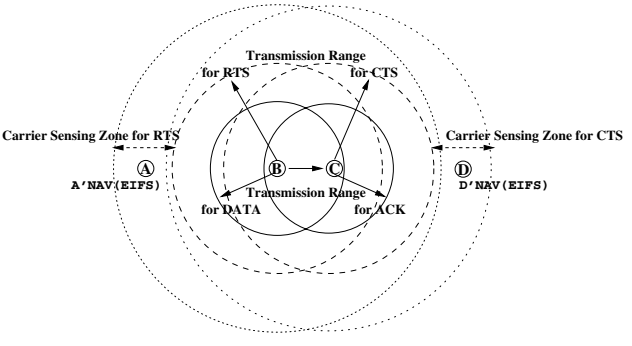


Fig. 1. The BASIC protocol

SIC scheme uses maximum transmit power for RTS and CTS packets, and consumes only the minimum necessary power for the DATA and ACK packets. However, this scheme has a drawback. As shown in Figure 1, suppose node B transmits a packet to node C. Node B sends an RTS to node C and then node C sends a CTS packet. Since these packets are sent at maximum power, nodes A and D that are in the carrier-sensing zones of nodes B and C, respectively, so it will only sense the signals and cannot decode the packets correctly. Nodes A and D will defer their transmissions for a sufficient period of time (i.e. EIFS duration) so as not to interfere with their RTS-CTS exchanges. However, since the DATA and ACK transmissions use only the minimum necessary power, the DATA transmitted by node B cannot be sensed by node A, and the ACK packet transmitted by node C cannot be sensed by node D. Therefore, if nodes A and D transmit after the EIFS period (which is set as their NAVs on sensing the RTS or CTS packet), the packet transmitted by node A would collide at node B with the ACK packet from node C, and the packet transmitted by node D would collide with the DATA packet at node C.

PCM (A Power Control MAC Protocol) [5] improves the BASIC scheme in order to minimize the probability of such collisions at sender node. The sender and receiver nodes transmit the RTS and CTS packets, as usual, with maximum power P_{max}. Nodes in the carrier-sensing zones of the sender and receiver nodes

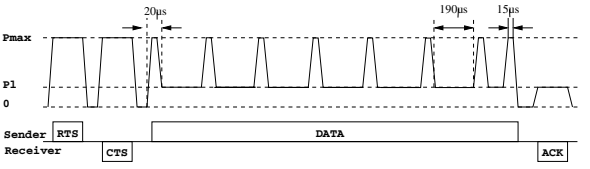


Fig. 2. The PCM protocol

set their NAVs to EIFS duration when they sense the signal but are not able to decode it. The sender node generally transmits with minimum necessary power, as in the BASIC scheme. However, in order to avoid collisions with packets transmitted by the nodes in its carrier-sensing zone, the sender node transmits the DATA packet at maximum power level P_{max} periodically. The duration of each transmission must be larger than the time required for physical carrier-sensing. Since the nodes in the carrier-sensing zone defer their transmissions for EIFS duration if they are not able to decode the received signal, the transmit power for the DATA packet is increased every EIFS duration. The changes of power level for RTS-CTS-DATA-ACK transmissions are depicted in Figure 2. Thus, this protocol prevents the collisions of ACK packets at the sender node. PCM achieves throughput very close to that of the 802.11 protocol while consuming much less energy. As mentioned before, the key difference between the PCM and BASIC schemes is that PCM periodically increases the transmit power P_{max} during the DATA packet transmission.

4 Our Fragmentation-Based Power Control MAC Protocol: F-PCM

Although PCM protocol prevents the collisions of ACK packets at a sender node, the protocol does not prevent collisions completely. Particularly, the packets transmitted by nodes in a carrier sensing zone of a CTS transmitted by a receiver can still collide with DATA being received at the receiver. As shown in Figure 1, node D is not located within the carrier sensing zone when node B sends its DATA to node C. So, if node D transmit after the EIFS time (which is set as their NAVs on sensing the RTS or CTS packet) expires, the packet transmitted by node D would collide with the DATA packet at node C.

To address this problem, we propose F-PCM (Fragmentation-based Power Control MAC Protocol) which incorporates a fragmentation technique used in the IEEE 802.11 MAC layer. Like BASIC and PCM, F-PCM allows the sender and receiver to send RTS and CTS with P_{max}, respectively. Also, ACK packet corresponding to each fragment is transmitted with maximum power. The following section explains how F-PCM works. In F-PCM, the corresponding ACK packet for each fragment forces the nodes, which have the possibility of producing collision at node C, to reset their NAVs. F-PCM can also reduce the number of those collisions at a receiver as well as at a sender.

4.1 Description of F-PCM

We will explain the operation of a sender, receiver, and nodes in a carrier sensing zone of the sender and the receiver in order to avoid the collisions. The other nodes within the transmission ranges of sender and receiver follow the procedure of the IEEE 802.11 MAC protocol in order to set their NAV values.

Basically, F-PCM takes advantage of a fragmentation technique used in the IEEE 802.11 MAC protocol and applies the concept of PCM to each fragment. In other words, the sender increases its transmission power to P_{max} during 20 μ seconds from the beginning of transmitting each fragment after the exchange of RTS-CTS with P_{max}. In addition, the receiver also transmits every ACKs except last one with P_{max} power. The ACK of last fragment is transmitted with required amount of power.

The nodes located within a carrier sensing zone of the sender or the receiver can sense these RTS-CTS and DATA-ACK exchanges, but they cannot decode it correctly. In F-PCM, when a node is located within carrier sensing range of other nodes, the node determines a new EIFS, denoted by n_EIFS, instead of the EIFS used by IEEE 802.11 MAC. F-PCM determines the n_EIFS based on the size of the fragment in order to defer the transmissions of the nodes, at least until a fragment is successfully sent to a receiver and the sender receives an ACK packet from the receiver (see Equation (1)).

$$n\_EIFS = T_{DATA\_Fragment} + T_{ACK} + 2 \times SIFS + 2 \times aSlotTime, \quad (1)$$

where $SIFS$ and $aSlotTime$ are defined in the IEEE 802.11 MAC standard and $T_{DATA\_Fragment}$ and T_{ACK} are the required times to transmit a fragment and an ACK packet, respectively. If the small EIFS defined in the IEEE 802.11 standard is used, the nodes located within a carrier sensing zone of a sender will initiate their transmissions after the expiration of the EIFS values, which can frequently collide with ACKs sent from a reciver at the sender.

In addition, when a receiver receives a DATA, the nodes can produce a collision at the receiver with their transmissions (after the expiration of their current n_EIFSs values). Therefore, whenever the receiver receives each fragment, it sends its ACK with P_{max}, which forces the nodes to reset NAVs to the n_EIFSs in order to avoid a collision with the next incoming fragment from the sender. Finally, the receiver does not need to use P_{max} in case of an ACK packet for the last fragment. Since it is the last fragment for a large DATA after RTS-CTS exchange, the nodes located within a carrier sensing zone of a receiver do not have a reason for deferring their transmissions. Thus, F-PCM allows the receiver to send its last ACK for the last fragment with the required amount of power needed to reach the sender.

In our scheme, due to the increased transmission power during a short interval for each fragment, nodes located within a carrier sensing zone of the sender or the receiver reset their NAVs to the n_EIFS periodically, as long as there exists a fragment to be sent.

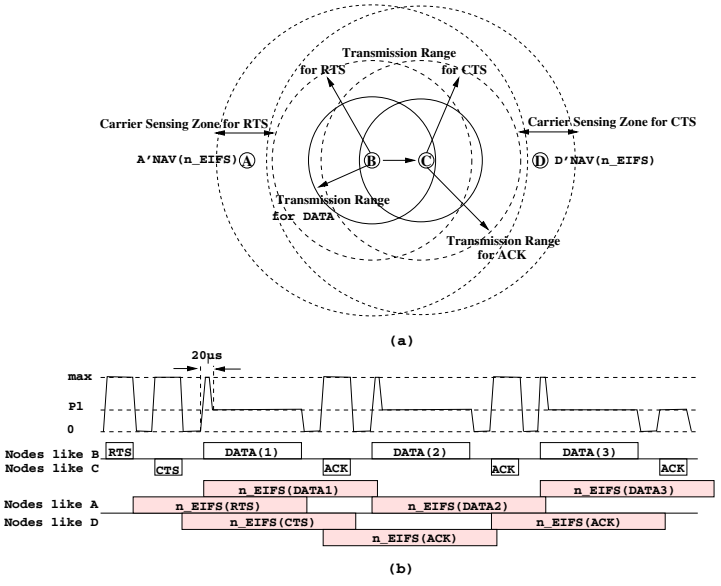


Fig. 3. Our proposed F-PCM protocol

4.2 Illustrative Example

As shown in Figure 3 (a), nodes B and C exchange their RTS and CTS with P_{max} before transmitting an actual DATA. Therefore, since nodes A and D are located within the carrier sensing zones of nodes B and C, respectively, they set their NAVs to n_EIFS in order to defer their transmissions until a DATA-ACK exchange of a fragment is performed successfully without a collision. After the exchange of RTS-CTS, nodes B and C send their DATA and ACK with the required minimum amount of energy to reach each other. However, node B increases its transmission power to P_{max} during 20 μ seconds from the beginning of transmitting the fragment whenever each fragment is sent. This forces node A to reset its NAV to n_EIFS in order to defer its transmission for the purpose of avoiding a collision at node B. Node C, which received a DATA from node B, also sends the corresponding ACK packet with P_{max}, which forces node D to defer its transmission in order to avoid a collision at node C with the next fragment from node B. Node C, however, sends the ACK packet for the last fragment with the minimum amount of power because it should not defer node D's transmission any longer. Figure 3 (b) shows when to increase nodes' power during the RTS-CTS-DATA(fragment)-ACK exchanges.

4.3 Discussion

F-PCM fragments a large DATA into small fragments, each of which is independently transmitted as an encapsulated frame with a header and trailer. The header contains the physical addresses of the sender and receiver, and the trailer

has a CRC (Cyclic Redundancy Check) for the frame. When a large DATA is fragmented, we might worry about the overhead of creating the header and trailer for each fragment. This implies that nodes will consume more energy. However, headers and trailers are not very large and furthermore, F-PCM does not require many fragments. It is understood that the frame body (MAC Service Data Unit = MSDU) has a maximum size of 2346 bytes, according to the IEEE 802.11 standard [7]. If the fragment size of 512 bytes is used, there are not many fragments for the MSDU. More importantly, although there exists additional information for F-PCM, it outperforms other protocols because it has fewer collisions and re-transmissions, as shown by the simulation results which are described in the next section.

5 Performance Evaluation

To estimate the performance of our F-PCM protocol, we developed our event-driven simulator. Network topologies were randomly created with the given number of nodes and we compared our F-PCM with BASIC and PCM in terms of the average throughput and the amount of energy expended in the network. In our simulation, the throughput is defined as the number of bytes successfully sent between the source and destination nodes in the end-to-end manner. We selected the random pairs of source and destination nodes whose connections lasted during random periods.

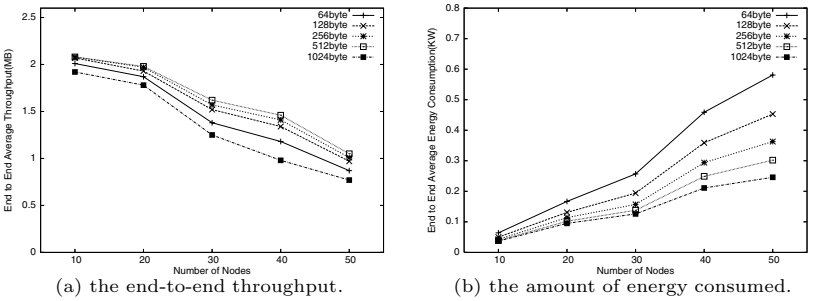
IEEE 802.11b DCF mode was used in the simulations as a basic MAC protocol. We used a network area of 1000 m x 1000 m and varied the number of nodes from 10 to 50. We assumed a transmission range and a carrier sensing zone of 250m and 550m, respectively. In order to compute the amount of energy expended, we adopted the energy model used in [8]. Other simulation parameters followed IEEE 802.11 standard [7]. We ran each simulation for 20 seconds with CBR source traffics.

5.1 Fragment Size Determination

Since F-PCM takes advantage of fragmentation technique, we attempted to find the best fragment size, considering its impact on the average end-to-end throughput and the amount of energy consumed. We measured them by varying fragment size from 64 to 1024 bytes using random network topology.

As F-PCM uses a maximum power for a short interval at the beginning of transmitting each fragment and during the ACK transmission, it forces nodes within the carrier sensing zones (of the sender and receiver) to defer their transmissions, which results in reducing the number of collisions. Therefore, we obtained similar results with different fragment sizes. However, we observed that the fragment size of 512 bytes produces the highest throughput without regard to the number of nodes (see Figure 4 (a)). When using a very large fragment size such as 1024 bytes, however, nodes located within the carrier sensing zones of the sender and receiver can neither participate in sending their data to other nodes nor receiving data from other nodes because their n_EIFSs are set to large

values based on fragment size as mentioned in Equation (1). We, thus, obtained low throughput. In addition, it is shown that as the number of nodes increases, the average throughput tends to generally decrease.



(a) the end-to-end throughput. (b) the amount of energy consumed.

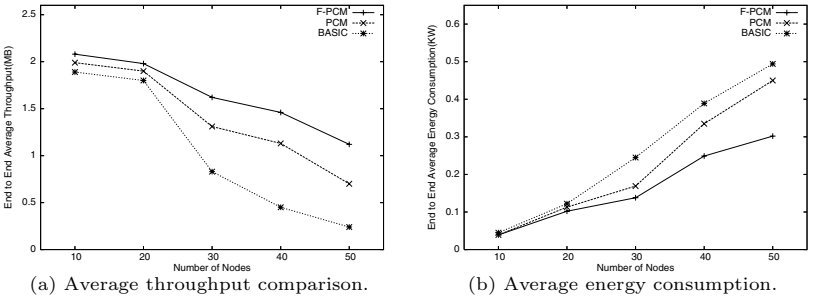
Fig. 4. Impact of fragment size on performance

In terms of energy consumption, however, the low frequency of power increases for fragment as well as ACK allows less energy to be consumed in the network. Figure 4 (b) also shows the energy consumption according to the number of nodes. As shown in the figure, as the number of nodes increases, that is, as the possibility of collisions increases, the difference of performance increases. On the other hand, we found that for small fragment size, the results are the opposite of those when using large fragments. As a result, we found that a size of 512 bytes is the most adequate in terms of performance. Thus, F-PCM requires a frame of greater than 512 bytes to be fragmented.

5.2 Simulation Results with Random Topology

First, we investigated F-PCM performance using random topology by varying the number of nodes whose positions are randomly selected. Note that the fragment size was fixed to 512 bytes according to aforementioned simulation results. We averaged the results from over 30 simulations. As shown in Figure 5 (a), when the number of nodes was large, we obtained low throughput due to a large number of collisions caused by much contention to channel access among nodes. Even in this random topology, F-PCM has better throughput performance than the others irrespective of the number of nodes, because of its ability to reduce the number of collisions at both sender and receiver.

From an energy consumption's perspective, F-PCM saves more energy since it has fewer re-transmissions (see Figure 5 (b)). In addition, we also observed that as the number of nodes increases, the performance difference increases. In other words, F-PCM shows better performance improvement in a dense network environment with high collision. In summary, F-PCM shows better throughput performance and energy saving than PCM by 24% and 17% on average, respectively.

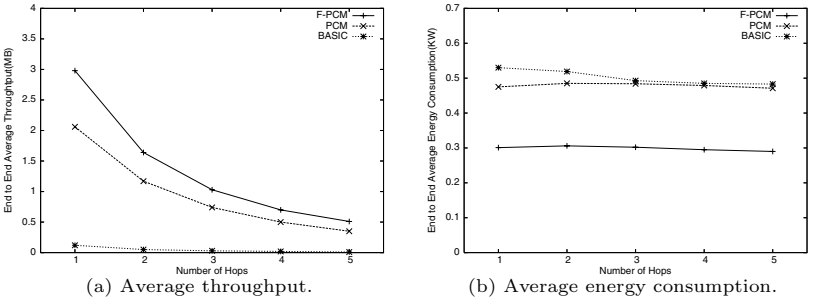


(a) Average throughput comparison. (b) Average energy consumption.

Fig. 5. Performance comparison under random network topologies

5.3 Simulation Results According to Hop Distance

Second, we investigated the end-to-end throughput and the amount of energy expended in the network according to hop distance between source and destination nodes. One of nodes which are n-hop away from a randomly selected source node was randomly chosen as its destination node (In this simulation, n increases from 1 to 5).



(a) Average throughput. (b) Average energy consumption.

Fig. 6. Performance comparison according to hop distance

Figure 6 (a) shows that F-PCM outperforms the others in terms of higher throughput irrespective of hop distance. It also indicates that when the number of hops is large, we obtained low throughput because the successful transmission of packets is more difficult to expect. From an energy consumption's perspective, more energy can be saved in F-PCM since it requires fewer re-transmissions than the others (see Figure 6 (b)). The amount of energy consumed while a packet is successfully transmitted to a far-away destination node is similar to that of energy consumed while several packets are safely transmitted to a near destination node. Therefore, we could not see a large difference between $n = 1$ and $n = 5$. In summary, F-PCM shows better throughput performance and energy saving than PCM by 42% and 38% on average, respectively.

6 Conclusion

We introduced F-PCM (Collision Prevention Scheme-based Power Control MAC) protocol to reduce collisions caused by the existence of interference range (called carrier sensing zone). F-PCM utilizes a fragmentation technique found in the IEEE 802.11 standard. RTS, CTS and ACK packets are transmitted with maximum power, and a large DATA is fragmented into several fragments which are then transmitted with the minimum amount of power required to reach the receiver. In order for nodes within the carrier sensing zone of a sender to defer their transmissions, the sender increases its transmission power to the maximum amount once at the beginning of transmitting each fragment. In addition, in order for nodes within the carrier sensing zone of a receiver to defer their transmissions, the receiver transmits its ACK with the maximum amount of power except for the last ACK, which does not require the nodes to delay their transmissions further. Extensive simulations showed that the most appropriate fragment size is 512 bytes and F-PCM outperforms BASIC and PCM, yielding throughput and energy saving improvement by 24% and 17% on average, respectively. In particular, F-PCM shows better performance than the others in a dense network environment with high collision. More simulation work considering node mobility is our future work.

References

1. Internet Engineering Task Force, "Manet working group charter", http://www.ietf.org/html.charters/manet-charter.html.
2. Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks", IEEE INFOCOM, 2002
3. B.Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An Energy Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," ACM Wireless Networks Journal, Vol. 8, 2002.
4. J.Gomez, A.T.Campbell, M.Naghshineh and C.Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks," IEEE ICNP 2001.
5. Eun-sun Jung and Nitin H.Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks," ACM MOBICOM 2002.
6. K. Zhang and K. Pahlavan, "Relation between transmission and throughput of slotted ALOHA local packet radio networks", IEEE Transactions on Communications, Vol. 40, March 1992, pp 577 - 583.
7. IEEE Computer Society LAN MAN Standards Committee. Wireless LAN MAC and PHY Specification, IEEE Std 802.11-1997.
8. L.M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," IEEE INFOCOM 2001.

Policy-Based Management in Ad hoc Networks Using Geographic Routing*

Farrukh Aslam Khan[1], Umer Zeeshan Ijaz[2], Kyung-Youn Kim[2], Min-Jae Kang[2], and Wang-Cheol Song[1,**]

[1] Department of Computer Engineering, Cheju National University, Jeju 690-756, South Korea
{farrukh, philo}@cheju.ac.kr
[2] Department of Electronic Engineering, Cheju National University, Jeju 690-756, South Korea
{umer, kyungyk, minjk}@cheju.ac.kr

Abstract. The management in mobile ad hoc networks is quite challenging as compared to management in wired networks. Several management strategies have been proposed by authors using either pro-active or reactive routing protocols. In this paper, we propose a policy-based management framework for ad hoc networks in which policy servers can efficiently communicate and service their clients using Location-Aided Routing (LAR) as underlying routing protocol. In our position-based system, all nodes including Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) have GPS capability. These nodes can estimate their positions with GPS by using Extended Kalman Filter (EKF), which also provides velocity information that is required for LAR algorithm to calculate distance traveled by a destination node. We present a dynamic clustering mechanism, a modification in COPS protocol, and a change in LAR protocol, which makes the management more efficient and effective. Our proposed system is first of its kind which uses position information in a management framework.

1 Introduction

Policy-based Network Management (PBNM) is one of the management strategies in which client nodes follow some policy for making decisions. The Resource Allocation Protocol (RAP) working group of IETF proposed a policy framework for establishing a scalable policy control model. The key elements in the model are shown in Fig.1 [11]. Since recently, people have been working on management in ad hoc networks and several frameworks have been proposed by various researchers [2], [4], [10], [12]. In [4], authors propose a hierarchical management architecture for ad hoc networks. The problem with such a hierarchical scheme in ad hoc networks is the cost of maintaining the hierarchy along with

* This work was supported by NCA Korea.
** Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 237–246, 2006.

node mobility. In [10], authors use the k-hop clustering algorithm for managing the network in distributed manner with a pro-active algorithm OLSR as the underlying routing protocol. Pro-active routing protocols send the topology information to all the nodes in the network. In ad hoc networks, the topology changes very frequently due to mobility, so the pro-active algorithm always has to maintain routing and topology information even if some routes are not used at all. Therefore, it is quite expensive in terms of computation. Geographic routing

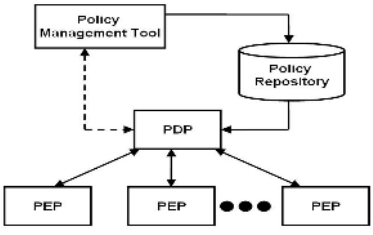


Fig. 1. Key elements of a policy-based management system

or location-based routing is another area which has received lots of attention during the past few years. Location-based routing requires the actual position of the mobile nodes to be known in advance. In order to locate the position of a wireless node, several methods have been proposed by researchers [1], [3], [8], [9]. GPS-based positioning was considered to be an expensive option by many scientists. But due to rapid advancement in the GPS technology, GPS receivers are now becoming easily available at lower costs with more accurate estimation of position. In this paper, we use the Location-Aided Routing (LAR) [7] as the underlying routing protocol in our policy-based network management framework. GPS receivers are used to locate the position of mobile nodes. In a conventional PBNM environment, sending route requests and messages can be costly as the PEPs have no information where exactly the PDPs are located. In case of position-based routing, the PEP knows the exact position of the PDP. So, it can send packets only in the direction of PDP, minimizing the routing overhead to a great extent. Also, the physical distances among nodes can be found easily. Moreover, location-based routing does not require the topology information of the whole network to be known. Especially, in case of LAR, routes are established on-demand using limited flooding which does not require maintaining and updating the routing tables.

In case of ad hoc networks, [1] and [9] explain the static model in which the anchor nodes are not mobile. Nodes perform a Least Square Method in order to locate their positions using trilateration. Since in our case, all the nodes including PDPs and PEPs are mobile and have GPS capability, simple Least Square Method is not suitable for estimating the position of moving nodes as it is used in the static case. Instead, with GPS we use the Extended Kalman Filter (EKF) [6] for estimating the position of mobile nodes. EKF is used in our

system because of two reasons. Firstly, it gives us the mobility pattern of mobile nodes which can be useful in predicting their future movements. Secondly, EKF gives us the velocity information which is required by the LAR routing protocol to determine the distance traveled by the destination node between two time intervals. Using this distance, we can anticipate the next position of the moving destination node. In Appendix, we show a detailed model for the Extended Kalman Filter.

2 Basic Policy-Based Model

It is assumed that there are enough PDPs available in the network to service their respective PEPs. In our model, every node including PDPs and PEPs know their own positions as all nodes have GPS receivers. Each PDP in the network periodically broadcasts its ID and position to all other nodes in the form of a position message. This means that every node knows the positions of all the PDPs. The client nodes store the information about PDPs in a Location Table. This information includes the PDP ID and position of each PDP. Similarly, each PDP also maintains a table of all the PEP nodes currently being served by it.

2.1 Path Selection and PDP Discovery with Modified LAR

As mentioned earlier, every client maintains a Location Table containing information of all the PDP servers present in the network domain. Any client that wants to use the policy of a particular PDP, it first selects the most suitable PDP from the Location Table. On the basis of the shortest distance, a PEP selects its PDP and sends a request message to all its neighbor nodes which are in the direction of the selected PDP server. The direction information can be determined and used by the LAR routing protocol. Using LAR's direction-based routing, we limit the number of packets sent to the neighboring nodes which reduces the routing overhead to a large extent.

Here we introduce a slight modification in the LAR protocol. In LAR, the source node sends a route request to its neighbor nodes in the direction of the destination. Upon receiving the route request, the destination node sends back a route reply message which may include its current position, current time and current speed. In our modified LAR algorithm for policy-based management, it is not necessary that the destination node should send the route reply message. Instead, any client node using the policies of the PDP server can send the route reply message back on the reverse path telling that there is a valid route available for the PDP server. If the node is not using the policies of that PDP, it just forwards the request to other nodes again in the direction of the PDP. This information makes the route discovery process faster and makes our framework work in a more efficient manner.

Upon receiving path information, the PEP sends a COPS <Client-Open> message to the selected PDP. The PDP replies back with <Client-Accept> or <Client-Close> message to the requesting node. If it is accepted, the node starts

using its policies. Otherwise, the node selects another PDP from Location Table which is periodically updated and sends the request to another PDP. The path discovery process with modified LAR is shown in Fig. 2. PEPs also send a

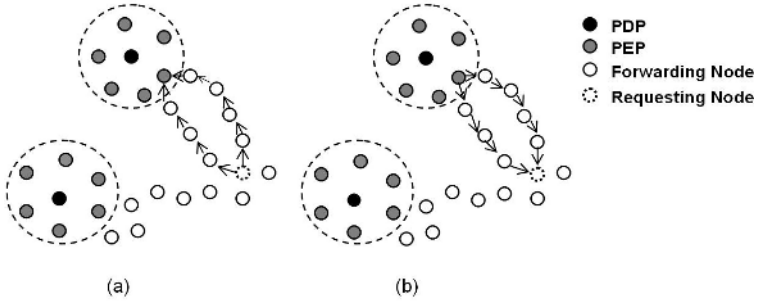


Fig. 2. (a) PEP sends a Request Message to its neighbors in the direction of a PDP. (b) PEPs using policies from PDP reply with path information (Modified LAR).

periodic Keep Alive (KA) message as in the original COPS protocol to inform about their connectivity to their respective PDP servers. In response to the KA message, PDP replies back with ECHO message.

2.2 Modifying COPS Protocol

The Common Open Policy Service (COPS) [5] protocol was proposed for efficient policy-based management in the network domains. In order to use the COPS protocol for ad hoc networks especially in location-based environment, we cannot use the original version of COPS protocol. Instead, we need to modify it according to the requirements of location-based environment as the packets are routed to the destination on the basis of actual position of the node. Therefore, the COPS headers and packet formats need to be changed and IP addresses should be replaced with actual positions of the network nodes.

Changing PDP Server. We have modified the current version of COPS and introduced another operation object as <Change-Server>. Whenever a node wants to change its current PDP and intends to join a new PDP, it cannot do it directly because the New PDP may not have enough room to accommodate another client node as it keeps a threshold value on the maximum number of nodes present in the cluster (Cluster management is explained in Section 3). So, in this case, the PEP contacts the Current PDP and sends a <Change-Server> message along with the New PDP ID. In response to this message, the Current PDP contacts the New PDP and negotiates if there is enough space to accommodate a new PEP. If the New PDP accepts then the Current PDP simply sends a <Client-Close> message to the PEP and the PEP sends a <Client-Open> request to the new PDP and joins the new cluster. The advantage of

this negotiation mechanism is that it guarantees the PEP that the new PDP has enough space to accommodate a new client. Otherwise, in case the PEP leaves the Current PDP directly and requests another PDP to join, it may not get permission to receive services from that PDP as the number of clients being served could have reached the threshold value. The COPS negotiation mechanism for the modified COPS protocol is shown in Fig. 3. In the figure, the negotiation

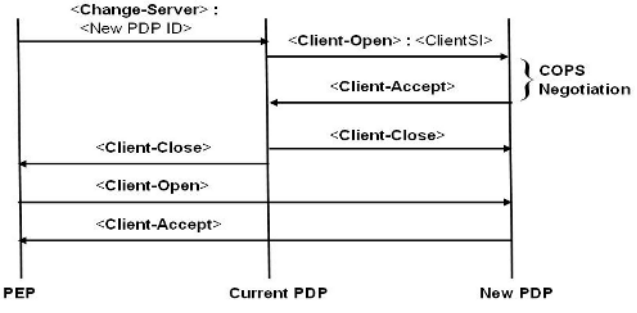


Fig. 3. Negotiation among PEP, Current PDP and New PDP using modified COPS protocol

for getting permission from the New PDP is done using the <ClientSI> object, which is used for relaying additional global information about the PEP to the PDP when required [5]. Another thing to note here is that while doing negotiations with the New PDP, the Current PDP behaves as a client node requesting for permission from the New PDP.

3 Dynamic Cluster Management

The concept of clustering for management in ad hoc networks has been used by several researchers [10] [4]. With the help of clustering, the management can be done in an efficient manner. We propose a simple and efficient clustering scheme in our position-based management architecture. As discussed in the previous section, the PEP periodically maintains a Location Table which contains the position information of PDPs. Utilizing this information; a client can calculate the Euclidean distance from PDP and join the cluster of that PDP which is at a smallest distance from the PEP. In other words, the clusters are made on the basis of Euclidean distance which can be calculated as follows:

$$d = \sqrt{(x_2 - x_1) + (y_2 - y_1)} \tag{1}$$

Since clients join a cluster based on distance, it may lead to a situation in which large numbers of nodes are physically present near one PDP trying to join its cluster and making it overloaded. This situation can be handled by setting a

maximum threshold value on number of clients joining one cluster. When a client requests to join a cluster, the PDP should check how many clients are already in the cluster. This record can be kept easily as the PEPs always send Keep Alive messages to the PDP showing their connectivity. If the number exceeds the threshold value, it should deny any more requests. In this case, the client can request the next nearest PDP to become a member. Hence, this dynamic clustering makes it easier for the servers to service their clients in an effective manner.

3.1 Selection of Cluster-Head in Case of PDP Failure

There can be a situation in which the PDP switches off or fails. In this case, the PEPs should select a node from themselves which can serve as a temporary PDP and facilitate other nodes in making policy decisions. If the PDP fails and other PDPs are very far from PEPs, the group of PEPs selects a cluster-head using the k-mean clustering algorithm [14]. The working of k-mean clustering algorithm is shown in Fig. 4. In the figure, 'c' is the total number of clusters; 'N' is the total

Repeat for l = 1, 2,...
Step 1: Compute the distances
$D_{ik}^2 = (x_k - v_i)^T (x_k - v_i)$,
$1 \le i < c, 1 \le k \le N$.
Step 2: Select the points for a cluster with the minimal distances, they belong to that cluster.
Step 3: Calculate cluster centers

$$v_i^{(l)} = \frac{\sum_{j=1}^{N_i} x_i}{N_i}$$

until $\prod_{k=1}^{n} \max | v^{(l)} - v^{(l-1)} | \neq 0$
Ending Calculate the partition matrix

Fig. 4. Working of k-mean clustering algorithm

number of nodes, $x_k = [x, y]$ i.e., position co-ordinates, $v_i = [v_x, v_y]$ i.e., centroid co-ordinates and l is the iteration number. In the original k-mean algorithm, iterations are run until there is no movement of nodes from the specified clusters. Since, in our case, we need to select one cluster-head out of all the PEPs, we need only one centroid to calculate and for that, we take the Euclidean distance of the centroid from all other nodes only once. After calculating the centroid, the PEP node which is nearest to the centroid becomes the cluster-head. In case, there are large numbers of nodes without any PDP, the PEPs can choose more than one cluster-heads using the k-mean algorithm. In this case, the process would be iterated more than once and more centroids need to be determined. After being selected as cluster-head, the node requests for policies to the nearest PDP. Upon acceptance of the request, the cluster-head downloads the policies

and starts serving other nodes as a temporary PDP. It will work temporarily as PDP because it may not have enough resources (e.g., battery, processing power etc.) to serve other nodes for long time. As soon as it finds another PDP near it, the temporary PDP relinquishes its duties as a server and starts working as a client. All other nodes which are in the threshold range of the PDP join it and the remaining nodes again opt for selecting another temporary PDP using the k-mean clustering algorithm and make a cluster. The simulation results for the k-mean algorithm are shown in section 4.

4 Simulations for Validity of EKF and K-Mean Algorithms

In order to test Extended Kalman Filter, synthetic data was generated by using Yuma almanac maintained at U.S Coast Guard Navigation Center [13]. From this data, ECEF co-ordinates of satellites were generated for 100 time steps with a sampling period of 10 seconds for mobile nodes present on the surface of earth moving at a certain speed. White Gaussian noises were added as process noise and measurement noise. Fig. 5 shows the position results obtained from Extended Kalman Filter applied to one particular node in motion.

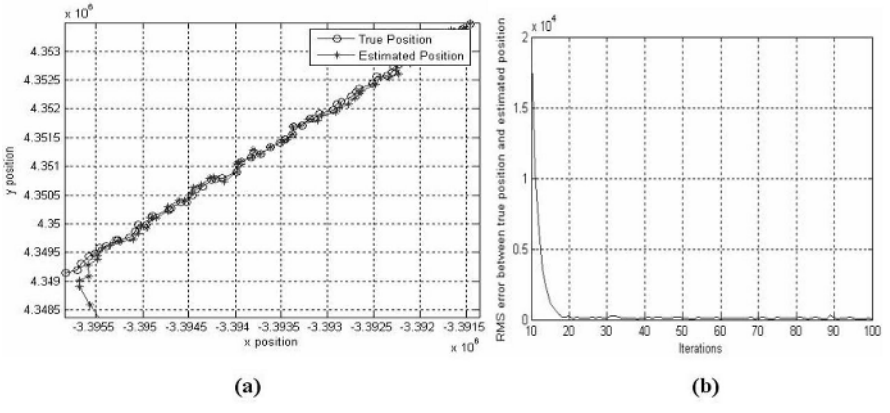


Fig. 5. (a) Comparison between true position and estimated position (b) RMS error between the true and estimated positions

The simulations show that Extended Kalman Filter for mobile nodes converges towards the true position. In this case, some transient time in start is required for the filter to stabilize. The example simulation results for k-mean clustering are shown in Fig. 6. The k-mean algorithm for this particular simulation classifies the nodes into three categories on the basis of distance from the centroid. The simulation clearly validates the k-mean algorithm to be used for clustering in an ad hoc network.

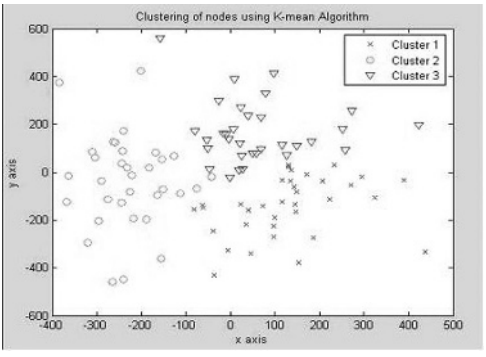


Fig. 6. K-mean clustering which groups the nodes into three clusters

5 Conclusion

In this paper, we have proposed a policy-based ad hoc network management framework which uses position-based routing for mobile nodes to communicate with one another. Our architecture uses Location-Aided Routing (LAR) as the underlying routing protocol and takes advantage of its direction-based routing mechanism for exchanging packets between Policy Decision Point (PDP) and Policy Enforcement Point (PEP). Positions of both PDPs and PEPs are determined by using GPS. Since the nodes are mobile, we use Extended Kalman Filter (EKF) to estimate their positions instead of the Least Square Method which is used in a static case. We take advantage of EKF's capability to calculate the velocity of a mobile node. The velocity is required by the LAR algorithm for determining the traveled distance of a destination node between two time intervals. We presented a dynamic clustering mechanism using k-mean algorithm which makes the management efficient. Our architecture is the first of its kind that uses position-based routing in a management framework. We proposed a modification in the Common Open Policy Service (COPS) protocol. Furthermore, we introduced a modification in the LAR routing algorithm which makes the working of our management framework more efficient. We also presented simulations for Extended Kalman Filter which prove the suitability of EKF for estimating position of a mobile node. The simulation results for the k-mean algorithm exhibit its immense power in making clusters, hence providing a way for the nodes to use dynamic clustering for efficient management.

References

1. Bischoff, R., Wattenhofer, R.: Analyzing Connectivity-Based Multi-Hop Ad hoc Positioning. PerCom (2004)
2. Chadha, R., Cheng, H., Cheng, Y., Chiang, J., Ghetie, A., Levin, G., Tanna, H.: Policy-Based Mobile Ad Hoc Network Management. POLICY (2004)

3. Capkun, S., Hamdi, M., Hubaux, J.P.: GPS-free Positioning in Mobile Ad hoc Networks. Proc. of Hawaii International Conference on System Sciences (2001)
4. Chen, W., Jain, N., Singh, S.: ANMP: Ad hoc Network Management Protocol. IEEE Journal on Selected Areas in Communications (1999) 1506-1531
5. Durham, D. et al.: The COPS (Common Open Policy Service) Protocol. IETF RFC 2748 (2000)
6. Grewal, M.S., et al.: Global Positioning Systems, Inertial Navigation and Integration. John Wiley and Sons, Inc. (2001)
7. Ko, Y-B., Vaidya, N.H.: Location-Aided Routing (LAR) in Mobile Ad hoc Networks. Wireless Networks (2000)
8. Niculescu, D., Nath, B.: Ad hoc Positioning System (APS). Proc. of IEEE Global Communications (GLOBECOM) (2001)
9. Niculescu, D., Nath, B.: DV based Positioning in Ad hoc Networks. Journal of Telecommunication Systems (2003)
10. Phanse, K., DaSilva, L.: Protocol Support for Policy-Based Management of Mobile Ad Hoc Networks. NOMS (2004)
11. Phanse, K., DaSilva, L.: Addressing the Requirements of QoS Management in Wireless Ad hoc Networks. Computer Comm., Vol. 26, no. 12 (2003) 1263-1273
12. Shen, C., Srisathapornphat, C., Jaikaeo, C.: An Adaptive Management Architecture for Ad hoc Networks. IEEE Comm. Magazine, Vol. 41, No. 2 (2003) 108-115
13. YUMA Almanac: http://www.navcen.uscg.gov/ftp/GPS/almanacs/yuma/
14. K-mean Tutorials: http://people.revoledu.com/kardi/tutorial/kMean/index.html

Appendix: Dynamic Position Estimation Using Extended Kalman Filter

For our case, we use the dynamic model to locate the position of nodes as well as velocity using the Extended Kalman Filter (EKF) method. We need at least 4 satellites to estimate the position of a receiver. The pseudo measurement equation defined for GPS for i^{th} satellite is as follows [6]:

$$\rho = \sqrt{(X^i - x)^2 + (Y^i - y)^2 + (Z^i - z)^2} + c\Delta t_r + c\Delta t_{sv} + c\Delta t_{ion} + c\Delta t_{tropo} + \eta \quad \text{(A-1)}$$

Where, $c\Delta t_r, c\Delta t_{sv}, c\Delta t_{ion} and\ c\Delta t_{tropo}$ are range corrections due to receiver clock offset error, satellite clock offset error, ionospheric error, and tropospheric error respectively. $\left[X^i\ Y^i\ Z^i \right]^T$ is Earth Centered Earth Fixed (ECEF)co-ordinate for the i^{th} satellite and η are other errors. Satellite clock offset error, ionospheric error and tropospheric errors can be computed by the correction terms sent by the satellite and in the receiver. Hence these computed corrections get dissolved in the pseudo-range. The only term needed is receiver clock offset error. For the derivation of EKF, we need state-space model and measurement equation. State equation is given as:

$$\bar{x}_{k+1} = \phi_k \bar{x}_k + w_k \quad \text{(A-2)}$$

For kinematic model, state transition matrix with constant velocity model and state vector are defined as:

$$\Phi_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \bar{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \\ z_k \\ \dot{z}_k \\ (c\Delta t_r)_k \end{bmatrix} \tag{A-3}$$

where Δt is sampling period. Here, for kinematic model, we have also considered the velocity of the moving node. Measurement equation is given as:

$$\rho_k = H_k(\bar{x}_k) + v_k \tag{A-4}$$

Since $H_k(\bar{x}_k)$ is nonlinear, so first step is to linearize it. After that, we define the pseudo-measurement equation.

$$\rho_k = H_k(\bar{x}_{k|k-1}) + J_k(\bar{x}_{k|k-1})(\bar{x}_k - \bar{x}_{k|k-1}) + v_k \tag{A-5}$$

$$y_k \equiv \rho_k - H_k(\bar{x}_{k|k-1}) + J_k(\bar{x}_{k|k-1})\bar{x}_{k|k-1} \tag{A-6}$$

Hence we obtain linearized pseudo-measurement equation as:

$$y_k = J_k(\bar{x}_{k|k-1})\bar{x}_k + v_k \tag{A-7}$$

The measurement matrix is defined as:

$$J_k(\bar{x}) = \begin{bmatrix} \frac{\partial \rho_k^1}{\partial x} & 0 & \frac{\partial \rho_k^1}{\partial y} & 0 & \frac{\partial \rho_k^1}{\partial z} & 0 & 1 \\ \frac{\partial \rho_k^2}{\partial x} & 0 & \frac{\partial \rho_k^2}{\partial y} & 0 & \frac{\partial \rho_k^2}{\partial z} & 0 & 1 \\ \frac{\partial \rho_k^3}{\partial x} & 0 & \frac{\partial \rho_k^3}{\partial y} & 0 & \frac{\partial \rho_k^3}{\partial z} & 0 & 1 \\ \frac{\partial \rho_k^4}{\partial x} & 0 & \frac{\partial \rho_k^4}{\partial y} & 0 & \frac{\partial \rho_k^4}{\partial z} & 0 & 1 \end{bmatrix} \tag{A-8}$$

To estimate state $\bar{x}$, we can formulate EKF based on dynamic models (A-2) and (A-7) as:

Initialize, $P_{0|-1}$, $\bar{x}_{0|-1}$.

Measurement Update (Filtering)

$$K_k = P_{k|k-1}J_k^T[J_k P_{k|k-1}J_k^T + R_k]^{-1} \tag{A-9}$$

$$\bar{x}_{k|k} = \bar{x}_{k|k-1} + K_k[\mathrm{y}_k - J_k \cdot \bar{x}_{k|k-1}] \tag{A-10}$$

$$P_{k|k} = (I_N - K_k J_k)\mathrm{P}_{k|k-1} \tag{A-11}$$

Time Update (Prediction)

$$P_{k+1|k} = \Phi_k P_{k|k}\Phi_k^T + Q_k \tag{A-12}$$

$$\bar{x}_{k+1|k} = \Phi_k \bar{x}_{k|k} \tag{A-13}$$

Effects of Storage Architecture on Performance of Sensor Network Queries

Kyungseo Park and Ramez Elmasri

Computer Science and Engineering,
The University of Texas at Arlington,
Arlington, TX 76019, USA
{kpark, elmasri}@cse.uta.edu

Abstract. Storage architecture in sensor networks is increasingly emphasized as an important characteristic, in addition to more traditional characteristics like routing protocols and data dissemination techniques. In this paper, we evaluate several types of storage in order to determine performance correlations between storage types and query types. We first classify the various types of query and storage architectures for sensor networks. We then evaluate storage architecture performance based on types of query. The evaluation metrics we use are the number of transmissions, energy, and end-to-end delay. Data delivery types and routing schemes have to also be considered since they are strongly related to the storage architecture. Based on the performance evaluations, we show what kind of storage is suitable for particular query characteristics.

1 Introduction

The ultimate goal for sensor networks is to query information that was collected from sensors. To achieve this goal, we need to solve several intrinsic problems, some of which are related to queries. One criterion for classifying query types is based on time. These categories are historical queries, snapshot queries, and long-running queries according to [1]. Besides the time criterion, we need to have more criteria to better characterize each type of query. Several papers [1,5,6] have discussed query types for sensor network data systems. These papers sometimes use different terms to represent similar query concepts, and also they have their own ways to categorize queries. In this paper, we characterize several basic elements of queries so that we can analyze the relationships among them.

There are different architectures (for example, routing, storage, and data dissemination) needed to maximize the overall performance in sensor networks depending on the kind of query types we use. Among these architectures, one of the new emerging areas is storage architecture for sensor networks. In general, sensor network storage can be classified into local storage, external storage, and data-centric storage depending on where or how to store the sensed data [7]. Each type of storage scheme has certain advantages based on the particular

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 247–256, 2006.

type of query. One goal of this paper is to specify the relationships between storage types and the query types that each storage supports best.

In order to store data, we need to have not only actual storage, but also a scheme for forwarding the data to the storage from the sensing sources or to a sink from the storage. That means, storage cannot exist by itself without being supported by data dissemination and routing protocols. Models for data delivery required by sensor network applications are classified into continuous, event-driven, and observer-initiated [9]. Continuous data delivery is useful for periodic monitoring systems, which have to send data periodically. Event-driven delivery considers that an event triggers an action, which is usually sensing or forwarding data to a sink. Observer-initiated delivery means that sources do sensing, gathering, or forwarding data whenever a user sends a query.

Besides these classifications, we need to classify architectures for routing protocols so that we can merge data-delivery type, storage type, and query type in one system. Routing protocols can be classified into hierarchical, data-centric, and geographical. Hierarchical routing is a protocol that has more than one structured level, and a typical example of this is a clustering structure in which sensor nodes are divided into clusters and each cluster has its own cluster head and members. Data-centric routing is a protocol that can handle data that is named by attribute-value pairs [4]. Geographical routing uses physical sensor locations in its protocol. However, in this paper, we consider it to be a protocol whose structure is neither hierarchical nor data-centric.[1] We analyze the various combinations of storage, data delivery, and query types to determine which combinations perform best together in sensor networks. After we evaluate and compare each combination, we determine what kind of combination is suitable for each query characteristics. We take into account the number of transmissions, energy, and the delay for packets from a source to a sink.

2 Classifications for Sensor Query Types, Data Dissemination Schemes, Storage Types, and Routing Schemes

Sensor network queries can be classified into four different groups [1,5,6]. The criteria we use for classification are time, aggregation, filter, and dimension. This classification is summarized in Table 1.

There are several schemes for data delivery in sensor networks. We can classify the schemes based on when or how the data should be delivered to the sink. The top class is classified into continuous, event-driven, and observer-initiated in terms of the data delivery required by the application [9].

- Continuous: data is generated and forwarded to a sink continuously according to the task that is already embedded in the sensor network or disseminated into the network initially when the network is initialized.

[1] Some hierarchical and data-centric protocols also utilize sensor locations.

Table 1. Query classification and different terms in several papers

criteria	classification	description
time	one-shot	a query is posed once and data is returned only for the particular moment
	continuous	continues from a particular moment to logically infinite time
	limited-range	continuous query that has a particular end time
aggregation	spatial-aggregate	aggregate data from several sensors that are in a certain region
	temporal-aggregate	aggregate data for a certain period at one sensor
	non-aggregate	if data do not need aggregation
filter	filtering	data returned only if it is within a specified range
	non-filtering	if data do not need filtering
dimension	one-dimensional	a query requests only one type of data
	multi-dimensional	a query requests two or more types of data

- Event-driven: data is generated or triggered by an event and is forwarded to a sink.
- Observer-initiated: data is gathered based on the specified queries, which are disseminated into the sensor network when submitted by a user.

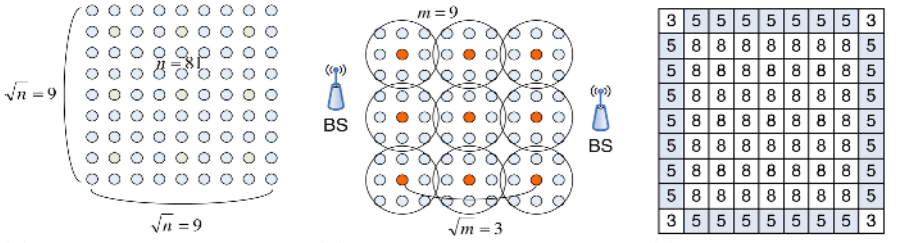
Storage types are generally classified into three categories: local storage, external storage, and data-centric storage [7]. Local storage means that sensor nodes store what they sensed on their own storage space. Usually the storage is a small amount of memory equipped on the sensor node. External storage is that all the data sensed from sensors is sent to an external storage and saved generally on a fixed hard drive attached to a computer. Data-centric storage requires that data are assigned names based on the type of data produced by the sensors, and stored at a particular node in a predetermined way, for example, by using geographic hashing function [7]. Determining where to store each data is based on the 'name' rather than specified by a node address.

So far, many data routing schemes have been proposed. We classify them into several categories that have intrinsic properties. Those are geographical, hierarchical, and data-centric routing schemes. [2]

- In geographical routing, nodes know the spatial position of their neighbors.
- Hierarchical routing provides a structured level on the network topology such as clustering or grid-based virtual topology [11].
- Data-centric routing uses named-data to route a packet unlike the traditional routing scheme that uses IP address. Typically, the type of query and data includes attribute-value pairs [4].

We now discuss the routing schemes that our analysis in section 3 is based on. In geographical routing, we assume a simple topology that has n homogeneous sensor nodes evenly distributed in a square area as shown in Fig.1(a) All the nodes have abilities to sense phenomena, relay packets, and aggregate data from neighbors. Among them, there are source nodes that send sensed data and a sink node that receives the data and connects to a base station, which is outside the

[2] Data-centric routing and data-centric storage refer to two different techniques.



(a) Basic geographical routing topology (b) Hierarchical routing topology (c) Number of neighbors

Fig. 1. Sample topologies for analysis. The n is total number of nodes, and the m is total number of cluster heads.

sensor network. In hierarchical routing, we assume a simple model that looks like LEACH (Low-Energy Adaptive Clustering Hierarchy [3]), but has an ability for cluster heads to communicate with neighboring ones in a multi-hop way, and can finally forward packets to the base station. Basic sensor nodes deployment is the same as in the geographical routing topology (see Fig.1(b)).

We now give some definitions of the terms that are used in the remainder of this paper.

- A *task* determines a job that the sensor network has to do. For example, a task could be to measure temperature or humidity or both. It can be embedded on the network before the sensors are deployed or it can be forwarded by a user when necessary. It is usually a long-running query, which is periodically continuous.
- *Raw data* is collected from sensors without being modified, such as pure measurement from a sensor.
- A *query* is a question or an interest to get a result from the sensor database. It is distinguished from task since it is more specific than task.
- An *Answer* is a response to a specific query. It is also called result in the general database area.

3 Cost Analysis for Three Metrics

In this section, we evaluate four sample queries in terms of three metrics based on the combinations of storage type and routing schemes. We analyze the worst case performance in this section and the average case is analyzed in section 4 via simulation. The following are the three metrics that we consider:

- Number of transmissions: we count the number of transmissions in the network when a query is issued from a sink until the sink gets an answer from the sources.
- Energy: we model the amount of energy dissipated in the network after a query is issued from a sink until the sink gets an answer from the sources.

Total amount of energy dissipated in the network and energy dissipated in a particular hot-spot node are considered.
- Delay: we model the time taken after a query is issued until the sink gets an answer.

From section 2, we have 36 combinations of queries by considering the criteria time, aggregation, filter, and dimension. We exclude 8 queries that contradict each other. Among the remaining 28 combinations, we pick four representative cases that include every characteristic at least once.[3]

- Case 1: one-shot, non-aggregate, non-filtering, and one-dimensional. An example is "give me the value of current temperature at sensor #1".
- Case 2: limited-range, spatial-aggregate, non-filtering, and one-dimensional. An example is "give me the value of average temperature at sensor #1 through #9 every 1 minute for the next one hour"
- Case 3: limited-range, temporal-aggregate, filtering, and one-dimensional. An example is "give me the value of average temperature of all the values sensed at sensor #1 every 1 minute for next one hour if it is greater than 70."
- Case 4: continuous, spatial-aggregate, filtering, and multi-dimensional. An example is "give me the values of average temperature if it is greater than 70 and average humidity if it is between 20 and 30 at sensors #1 through #9 every 1 minute from now on."

The sample network that is used in cost analysis satisfies the following conditions:

- All the sensors are static.
- Transmission range includes all the one-hop nodes as neighbors.
- Sources can be any sensor nodes in the network and a sink can be any node either at one of the four corners or at one of the four sides.
- The link between two nodes is robust, so there is no retransmission.[4]
- Sensor nodes have enough storage capacity so as to hold what they sense without loss.

For the cost analysis of the three metrics, we use n sensor nodes evenly distributed in the square grid networks for geographical routing scheme, which forms $\sqrt{n}$ by $\sqrt{n}$ square as shown in Fig. **??**. A transmission range of a node can cover only one-hop neighbor. Fig. **??** shows the number of neighbors at each node in the geographical routing network. A base station that connects the sensor network to the outer network is reachable by one hop from the sink. For the hierarchical routing scheme, the topology model is basically the same as the geographical one, and every cluster head has eight cluster members as shown in Fig. **??**. A transmission range for a cluster head is to cover its one-hop neighbor cluster heads (up to eight) so that they can communicate with each other. The basic algorithm for routing forwards a packet to a node that is closest to the destination among its neighbors.

[3] Because of space limitations, a complete analysis of all query types is not feasible in this paper.

[4] This is possible since we compare several architectures horizontally.

Table 2. The maximum number of transmissions for query case 1. The first $\sqrt{n}-1$ is for task, if any. The 'x' represents not applicable.

routing	Geographical			Hierarchical	
storage	LS	ES	DCS	LS	ES
Q	$\sqrt{n}-1$	$\sqrt{n}-1$	$\sqrt{n}-1$, $\sqrt{n}-1$	$\sqrt{m}$	$\sqrt{m}$
S	x	$\sqrt{n}$	$\sqrt{n}$	x	$\sqrt{m}+1$
A	x	x	x	x	x
R	$\sqrt{n}$	x	$\sqrt{n}$	$\sqrt{m}+1$	x

3.1 The Number of Transmissions

We consider two routing schemes, geographical and hierarchical for the number of transmissions. All three storage types are evaluated using geographical routing. Local storage and external storage are both considered in hierarchical routing since data-centric storage is not suitable for hierarchical clustering scheme.

For each query, we have four steps to determine the number of transmissions in detail: sending query (or task), storing sensed data, aggregating data, and sending answers with respect to the three types of storage. The basic idea to count the number of transmissions is that we have $\sqrt{n}$-1 hops at most since one of the longest paths is from lower-left corner to upper-right corner and one packet is responsible for a query and data. Based on that, the number of transmissions is analyzed step by step and query by query. Table. 2 shows the number of transmissions in query case 1 step by step.[5] 'Q' stands for sending query (or task), 'S' for storing sensed data, 'A' for aggregating data, and 'R' for sending answers. 'LS', 'ES', and 'DCS' stand for local storage, external storage, and data-centric storage, respectively, and m is the number of cluster heads. The total maximum number of transmissions for one storage type is equal to the sum of all the rows in the corresponding column.

3.2 Energy

[8] suggests the total amount of energy consumed by the network for each transmitted packet as the summation of energy consumed for transmitting, receiving, and reading only the packet header. Based on this, we have our own model for energy consumption in unicast transmission model. The model does not consider energy for processing:

$$E_u = e_{tx} + e_{rx} + (m_i - 1)e_{oh} \tag{1}$$

where E_u is the total amount of energy consumed for transmitting one packet, e_{tx} is the amount of energy consumed by a sender for transmitting one packet, e_{rx} is for receiving one packet, e_{oh} is for overhearing any packet that is not destined to the node, and m_i is the number of neighbors that are within radio range of the i^{th} node.

Based on this basic model, we can extend it to the amount of energy consumed by peer-to-peer unicast way. We can derive the maximum amount of energy spent by transmitting a packet from a source to a sink based on the equation (1).

[5] Because of space limitations, we show the analysis only for query case 1.

Table 3. Energy dissipated in a hot-spot node in case 1. c_{tx}, c_{rx}, and c_{oh} stand for energy consumed by cluster heads in transmitting, receiving, and overhearing a packet respectively. The 'x' represents not applicable.

routing	Geographical					Hierarchical		
storage	LS		ES	DCS		LS		ES
node	sink	aggN	sink	sink	storage	sink	aggN	sink
Q	$e_{tx}+e_{oh}$	x	$e_{tx}+e_{oh}$	$e_{tx}+e_{oh}$,$e_{tx}+e_{oh}$	e_{rx}	$c_{tx}+c_{oh}$	$c_{tx}+c_{rx}$	$c_{tx}+c_{oh}$
S	x	x	e_{rx}	x	e_{rx}	x	x	c_{rx}
A	x	x	x	x	x	x	x	x
R	e_{rx}	x	x	e_{rx}	$e_{tx}+e_{oh}$	c_{rx}	$c_{tx}+c_{rx}+c_{oh}$	x

$$E_{ss} = \sum_{i=1}^{\sqrt{n}} E_u = \sqrt{n}(e_{tx}+e_{rx}) + (7\sqrt{n}-6)e_{oh} \tag{2}$$

When we consider the total energy dissipated by all the sensors in the network, the results are proportional to the total number of transmissions that are analyzed in section 3.1 since the analysis considers end-to-end transmission. Therefore, we only focus on the amount of energy dissipated in a hot-spot node, which can be a sink, an aggregate node, a node for data-centric storage or any other node that uses the most energy depending on the situation. This is needed since the network is not available if a hot-spot node with an important role dies. Table 3 shows the total energy dissipated at a candidate hot-spot node for query case 1. An 'aggN' means aggregate node in Table 3.

We adopt approximate unitless ratio of 5:3:1 to give values for the energy used for transmitting, receiving, and overhearing from [2]. Also, we adopt channel path loss model from [10],

$$\rho = a\delta^{\gamma} + b \tag{3}$$

where ρ is transmission power, δ is a distance between sending node and receiving node, γ is power loss constant ($2 < \gamma < 4$), and a and b are constants. We choose $\gamma = 3$, $a = 1$, and $b = 0$ to simplify the model.

3.3 End-to-End Delay

In the case of reducing energy consumption in sensor networks by decreasing the transmission range, there is a trade-off between energy and delay. Shorter transmission range saves energy consumption at transmitting nodes, but it will increase the number of hops for packet delivery, thus increasing end-to-end delay. Therefore, we need to consider end-to-end delay with energy consumption. We assume the end-to-end delay is proportional to the number of hops from a source to a sink. Since we consider only the worst case, the number of hops from a source to a sink can be bounded by linear combination of $\sqrt{n}$ or $\sqrt{m}$, which are the number of hops from the farthest source (or, the farthest source among cluster heads) to a sink.

4 Performance Evaluation

In this section, we evaluate our three metrics by simulation to measure average cases. We assume robust links, hence there is no retransmission due to channel errors. Nodes are static and distributed evenly on the square grid. The number of nodes is 9801, whose square root is 99, which is the number of nodes in one side of the topology. Nodes in geographical routing are homogeneous but ones in hierarchical routing are not homogeneous. A cluster head has transmission range that can cover its eight cluster head neighbors, but a cluster member can only cover its one-hop neighbors including its own cluster head. The simulation is based on the cost analysis in section 3. Discrete time event driven simulator is used to simulate all combinations of four query cases and five types of different routing and storage.

For geographical routing, a sink node is fixed at the upper-right most node, and a source is generated randomly. In our simulation, an aggregate query such as query case 2 and 3 aggregates data from one-hop neighbors of the source, which are generally 9 nodes including the source node itself. For a hierarchical routing, a sink node is fixed at upper-right most cluster head, and a source can be any nodes regardless of being a cluster head or a cluster member. But for spatial aggregate query such as in query case 2 and 4, a query is restricted to have 9 nodes to be aggregated.

The energy measurements in Table 4 represents a hot-spot node in the network, which can be a sink, aggregation, or storage node. Delay measures the time from when a query packet is sent into the network until an answer packet is received at the sink node. The simulation ends when the last packet is sent back to the sink or when any node runs out of energy.

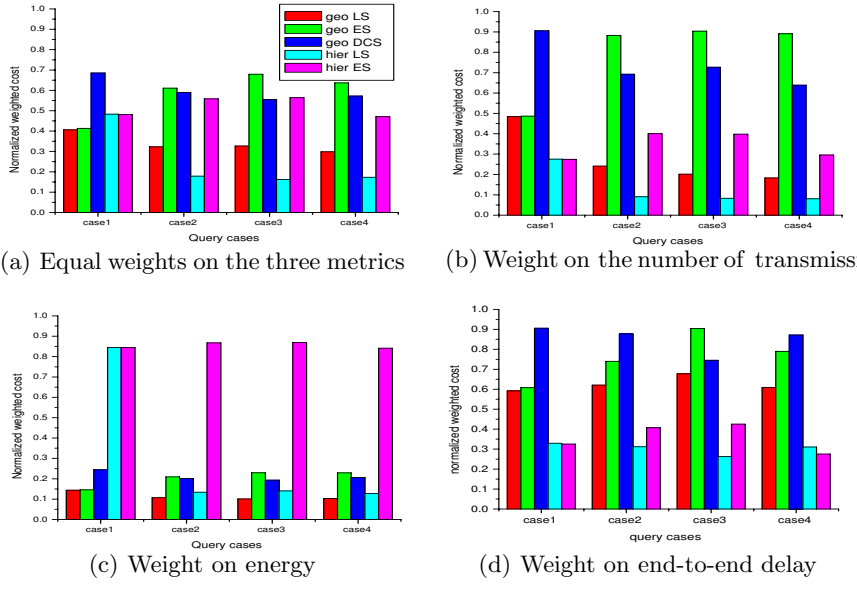
Table 4 shows the simulation results for the three metrics on the four query cases and five storage types. Fig. 2 shows the comparative analysis with different weights for the number of transmissions, energy, and delay on the average. For the equal weights on the three metrics (Fig. 2(a)), hierarchical local storage is most efficient overall. Hierarchical local storage has the overall lowest number of transmissions and the overall lowest delay since it uses longer transmission range that shorten the number of hops for forwarding packets, which brings the overall best cost to the hierarchical local storage in Fig. 2(a). But hierarchical storage is not efficient for one-shot and non-aggregate query such as query case 1, since the geographical storage outperforms it due to having smaller number of neighbors.

Geographical local storage consumes less energy than the other schemes because of combination of factors: energy consumption in transmitting, receiving,

Table 4. Results for the simulation (shows unitless values)

metrics	Number of transmissions					Energy					Delay				
routing	Geo			Hier		Geo			Hier		Geo			Hier	
storage	LS	ES	DCS	LS	ES	LS	ES	DCS	LS	ES	LS	ES	DCS	LS	ES
case1	116.4	116.4	224.9	41.8	41.8	0.1	0.1	0.2	3.6	3.6	1.2	1.3	1.8	0.5	0.5
case2	1185.8	5761.8	4242.6	303.2	1919.2	3.2	8.2	7.5	25.1	219.5	1.4	1.5	1.9	0.7	0.7
case3	94.5	640.2	512.2	31.6	209.0	0.1	0.9	1.0	3.3	24.9	1.1	1.4	1.1	0.4	0.5
case4	1467.3	10999.8	7337.2	450.9	2425.5	5.8	16.3	14.8	32.4	300.3	1.4	1.6	1.9	0.7	0.4

and overhearing, and several candidate hot-spot nodes. External storage for both geographical and hierarchical routing is not suitable for limited range or continuous query cases since they have to send data periodically without having any aggregation or filtering, which increases the number of transmissions and energy consumption. Geographical data centric storage has the worst delay, since it needs extra hops to save sensed data and retrieve them.



(a) Equal weights on the three metrics

(b) Weight on the number of transmission

(c) Weight on energy

(d) Weight on end-to-end delay

Fig. 2. Performance cost comparison for each query case and storage type with different weight for the three metrics based on simulation result

5 Conclusions and Future Work

In this paper, we evaluated the performance of four representative types of sensor network queries when different sensor storage architectures are deployed. Routing and data dissemination schemes were also considered in the simulation. Our results show that local storage outperforms the other schemes.

We plan to extend our measurements to consider additional metrics, such as local storage limitations when compared to the large capacity available in external storage. Additional types of queries will be considered, as well as different network topologies.

References

1. P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *IEEE Personal Communications*, 7(5):10–15, October 2000.
2. S. Coleri, A. Puri, and P. Varaiya. Power efficient system for sensor networks. In *ISCC*, pages 837–842, 2003.

3. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.
4. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MOBICOM*, pages 56–67, 2000.
5. X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-dimensional range queries in sensor networks. In *Proc. of the 1st international conference on Embedded networked sensor systems, SenSys '03*, pages 63–75, 2003.
6. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. In *OSDI*, 2002.
7. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: a geographic hash table for data-centric storage. In *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02*, pages 78–87, 2002.
8. K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proc. of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.
9. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless microsensor network models. *SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, 2002.
10. S. Wu and K. S. Candan. Gper: Geographic power efficient routing in sensor networks. In *ICNP*, pages 161–172, 2004.
11. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *MOBICOM*, pages 148–159, 2002.

Mobility-Aware Distributed Topology Control for Mobile Multi-hop Wireless Networks*

Zeeshan Hameed Mir, Deepesh Man Shrestha, Geun-Hee Cho, and Young-Bae Ko

Graduate School of Information and Communication, Ajou University, South Korea
{zhmir, deepesh, khzho, youngko}@ajou.ac.kr

Abstract. In recent years mobile multi-hop wireless networks have received significant attention and one of the major research concerns in this area is topology control. While topology control problem in ad hoc networks is NP-complete, several heuristic and approximation based solutions have been presented. However, few efforts have focused on the issue of topology control with *mobility*. In this paper, we introduce a new topology control scheme in the presence of mobile nodes. The proposed scheme predicts future proximity of neighboring nodes and applies power control such that the network connectivity is maintained while reducing energy consumption. Simulation results show that the optimal power selection based on location prediction gives better performance in terms of energy and connectivity.

1 Introduction

In a dense mobile multi-hop wireless network each node might have several neighboring nodes. A node in such networks has to decide upon the use of appropriate links intending to attain better utilization of available bandwidth i.e., by allowing concurrent transmissions and energy saving. Selection of such a subset of neighbors for establishing links is the main objective of topology control.

Topology control have been addressed by many researchers with a common goal of achieving an optimal transmission range that can satisfy two contrasting requirements of reduced interference and network connectivity simultaneously. Lower transmission power results in the network to be partitioned. On the other hand, a node with higher transmission power often causes interference and affects overall network capacity and energy [1]. Moreover, the topology control in presence of mobile nodes is a non-trivial problem. Most of the previous work in topology control have not considered mobility, but used a graph model for network analysis [2].

* This work was supported by the Ubiquitous Autonomic Computing and Network Project, the 21^{st} Century Frontier R&D Program, and the ITRC (Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment), the Ministry of Information and Communication in Korea. Also, it was supported in part by grant No. R01-2006-000-10556-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 257–266, 2006.

Our work is motivated by the following two reasons: (1) Node mobility causes network topology to change dynamically. In this situation, by exploiting non-random mobility pattern each node can predict the future state of neighborhood topology and thus a set of minimum transmission powers required to reach its each neighbor. (2) Optimal transmission power selection based on estimated minimum power information from one-hop neighbors produces a robust topology. A robust topology is the one which maintains connectivity through multi-hop communication and as a result, overall coverage region of the network.

Mobility prediction is not a new concept and has been used before in mobile ad hoc networks, for example Link Estimation Time (LET) was previously introduced for estimating link duration and to perform re-routing before the route breaks [3]. In our proposed scheme each node predicts future position of its neighbor nodes and based on this information it estimates the optimal transmission power required to reach them. We assume that each mobile node is aware of its location relative to some coordinate system so that it can calculate distances to its neighbors. For that purpose the availability of Global Positioning system (GPS) [4] would be ideal, but for deployment scenarios where the use of global coordinate system is not feasible other mechanisms for node localization such as Relative Positioning System [5] can also be utilized. It is shown using simulations that the proposed algorithm maintains connectivity while conserving significant energy.

The rest of the paper is organized as follows. Section 2 presents related work regarding topology control in ad hoc networks. Section 3 explains the proposed scheme. Section 4 deals with performance evaluation and finally in Section 5 we provide conclusions and future work.

2 Related Work

According to the taxonomy presented in [2], topology control algorithms can be divided into homogeneous and heterogeneous approaches. Among those that are classified considering heterogeneity are based on location, direction and neighbor information.

In localized algorithms, either a central entity computes a set of optimal transmission ranges and assigns them to individual nodes or each node computes the minimum transmission power in a distributed manner. In [6] two centralized algorithms are presented that focuses on preserving connectivity and bi-connectivity in static network. They require global information to compute topology making them unfeasible for mobile scenarios. It also reports on the two heuristic-based distributed schemes namely Link Information No Topology (LINT) and Local Information Link-State Topology (LILT). Both LINT and LILT utilizes node degree information to adaptively adjust a transmission range. LILT also exploits global link-sate information available from a routing protocol. However, the adjustment of transmission power based on node degree is not very effective because higher degree make nodes to lower their transmission range or vise versa which degrades network connectivity and performance.

Authors in [7], proposed a distributed topology control algorithm for multi-hop wireless networks that constructs neighborhood graphs (RNG) based on the direction information. It increases the transmission power based on angle spanned by the neighboring node. The paper claims that this strategy lowers interference, saves energy and provides reliability. However, it does not address the issue of network partitioning due to change in power levels, heterogeneity and mobility.

In [8], each node in the network computes a strongly connected topology based on local neighborhood and channel propagation model information. Most of the previous research efforts [7,8] used constructions from the field of computational geometry with an overall objective of preserving energy-efficient paths. These implications have been disproved by latest findings that emphasize more on generating interference-minimal topology [9] [10].

The work that is closest to ours is presented in [11]. This work extends previous topology control algorithms for static networks (for example, RNG-based) by making them adaptive to mobility. Our scheme predicts the future state of the topology and estimate an optimal transmission power such that the connectivity is maintained, which is in contrast to [11] where larger than actual transmission range is used to preserve the connectivity. Moreover, we construct a robust topology by controlling power on each node based on location information. To the best of our knowledge, previous research efforts have not exploited topology prediction for power control.

3 Mobility-Aware Distributed Topology Control

The proposed scheme is divided into two main phases. First, each node sends HELLO packets with maximum transmission power (P_{max}) to learn the future state of neighborhood topology. HELLO packets comprise node's predicted position and a list of minimum transmission powers that is required to communicate with its one-hop neighbors at some point later in time. Secondly, each node selects an optimal power level ($P_{optimal}$), such that a neighbor demanding higher transmission power can be reached through one that requires lower transmission power level.

The main idea of our scheme is illustrated in Fig. 1. In Fig. 1(a), the initial topology at some arbitrary time (t_0) is depicted. At this time instant HELLO packets are exchanged among the neighbors with maximum transmission power. Fig. 1(b) shows the predicted future position of the nodes at time ($t_{0+\alpha}$). Nodes 5, 4, 3 and 2 are directly reachable from each other. Fig. 1(c) shows that each node adjusts the power required to reach their neighbor such that the connectivity in future is retained. For example, node 2 computes the required power for nodes 5, 4, 3 and 1. Based on this neighbor information it sets up the link with node 3 and 1.

3.1 Topology Prediction and Transmission Power Estimation Phase

At t_0, each node computes a list of minimum transmission powers required to reach its 1-hop neighbors for time $t_{0+\alpha}$. Time instances t_0 and $t_{0+\alpha}$ represents

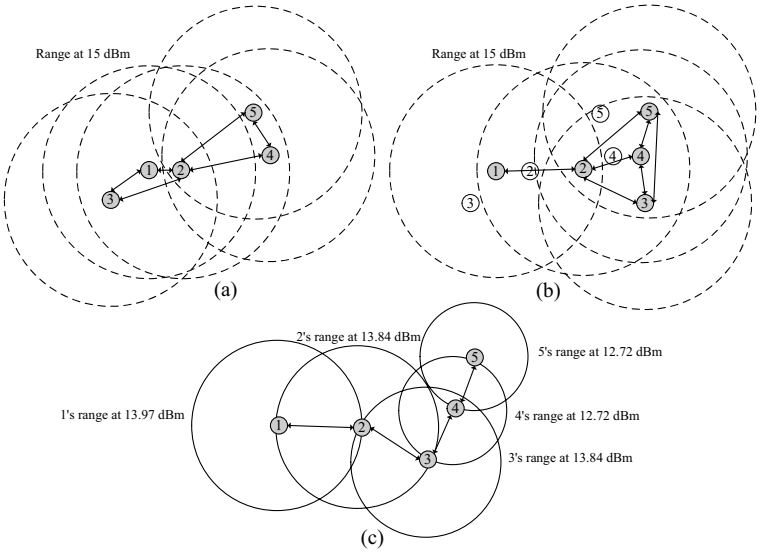


Fig. 1. (a) Initial topology. (b) Predicted topology with maximum transmission range. (c) Topology after transmission power adjustment.

current time and new time, respectively and α is the time increment in seconds. The list is constructed as follows:

A node predicts its own future position given its current position, speed and direction using following two equations, such that:

$$x(t_{0+\alpha}) = x(t_0) \pm s * (t_{0+\alpha} - t_0) * \cos(\theta) \tag{1}$$

$$y(t_{0+\alpha}) = y(t_0) \pm s * (t_{0+\alpha} - t_0) * \sin(\theta) \tag{2}$$

Here ($x(t_{0+\alpha})$, $y(t_{0+\alpha})$) denote the position of a node at $t_{0+\alpha}$, s is the current speed which is bounded by some maximum value, and θ is the direction. Next, future distance to each of its 1-hop neighbors is calculated using Eq. (3). Assume that there are two neighbor nodes, node A and node B, then:

$$d(t_{0+\alpha})_{AB} = \sqrt{(x_A(t_{0+\alpha}) - x_B(t_{0+\alpha}))^2 + (y_A(t_{0+\alpha}) - y_B(t_{0+\alpha}))^2} \tag{3}$$

Finally, we can utilize two-ray ground path loss model to predict the mean signal strength P_r for an arbitrary transmitter-receiver separation distance d [13, 12] based on wireless propagation model given by following equation:

$$P_r(d(t_{0+\alpha})_{AB}) = \frac{P_t * G_t * G_r * (h_t^2 * h_r^2)}{(d(t_{0+\alpha})_{AB})^\eta * L} \tag{4}$$

Using Eq. (4) node A estimates the minimum power required to reach node B, provided that transmission power P_t and the predicted distance $d(t_{0+\alpha})_{AB}$ are known.

3.2 Optimal Power Selection Phase

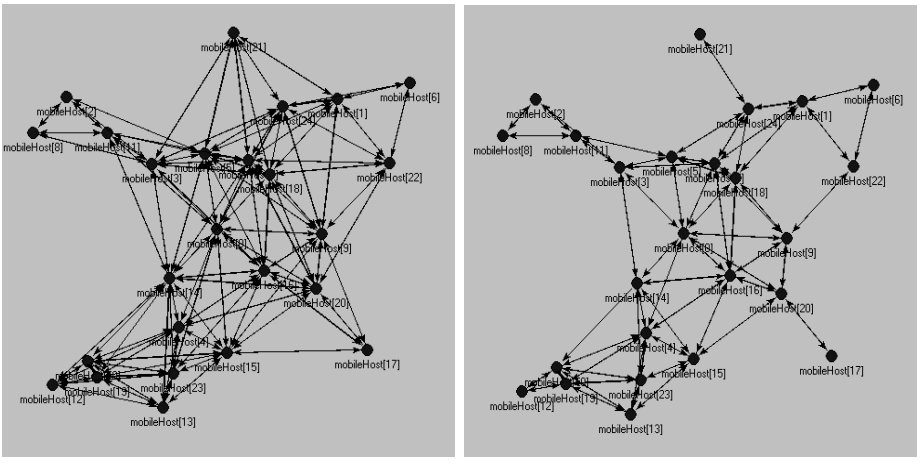
On receiving HELLO packets, each node draws a future topology map in terms of minimum transmission power required to reach its 2-hop neighbors. Each node constructs this topology map by maintaining two data structures. (1) Local view list L consists of two fields, one-hop neighbor's identity and minimum power. (2) Extended view list E includes neighbor's identity, neighbor's-neighbor identity and estimated transmission power.

Using both local view and extended view lists the proposed algorithm selects an optimal power $P_{optimal}$, such that a neighbor requiring higher transmission power can be reached through an intermediate neighbor node. Selection of $P_{optimal}$ is done by comparing transmission power required by node itself and its nearest neighbor to reach the farthest one. If the nearest neighbor does not cover the distant ones, our algorithm searches for another neighbor with relatively higher transmission power, such that the connectivity among all neighbors is retained. Pseudo-code for finding the optimal power is formally given in Algorithm 3.1.

Algorithm 3.1: OPTIMALTXPOWER(L, E)

comment: Node A receives HELLO packet from Node B and $P_{optimal}$ selection.
RECIEVEHELLO(p)
UPDATEEXTENDEDVIEW(p)
UPDATELOCALVIEW(p)
comment: Sort L in descending order of minimum TX power field.
SORTLOCALVIEW($L, txPower$)
$i \leftarrow 0$
$j \leftarrow L.length() - 1$
comment: Initially Optimal power is set to reach farthest neighbor.
$P_{optimal} \leftarrow L_i.txpower$
if $(j \neq i)$
then
  while $(j \geq i)$
  do
    $x \leftarrow L_i.nodeID$
    comment: Is Node x reachable from Node L_j.nodeID?
    if (REACHABLE($x, L_j.nodeID$))
    then
      if $(E_{(L_j,x)}.txpower < E_{(A,x)}.txpower)$
      then $P_{optimal} = E_{(A,L_j)}.txpower$
      else
        $j \leftarrow j - 1$
        continue
    $i \leftarrow i + 1$
return $(P_{optimal})$

Fig. 2 illustrates topologies obtained from full transmission power (no topology control) and proposed optimal power selection algorithm. For this simulation instance we have used two-ray ground reflection model on a network size of 25 nodes uniformly distributed over an area of 512 x 512m^2. Fig. 2(a) depicts the



(a) (b)

Fig. 2. (a) Topology in full transmit power. (b) Topology after optimal transmission power level selection.

topology at maximum power (18dBm) with an approximate transmission range of 170 meters (i.e., without topology control), resulting in an average local neighborhood density of 8.08 neighbors per node. For topology controlled network in Fig. 2(b), the average optimal power is 12.07dBm and the average neighborhood density is 4.88. With lower local neighborhood density and transmission power fewer edges are formed while network connectivity is maintained. Furthermore, it requires less energy to communicate.

4 Performance Evaluation

For performance evaluation we implemented and compared our scheme with pure flooding algorithm (no transmission power control) and LINT protocol [6] using network simulator ns-2 [14].

The flooding algorithm runs with a default transmission power level (i.e., 24.5dBm). LINT on the other hand performs transmission power adjustment based on node degree. If the number of neighbors is less then 6, we run LINT protocol in full power and reduce the transmission power gradually as the node degree increases. The choice of discrete power levels is according to the commercially available CISCO Aironet 350 series wireless LAN card [15]. It has six power levels (24, 21, 18, 13, 7 and -3 dBm) where these power levels corresponds to 250, 210, 170, 130, 90 and 50 meters of transmission range, respectively. Table 1 summarizes the values for all the parameters used in our ns-2 simulations.

Two non-random network mobility models namely deterministic and semi-deterministic mobility models are utilized. (1) In Deterministic mobility model,

Table 1. Parameters and their values in ns-2

Parameters	Meaning	Value
G_t	Transmitter antenna gain	1.0
G_r	Receiver antenna gain	1.0
h_t	Transmitter antenna height	$1.5m$
h_r	Receiver antenna height	$1.5m$
η	Path Loss Exponent	4
L	System Loss Factor	1.0 (i.e., no loss)
α	Time Increment	$10s$

deviation in the movement of the node is set to zero degree. (2) In Semi-deterministic mobility model, deviation is varied from -15 degree to +15 degree, so node movement would vary in 30 degree columnar width. We modified the existing *setdest* program that is included in CMU version of ns-2 [14] for generating our simulation scenarios.

4.1 Simulation Environment

In our simulation model, nodes are randomly placed in a 1000 x $1000m^2$ grid. All nodes move around this region from 1m/s to maximum speed of 20m/s with pause time set to 0. Total simulation duration is 300 seconds. In our experiments we have varied the network size from 50 to 150 in increments of 25 nodes. Following metrics have been used for performance evaluation. (1) *Average Overhead*, is defined as the number of packets received per data transmission per node. (2) *Average Transmit power*, is the ratio of total transmission power and network size. Note that HELLO packets are included only in our scheme and LINT. (3) *Delay*, is given as the time elapsed between data sent from the source node and data received at the intended destination.

4.2 Simulation Results

We begin by examining the overhead of broadcasting data packets in the networks. Fig.3(a) and (b) plots the average overhead per data transmissions, also averaged over total number of nodes as the function of network size for deterministic and semi-deterministic mobility model respectively. Average overhead of the proposed scheme is consistently lower than the basic scheme and LINT. As described in the previous section the effect of applying topology control causes significant reduction in average local neighborhood density, as a result fewer neighboring nodes forwards the data packets. These facts also helps in undermining the advantages of transmitting at full power, which often forwards packets with less number of intermediate nodes. The same benefits bear for LINT, however whenever network turns into undesirable connectivity, nodes switch towards full transmission mode. Since, each node deviates its direction angle in a relatively smaller columnar width the overhead for both mobility models are quite comparable.

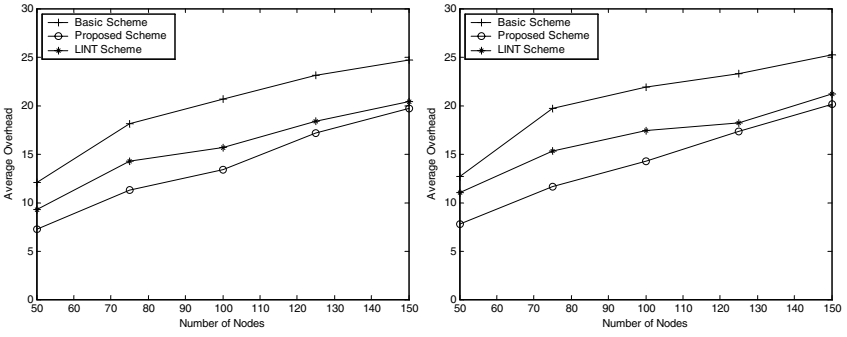


Fig. 3. Results for Average Overhead. (a) Deterministic (b) Semi-Deterministic Mobility Model.

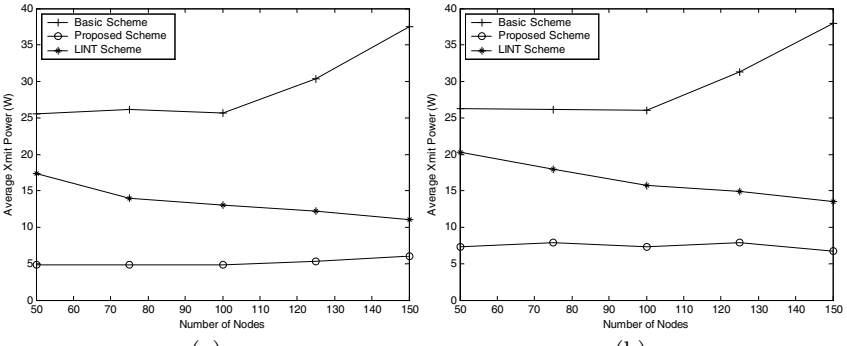


Fig. 4. Results for Average Transmit Power. (a) Deterministic (b) Semi-Deterministic Mobility Model.

Fig.4(a) and (b) presents the average transmits power (in Watts) for both mobility models. These normalized results show that the impact of not regulating the transmission power results in a significantly higher power usage as compared to proposed scheme. Our scheme retains connectivity with nodes that are present at the boundary of transmission range with minimal power whereas LINT enforces to operate at higher transmission power levels in this situation. For smaller network sizes LINT has to increase the power of some isolated nodes. The power consumption for semi-deterministic mobility model is comparatively higher than the deterministic model, which is required in order to compensate limited deviation that neighbor nodes can have.

Fig.5(a) and (b), depicts the average delay (in seconds) in proposed scheme and others. Generally, delay incurred is comparatively higher for all while the network is dense. With moderate traffic load, the delay for non-topology controlled network is lower because it takes small number of hops to disseminate packets in the network as compared to topology controlled one. The delay in both mobility models is similar in all schemes.

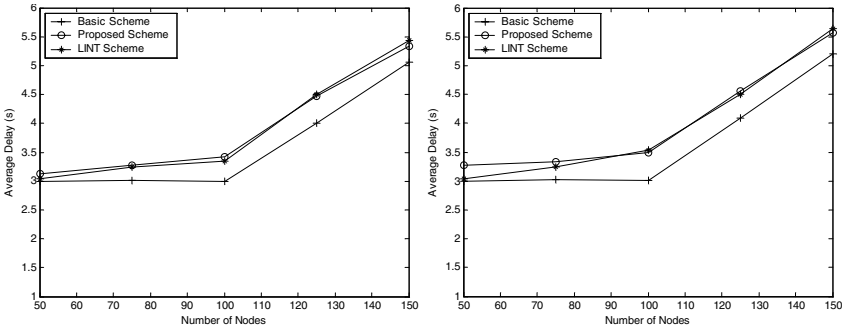


Fig. 5. Results for Average Delay. (a) Deterministic (b) Semi-Deterministic Mobility Model.

5 Conclusion and Future Work

In this paper we have presented a new topology control algorithm in presence of mobile nodes. Our scheme exploits non-random node mobility pattern to predict future state of network topology. Each node runs a distributed algorithm to estimate the minimum power required to successfully communicate with its each neighbor. Finally nodes adjust their transmission power to optimum level to achieve robust topology. Simulation results show that our approach advocates multi-hop communication among nodes for efficient utilization of scarce system resources, such as power and bandwidth in dynamic ad hoc wireless environment. Furthermore, since our protocol operates over a set of discrete power levels, therefore location information is not required to be highly precise. However, another aspect of our algorithm is that it depends upon one-hop HELLO packet, which must be sent periodically with additional neighbor information.

Our future plan is to extend the simulation for different parameters including random mobility model and to apply proposed scheme on link state routing protocols so that the effect of link breakage due to dynamic transmission power level changes can be studied.

References

1. M. Krunz, A. Muqattash, and S-J. Lee, "Transmission Power Control in Wireless Ad Hoc Networks: Challenges, Solutions, and Open Issues", *IEEE Network*, Vol. 18, No. 5, pp. 8-14, 2004.
2. P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks", *ACM Computer Survey*, Vol. 37, No. 2, pp. 164-194, 2005.
3. W. Su, S-J. Lee, and M. Gerla, "Mobility Prediction and Routing in Ad hoc Wireless Networks", *International Journal of Network Management*, Vol. 11, No. 1, pp. 3-30, 2001.
4. B.W. Parkinson et al., "Global Positioning System: Theory and Application", *American Institute of Aeronautics*, Volume 1 and 2, 1996.

5. S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-free Positioning in Mobile Ad hoc Nnetworks", *In Proc. of Hawaii International Conference On System Sciences, (HICSS '01)*, January 2001.
6. R. Ramanathan, and R. R-Hain, "Topology Control for Multihop Wireless Networks using Transmit Power Adjustment," *In Proc. of the IEEE Computer and Communications Societies(INFOCOM '00)*, 2000.
7. S. A. Borbash, and E. H. Jennings, "Distributed Topology Control Algorithm for Multihop Wireless Networks", *In Proc. of World Congress on Computational Intelli-gence (WCCI '02)*, May 2002.
8. V. Rodoplu, and T. H. Meng, "Minimum Energy Mobile Wireless Networks", *IEEE Journal Selected Areas Communication*, Vol. 17, No. 8, pp. 1333-1344, August 1999.
9. M. Burkhart, P. V. Rickenbach, R. Wattenhofer, and A. Zollinger, "Does Topology Control Reduce Interference?", *In Proc. of the 5th ACM International Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc'04)*, May 2004.
10. K. M. Nejad, and X-Y Li, "Low-Interference Topology Control for Wireless Ad Hoc Networks", *Ad Hoc and Sensor Wireless Networks*, Old City Publishing Inc., March 2005.
11. J. Wu, and F. Dai, "Mobility-sensitive Topology Control in Mobile Ad hoc Networks", *In Proc. of Parallel and Distributed Processing Symposium (IPDPS '04)*, April 2004.
12. Y-B. Ko, S-J. Lee, and J-B. Lee, "Ad hoc Routing with Early Unidirectionality Detection and Avoidance", *In Proc. of International Conference on Personal Wireless Communications (PWC '04)*, September 2004.
13. K. Pahlavan, and P. Krishnamurthy, "Principles of Wireless Networks", *Prentice Hall Communications*, 2002.
14. ns-2 network simulator. http://www.isi.edu/nsnam/ns.
15. S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol", *In Proc. of European Wireless (EW '02)*, 2002.

Synchronizing TCP with Block Acknowledgement over Multi-hop Wireless Networks

Changhee Joo[1], Saewoong Bahk[1,⋆], and Hyogon Kim[2]

[1] School of Electrical Engineering and INMC
Seoul National University, Seoul, Korea
{cjoo, sbahk}@netlab.snu.ac.kr
[2] Department of Computer Science and Engineering
Korea University, Seoul, Korea
hyogon@korea.ac.kr

Abstract. While TCP is highly successful in the wire-line Internet, its performance fast degrades as the number of hops increases in multihop wireless networks. It is due to not only the half-duplex nature of the wireless medium, but also the two-way feature of TCP. In order to improve TCP performance in multihop wireless networks, we pay attention to its cumulative ACK policy. Systematically exploiting the redundant nature of the ACK policy, we replace some ACKs with data packets. We call this scheme block acknowledgement. In consequence, the throughput is significantly increased at the slight weakening of ACK reliability. We evaluate the performance of TCP with and without the block acknowledgement through simulations and analysis.

1 Introduction

As wireless networks become prevalent, there is an increasing demand of network connectivity in infrastructureless environments such as emergency situation. TCP is a natural choice as transport layer protocol because of its widespread use in the Internet. However, it has been shown that TCP performs poorly in multihop wireless environment.

TCP provides a reliable delivery service by using acknowledgement (ACK), which is returned by the receiver for each data packet. This two-way feature does not cause a serious problem in wired networks because most of links are full-duplex. In wireless networks, however, links are usually half-duplex. Since the wireless medium allows only a single, one-way transmission at a time, TCP ACKs compete with data packets. As a consequence, they share the bandwidth for data and may even cause collisions with data packets.

⋆ This research was supported partially by the University IT Research Center Project and the Ubiquitous Autonomic Computing and Network Project, Ministry of Information and Communication, in Korea.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 267–275, 2006.

Due to this peculiarity of the wireless medium, it is a common knowledge that TCP fails to achieve its best performance even in the absence of mobility [1]. So there have been substantial efforts to improve TCP performance in multihop wireless networks.

The delayed ACK option, in which the receiver sends an ACK for every other received packet instead of every packet [2], can improve TCP performance in multihop wireless networks [3,4]. By halving the number of ACKs, it decreases the amount of backward traffic on a half-duplex wireless link, thus allowing less collision with data packets in wireless links and reducing the instability in TCP algorithms. For short connection, a large initial window option [5] is recommended to be used together.

ACK thinning is a generalized term for intentional ACK drops such as the delayed ACK option. It is used to boost TCP performance in asymmetric networks [6]. The receiver sends out an ACK for a number of data packets, which is adjusted in a manner of AIMD (Additive Increase and Multiplicative Decrease) based on information conveyed in additional TCP options. Altman *et al* also proposed dynamic delayed ACK, which changes the frequency of ACKs according to sequence number [7].

In this paper, we extend the ACK thinning and propose a systematic solution to reduce the number of ACKs for increased throughput, while controlling the risk of reliability loss.

The rest of the paper is organized as follows. We propose TCP block acknowledgement in section 2, which is a general algorithm reducing the number of ACKs. The performance in a simple chain topology is evaluated through both simulation and analysis in section 3. We conclude our paper in section 4.

2 TCP Block Acknowledgement

Since TCP ACK consumes wireless bandwidth and sometimes collides with TCP data packet, performance can be improved by reducing the frequency of ACKs. In this paper, we group a series of data packets into a block and make the receiver generate a single ACK for the group. This *block acknowledgement* is a kind of ACK thinning. The delayed ACK option is equivalent to the block acknowledgement with the fixed block size of 2.

In this section, we discuss the issues in block acknowledgement. They include the block size, *who* decides the block size, and some optimizing algorithms that maximize the gain from the block acknowledgement.

The block size is a crucial parameter, determining the trade-off between performance and reliability. As the block becomes larger, an ACK acknowledges more data and TCP gains throughput if there is no ACK loss. However, if the loss does happen, it has worse effect on TCP performance with larger block size.

A constraint on the block size is that it should be less than TCP transmission window size. Otherwise, the sender could wait for an ACK after sending out all data in its window, while the receiver waits for more packets before sending out an ACK – a typical deadlock situation. The receiver could use a timer in this

case in order to break the deadlock, but the excessive waiting time until the timeout would significantly degrade the performance.

Since the block size should be smaller than the sender's transmission window, which is highly dynamic depending on congestion situation, it should be the sender that controls the block size[1]. In our scheme, the sender sets the block size to

$$\text{Block size} = min(window/\alpha, MaxBlockSize), \tag{1}$$

where *window* is the sender's transmission window size (minimum of congestion window and the receiver's advertised window), and $MaxBlockSize$ is the upper bound of the block size. The parameter α determines the ACK rate. TCP usually sends out a window worth of packets in an RTT, to which the receiver responds with α ACKs. Thus α/RTT becomes the ACK rate. Finding an optimal α should depend on the dynamic ACK loss rate, and we leave it as a future work. Instead, we fix α to 2 in this paper, making two ACKs return for a window worth of packets.

For our block acknowledgement, the sender and the receiver need an agreement on the block size. Since the sender decides the block size, the receiver should be able to detect the block boundary in order to generate an ACK for a block. We solve this using a single bit in TCP header, which we call *block indicator* in this paper. A single bit from unused offsets in the TCP header is used. The sender sets the bit to 0 or 1, toggling at the start of a block and the receiver sends out an ACK upon the change of the bit. Fig. 1 illustrates an example of packet exchanges with $\alpha = 2$. Solid arrows represent transmissions of data packets, and dotted arrows, ACKs. The block indicator is shown at the tail of an arrow. It is very simple and robust to loss of packet at the block boundary.

There are some unwanted side effects that stem from the block acknowledgement.

- Slow window growth : TCP inflates its window based on the number of received ACKs. If it decreases, window grows slowly, resulting in fairness problem with other TCPs.
- Packet burstiness : Under block acknowledgement, an ACK acknowledges more than a data packet. The sender would put burst packets into the network on an ACK arrival.
- Congestion from duplicate ACKs : Out-of-order packets strongly implies congestion in the network. Duplicate ACKs for all out-of-order packets possibly worsen the congestion in multihop wireless networks because ACKs share the bandwidth with data packets.

For the above problems, the following solutions can be employed.

- Sender adaptation [6] : The sender can avoid a slowdown in window growth by taking into account the amount of data acknowledged by an ACK.

[1] Otherwise, whichever entity decides the block size should be informed or be able to infer the sender's window size.

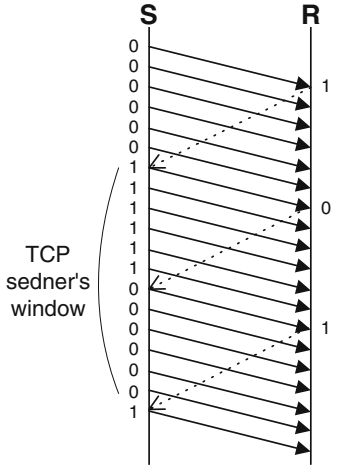


Fig. 1. TCP packet exchanges with block indicator ($\alpha = 2$)

- TCP pacing [6,8] : The sender forces to space data packet evenly over an RTT in order to avoid packet burstiness.
- Removing duplicate ACKs : Instead of sending duplicate ACK for each out-of-order packet, the receiver sends an ACK with an option like SACK (Selective Acknowledgement [9]) upon the third out-of-order packet. The sender immediately retransmits the lost packet upon receiving the ACK.

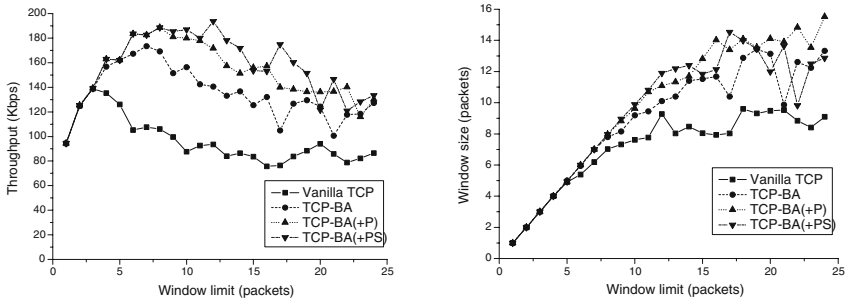
In order to explore the efficacy of these solutions when used with the block acknowledgment, we simulate in the next section TCP block acknowledgement with sender adaptation (TCP-BA), with sender adaptation and pacing (TCP-BA(+P)), and with all of them combined (TCP-BA(+PS)).

3 Performance Evaluation

3.1 Simulation

We simulate a simple chain topology using ns-2 [10]. Neighboring nodes are apart by 200 m in a chain topology, and a TCP connection is established between end nodes without background traffic. We use the standard IEEE 802.11 DCF MAC with 2 Mbps physical rate, 250 m transmission range, and 550 m interference range, *i.e.*, the ratio of interference and transmission range r is about 2. As for routing protocol, AODV is used. The data packet size is fixed to 512 bytes unless otherwise specified.

Throughput and window size of the TCP connection are measured over 8-hop chain topology, while varying the window limit from 1 to 24. The results presented in Fig. 2 show that i) TCP performance degrades if the window size exceeds a certain threshold, and ii) TCP can achieve better performance using the



(a) Throughput (b) Average window size

Fig. 2. Performance of TCP and TCP-BAs with different window limits in the 8-hop chain topology

block acknowledgement, which is more effective with optimization algorithms. Note that pacing with vanilla (conventional) TCP does not result in significant performance improvement in multihop wireless network [11], while it generates a synergistic effect on performance if coupled with the block acknowledgement.

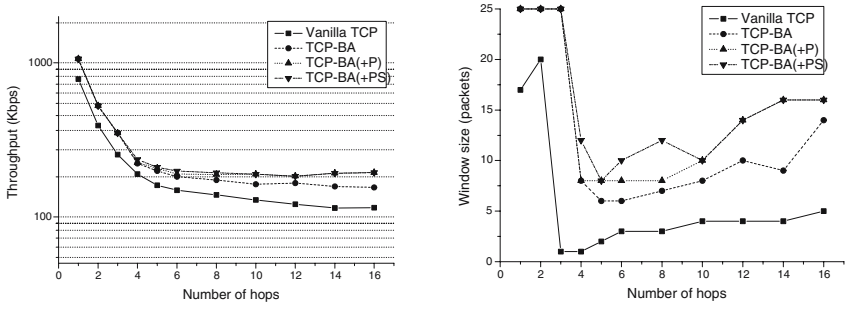
Now, we repeat the simulation with different window limits, varying the number of hops from 2 to 16 in chain topology. Fig. 3 illustrates the maximal throughput and the window limit that maximizes the throughput in each hop. Up to 4 hops, TCP throughput drops dramatically because the network can not make two simultaneous transmissions, *e.g.*, a transmission from node 3 to 4 interferes with transmission from node 0 to 1 because node 1 is placed in the interference range of node 3. In this short connection, the maximal throughput of vanilla TCP does not change much with window limits, but TCP-BAs benefit more with larger window limit because more ACKs are omitted. This is why TCP-BAs have 25 or more window limit for the maximal throughput.

For the 4-hop case, TCP often experiences link failure due to hidden/exposed terminal problem. Since the link failure frequently occurs when a number of packets are outstanding, TCP with smaller window limit achieves more throughput.

In long connection of over 4 hops, multiple transmissions out of interference range are possible. TCP can benefit from larger window size because more spatial reuses are available as the number of hops increases.

3.2 Analysis

The long-term throughput of h-hop TCP connection can be also obtained from simple analysis. When h is less than $r+2$, there can be only a single transmission at a time because the hidden/exposted terminal is located $(r+1)$-hop away. When a packet is sent out from the sender, it takes $h \cdot T^{MAC}_{tcp\_data}$ to arrive at the receiver. Then the ACK from the receiver returns to the sender in another $h \cdot T^{MAC}_{tcp\_ack}$, where $T^{MAC}_{tcp\_data}$ and $T^{MAC}_{tcp\_ack}$ are the times for RTS-CTS-DATA-ACK



(a) Maximal throughput

(b) Throughput maximizing window limit value as a function of hops

Fig. 3. The best performance of TCP and TCP-BAs with different number of hops

exchanges of TCP data packet and TCP ACK over a single hop. Then the required time for a RTS-CTS-DATA-ACK exchange over a single hop is

$$\begin{aligned} T^{MAC} =& T_{DIFS} + T_{contention} + T_{RTS} + T_{SIFS} \\ &+ T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} + T_{ACK}, \end{aligned}$$

where T_{RTS}, T_{CTS}, T_{DATA}, T_{ACK} are respective transmission times of RTS, CTS, MAC DATA, MAC ACK frame, and $T_{contention}$ is the length of the contending period. Assuming no contention under optimal MAC scheduling, we use average contending period of 16 slot time for $T_{contention}$.

If TCP has a window size of W, the sender sends out W packets before receiving an ACK. However, since there can be only a single transmission over the network at a time, vanilla TCP can transfer W packets in $W \cdot h \cdot (T^{MAC}_{tcp\_data} + T^{MAC}_{tcp\_ack})$. Then, vanilla TCP's throughput for $h \leq r + 2$ is obtained as

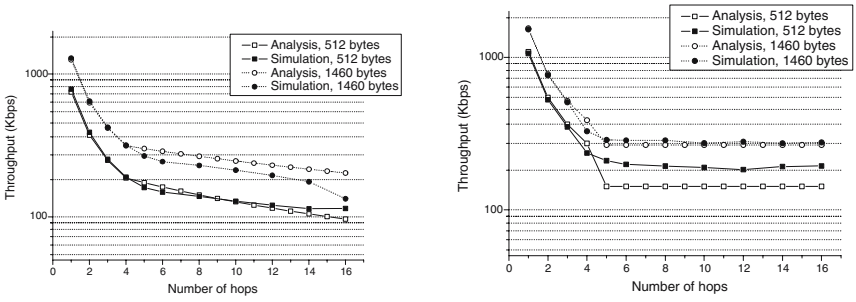
$$Throughput\ of\ vanilla\ TCP = \frac{W \cdot S}{W \cdot h \cdot (T^{MAC}_{tcp\_data} + T^{MAC}_{tcp\_ack})}, \tag{2}$$

where S is the data packet size. It means that the throughput of vanilla TCP is not a function of W for $h \leq r + 2$. Therefore, for vanilla TCP, the smallest window limit leads to the best performance owning to the absence of collision.

In case of TCP-BA, we can apply the same analysis except for the number of ACKs. Since it returns α ACKs for W packets, its throughput for $h \leq r + 2$ becomes

$$Throughput\ of\ TCP - BA = \frac{W \cdot S}{h \cdot (W \cdot T^{MAC}_{tcp\_data} + \alpha \cdot T^{MAC}_{tcp\_ack})}. \tag{3}$$

Unlike vanilla TCP, throughput of TCP-BA depends on W. It can achieve higher throughput with larger window limit. In the usual case of $(W \cdot T^{MAC}_{tcp\_data} \gg \alpha \cdot T^{MAC}_{tcp\_ack})$, throughput is bounded by $\frac{S}{h \cdot T^{MAC}_{tcp_data}}$.



(a) Vanilla TCP (b) TCP-BA

Fig. 4. Analysis and simulation results of vanilla TCP and TCP-BA with different number of hops

For $h > r + 2$, TCP can benefit from spatial reuse: multiple transmissions at the same time over the network. In order to simplify analysis, suppose all W data packets are evenly paced with $(r + 2)$ hops, and making synchronized progress. Also, assume that when there is a returning ACK, all data packets stop moving and wait until the ACK arrives at the sender. Then, a data packet will arrive at the receiver after $h \cdot T^{MAC}_{tcp\_data}$ if there is no returning ACK. However, note that as a data packet enters the network, in front of the data packet there are $W - 1$ preceding (unacknowledged) data packets in the TCP pipe. For vanilla TCP, therefore, the packet will meet $W - 1$ ACKs for the previously unacknowledged data packets in its way to the receiver. Hence, it spends $h \cdot T^{MAC}_{tcp\_data}$ on moving to the receiver and waits $(W-1) \cdot h \cdot T^{MAC}_{tcp\_ack}$ to make way for ACKs. Arriving at the receiver, it will take $h \cdot T^{MAC}_{tcp\_ack}$ for its ACK to return to the sender. Hence, the RTT is $(h \cdot T^{MAC}_{tcp\_data} + W \cdot h \cdot T^{MAC}_{tcp\_ack})$, and the throughput for $h > r + 2$ becomes

$$Throughput\ of\ vanilla\ TCP = \frac{W \cdot S}{h \cdot T^{MAC}_{tcp\_data} + W \cdot h \cdot T^{MAC}_{tcp\_ack}}. \quad (4)$$

Since only one transmission can exist in a $(r + 2)$-hop range in our model, we conjecture that TCP achieves the best performance with $W = \frac{h}{r+2}$ as in [1]. Then we have

$$Throughput\ of\ vanilla\ TCP^* = \frac{S}{(r + 2) \cdot T^{MAC}_{tcp\_data} + h \cdot T^{MAC}_{tcp\_ack}}. \quad (5)$$

Note that the throughput of vanilla TCP decreases with the number of hops.

Using the same analytical method, we can demonstrate that TCP-BA achieves better throughput than vanilla TCP. Moreover, the throughput of long-haul connections is maintained even with the increasing number of hops. Since the

packet[2] meets $(\alpha - 1)$ ACKs whatever the window size, the RTT becomes $(h \cdot T^{MAC}_{tcp\_data} + \alpha \cdot h \cdot T^{MAC}_{tcp\_ack})$. The throughput of TCP-BA for $h > r + 2$ can be expressed as

$$Throughput\ of\ TCP - BA = \frac{W \cdot S}{h \cdot T^{MAC}_{tcp\_data} + \alpha \cdot h \cdot T^{MAC}_{tcp\_ack}}. \quad (6)$$

With the same assumption of $W = \frac{h}{r+2}$, it becomes

$$Throughput\ of\ TCP - BA^* = \frac{S}{(r+2) \cdot T^{MAC}_{tcp\_data} + (r+2) \cdot \alpha \cdot T^{MAC}_{tcp\_ack}}. \quad (7)$$

We compare the analysis with simulation in Fig. 4. The simulation results presents the best throughput among simulation runs with different window limits. The analysis (with $r = 2$) matches well with simulation for $h \leq 4$. For $h > 4$, there is some discrepancy because we assumed $W = \frac{h}{r+2}$ in the analysis. We observe that TCPs in simulation take a little more window size than $\frac{h}{r+2}$ especially for TCP-BA. What is more important here, however, is that in both analysis and simulation TCP-BA maintains its throughput unlike the vanilla TCP. That the throughput is maintained regardless of h, is another strong merit of the block acknowledgement, along with the absolute performance improvement.

4 Conclusion

We optimize TCP performance in multihop wireless networks through block acknowledgement. The block acknowledgement improves TCP performance by reducing the number of ACKs at the manageable risk of reliability, and LECN enables TCP to tune its window size precisely.

The block acknowledgement generalizes the delayed ACK, forcing the receiver to send an ACK for a block of data packets. Since the block size depends on sender's transmission window size, it is the sender that controls the block size. The sender uses a single-bit block indicator to inform the receiver of the boundary of a block. Simulation results corroborates that the block acknowledgement enhances TCP performance and prevents throughput reduction with hops. It achieves further improvement when combined with auxiliary algorithms of sender adaptation, pacing, and removing duplicate ACKs.

References

1. Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, M. Gerla: The Impact of Multihop Wireless Channel on TCP Throughput and Loss, INFOCOM, 2003.
2. R. Braden: Requirements for Internet Hosts - Communication Layers, RFC 1122, October 1989.

[2] In order to measure RTT, we assume that it is the first packet of new block, which triggers an ACK.

3. A. Kherani, R. Shorey: Performance Improvement of TCP with Delayed ACKs in IEEE 802.11 Wireless LANs, WCNC, 2004.
4. S. Xu, T. Saadawi: Performance evaluation of TCP algorithms in multi-hop wireless packet networks, Wireless communications and mobile computing, 2002.
5. M. Allman, S. Floyd, C. Partridge: Increasing TCP's Initial Window, RFC 2414, September 1998.
6. H. Balakrishnan, V. Padmanabhan, R. Katz: The Effects of Asymmetry on TCP Performance, ACM Mobicom, 1997.
7. E. Altman, T. Jimenez: Novel delayed ACK techniques for improving TCP perfomance in multihop wireless networks, PWC, 2003.
8. A. Aggarwal: Understanding the performance of TCP pacing, INFOCOM, 2000.
9. M. Mathis, J. Mahdavi, S. Floyd, A. Romanow: TCP Selective Acknowledgment Options, RFC 2018, October 1996.
10. The UCB/LBNL/VINT Network Simulator (NS), Available at "http://www.isi.edu/nsnam/ns/".
11. K. Chen, Y. Xue, S. Shah, K. Nahrstedt: Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks, Computer Communication, 2004.

A Grid-Based Manycast Scheme for Large Mobile Ad Hoc Networks*

Shiow-Fen Hwang[1], Kun-Hsien Lu[2], and Chyi-Ren Dow[1]

Department of Information Engineering and Computer Science
Feng Chia University, Taichung, Taiwan 40724, R.O.C.
[1] {sfhwang, crdow}@fcu.edu.tw,
[2] p9217988@webmail.fcu.edu.tw

Abstract. Services providing between clients and servers is an important issue in mobile ad hoc networks. Since services become complex and various increasingly, current group communication mechanisms like multicast or anycast do not completely support some kind of services such as NTP (Network Time Protocol) or threshold cryptography, etc. Manycast is a novel group communication paradigm in which a client communicates simultaneously with a number k of m equivalent servers in a group and can support these kinds of services. In this paper, we present a grid-based manycast scheme that employs manycast level calculation algorithm to predict a manycast region and uses an efficient grid broadcast method to support manycast delivery for large ad hoc networks. Simulation results show that our scheme not only reduces significantly more manycast overhead and the average number of replied servers but also keeps high manycast successful rate.

1 Introduction

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes which are self-organization and self-configuration without the aid of any established infrastructure or centralized administration such as the base stations. The mobile nodes can move arbitrarily and the data transmissions can be accomplished via the nearby mobile nodes interchanging messages. The network topology dynamically changes frequently due to node mobility. This kind of networks is especially important and useful in the regions such as battlefields, disaster relief, etc. In recent years, there are many different kinds of services and resources in the network environment such as the multimedia information service, file-sharing service, or print service, etc. However, current group communication mechanisms such as multicast or anycast do not completely support this type demand when a client must contact two or more distributed servers simultaneously.

In [1], C. Carter et al. proposed a novel group communication mechanism called *manycast* and described why it must be implemented in network layer

* This research is supported by the National Science Council of Republic of China under grant number NSC 93-2213-E-035-030.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 276–285, 2006.

to support the service-oriented group communication. Different to multicast, manycast is a group communication mechanism to support a variety of MANET services where a client communicates simultaneously with an arbitrary number k of servers from m servers of a group. Moreover, manycast provides a bidirectional channel to enable request/reply communication between a client and servers, not merely one-way dissemination of data. Fig. 1 illustrates an example of the activity of manycast delivery while a client broadcasts a request packet to entire network, then service providers reply messages to the client.

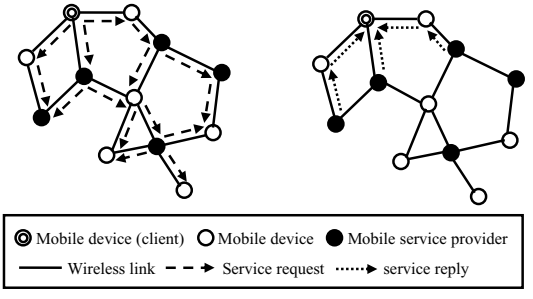


Fig. 1. The activity of manycast delivery

Many actual network applications can communicate in the manycast pattern. For example, in version 4 of Network Time Protocol (NTP) [2], a client needs to find three best/nearest servers with which synchronizes its clock. Other issue such as information security also takes advantage of manycast. The ITTC (Intrusion Tolerance via Threshold Cryptography) project [3] distributes the cryptography key across several servers using threshold cryptography. Each server only possesses a part of the secret key so that a client must contact several servers simultaneously to obtain a complete secret key. Cornell On-line Certificate Authority (COCA) [4] which introduced a distributed certificate authority for wired network is a similar application to the ITTC project. Furthermore, Mobile Certificate Authorities (MOCA) [5] extends COCA to wireless ad hoc networks and performs certification requests and replies by enhancing an ad hoc routing protocol with new message types.

It is a challenge to design a reliable manycast with high performance and low overhead due to the characteristics and limitations of MANETs. C. Carter et al. [1] presented several extensions for ad hoc routing protocols to support manycast delivery and consequently they recommended the scoped-flood scheme because it keeps the benefit of flooding which can resist mobility and has lower transmission overhead compared to other modified schemes. However, scoped-flood scheme still suffers from some weakness including the determination of TTL-limited scoped area that covers at least k servers, server reply implosion, and scalability problem.

First, scoped-flood scheme uses a network-wide broadcast to perform manycast delivery when a client initializes a request packet for the first time to set

TTL value for future usage. However, the network-wide broadcast may cause redundant rebroadcasts, contention, and collision, which are well-known as the broadcast storm problem [6]. A client uses the TTL value obtained from first request to limit the flooding area at the next request. Unfortunately, the evaluated TTL value may become inaccurate due to node mobility. Second, server reply implosion may happen such as the ARP implosion described in [7] when there are too many servers receive a request packet and reply it to a client. Finally, considering the scalability of network environment, when the size of the network extends and the number of nodes and servers increase, the performance of scoped-flood scheme will rapidly drop due to the increase of redundant transmissions.

In order to improve the defects as motioned above, we propose a grid-based manycast scheme that employs manycast level calculation algorithm to predict a manycast region and uses an efficient grid broadcast method to support manycast delivery. The objectives of our scheme are to reduce much more redundant transmissions and the number of replied servers than scoped-flood scheme. Simulation results show that our scheme has better performance and is suitable for large ad hoc networks than scoped-flood scheme.

The rest of this paper is organized as follows. In Section 2, the formation of network environment is described. Section 3 presents a grid-based manycast scheme including server request and server reply phase. Section 4 describes the simulation environment and results. Section 5 is the conclusion.

2 Network Environment

We assume that the network is partitioned into several 2D logical grids. Each grid is a square area of size $d \times d$, denoted by (x, y) as shown in Fig. 2. Each node has a unique ID such as IP address or MAC address and is location-aware by being equipped with a positioning device like a GPS receiver. Therefore, a node can easily map its location information into the grid coordinate. The node which is nearest to the physical center of the grid will be elected as the gateway of the grid. In our network environment, the gateway election and maintenance rule are the same as GRID protocol [8] and each node maintains one-hop neighbor list by periodically exchanging HELLO message.

As mentioned above, each grid is a square of $d \times d$. Let r be the transmission range of a node. We define the value of d as $\frac{\sqrt{2}}{3}r$ to make sure that a gateway can almost communicate with any gateway in its eight neighboring grids. In addition, assume that a client c is located in grid (x_0, y_0). For each grid (x_i, y_i), we define its *relative grid coordinate* and *level* with respect to (x_0, y_0) as $(x_i - x_0, y_i - y_0)$ and $max\{|x_i - x_0|, |y_i - y_0|\}$, respectively. For example, the relative grid coordinate and the level of grid $(0, 2)$ with respect to grid $(2, 1)$ are $(-2, 1)$ and 2, respectively, as shown in Fig. 3 and Fig. 4. The grids with same level can be classified into four types of shapes, O, C, L, I, as shown in Fig. 5. This classification is useful for calculating manycast level value which will be described in next section.

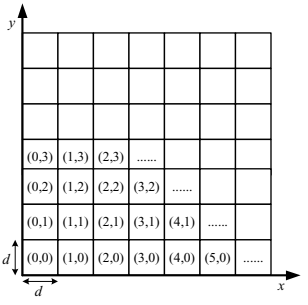


Fig. 2. Logical grids to partition physical network

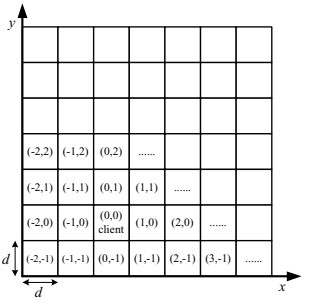


Fig. 3. Relative grid coordinate with respect to grid (2,1) in Fig. 2

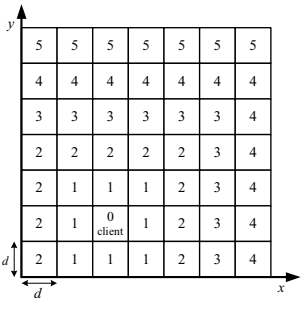


Fig. 4. The level of each grid

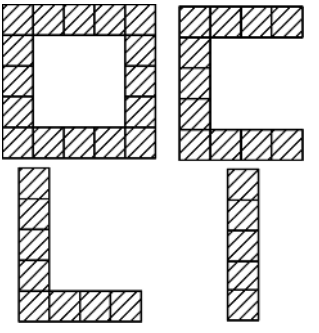


Fig. 5. Four types of shapes, O, C, L, I

3 Grid-Based Manycast Scheme

Our proposed grid-based manycast scheme contains two phases: (1) server request and (2) sever reply. When a client c has to communicate with k servers, then it executes server request phase. First, the client c employs manycast level calculation algorithm to obtain a manycast level value called *McLevel*. The grids that levels are smaller than or equal to *McLevel* form a manycast region. After the client c obtained *McLevel*, it locally broadcasts a manycast request (MREQ) packet to its gateway. The gateways in the manycast region received the MREQ packet perform grid broadcast algorithm to determine whether to rebroadcast it or not. That is to say, not all gateways in the network need to rebroadcast the MREQ packet. In server reply phase, when a server located in the manycast region receives a MREQ packet for the first time, it replies a manycast reply (MREP) packet to the client c by using greedy routing method [9]. In addition, we also attempt to reduce the number of replied servers and avoid too many reply messages by using reply probability P_r. The details of the two phases are described as follows.

3.1 Server Request Phase

In this phase, we assume that there are m number of equivalent servers in the networks. When a client c wants to communicate with a number k of m equivalent servers, it first executes manycast level calculation algorithm (Algorithm 1) which outputs *McLevel* to predict how many levels a MREQ packet needs to reach. In Algorithm 1, *GN* is the predicted number of grids that approximately contains k servers. C_1, C_2 and C_3 are the total number of grids in O-shapes, C-shapes, and L-shapes, respectively.

For example, the network is partitioned into 17×11 grids in Fig. 6. We assume that there are 50 servers in the network and a client c that locates in grid $(5, 2)$ needs to communication with 15 servers. According to our algorithm, *GN*, s, t, h_1, h_2, h_3, h_4, C_1, C_2, and C_3 are 57, 5, 2, 2, 17, 11, 3, 24, 63, and 66, respectively. Because $C_1 + 1 = 25 < 57$ and $C_1 + C_2 + 1 = 24 + 63 + 1 = 88 > 57$, then the client c can find that r and *McLevel* $(= r + h_1)$ are 2 and 4, respectively. Therefore, the grids that levels are smaller than or equal to 4 form a manycast region (indicated by gray area in Fig. 6).

After the client c obtained *McLevel*, it initializes a MREQ packet and locally broadcasts it to its gateway. The MREQ packet consists of the following four fields.

1. *client_id*: the ID of the client
2. *client_grid*: the grid coordinate of the client that initialized this MREQ packet
3. *c_seq*: the sequence number of the manycast request
4. *McLevel*: the manycast level value of the manycast request

When the gateway of the client c receives the MREQ packet, it performs the grid broadcast algorithm as shown in Algorithm 2 to determine whether to rebroadcast the MREQ packet or not. The gateways in the manycast region received the MREQ packet from other gateways also do broadcast determination according to our grid broadcast algorithm. For example, in Fig. 7, black dot represents gateway and oblique line represents packet transmission. There are only 23 gateways of 63 gateways in this manycast region (gray area) need to forward the MREQ packet and other gateways or nodes do not need to rebroadcast the MREQ packet. In general, at most 40% of gateways in a manycast region need to rebroadcast a MREQ packet per manycast request. As Fig. 7 indicates, if each grid has 5 nodes in average, then there are 935 nodes in the networks, but only 23 nodes (gateways) need to rebroadcast MREQ packets. Therefore, our grid broadcast method is suitable for the dense and large network environments and can reduce more redundant rebroadcasts than other flooding-based schemes.

3.2 Server Reply Phase

In server reply phase, when a server located in the manycast region receives the MREQ packet for the first time, it replies a MREP packet to the client c. Note

Algorithm 1. Manycast level calculation algorithm

A client c with grid (x_0, y_0) would like to communicate with k equivalent servers from m servers ($k \leq m$) in the network with $A \times B$ grids.
Iuput: $s \leftarrow min\{x_0, A - x_0 - 1\}$, $t \leftarrow min\{y_0, B - y_0 - 1\}$, $GN \leftarrow \lceil \frac{ABk}{m} \rceil$,
$h_1 \leftarrow min\{s, t\}$, $h_2 \leftarrow max\{A, B\}$, $h_3 \leftarrow min\{A, B\}$, $h_4 \leftarrow |s - t|$,
$C_1 \leftarrow 4h_1(h_1 + 1)$, $C_2 \leftarrow h_4[6h_1 + 5 + 2(h_4 - 1)]$,
$C_3 \leftarrow (h_3 - 2h_1 - h_4 - 1)[(4h_1 + 3h_4 + 3) + (h_3 - 2h_1 - h_4 - 2)]$.
Output: Manycast level value (*McLevel*)
if $C_1 + 1 \geq GN$ **then**
  find the smallest integer r such that $1 + 4r(r + 1) \geq GN$;
  $McLevel \leftarrow r$;
  return *McLevel*
else if $C_1 + C_2 + 1 \geq GN$ **then**
  find the smallest integer r such that $r[6h_1 + 5 + 2(r - 1)] \geq GN - C_1 - 1$;
  $McLevel \leftarrow r + h_1$;
  return *McLevel*
else if $C_1 + C_2 + +C_3 + 1 \geq GN$ **then**
  find the smallest integer r such that $r[4h_1 + 3h_4 + 3 + 2(r - 1)] \geq GN - C_1 - C_2 - 1$;
  $McLevel \leftarrow r + h_1 + h_4$;
  return *McLevel*
else
  find the smallest integer r such that $r \times h_3 \geq GN - C_1 - C_2 - C_3 - 1$;
  $McLevel \leftarrow r - h_1 + h_3 - 1$;
  return *McLevel*
end if

that servers located in level (*McLevel*+1) probably receive MREQ packets. For example, in Fig. 7, servers located in level 5 have a chance to receive MREQ packets from gateways located in level 4. In our scheme, we also allow servers located in level (*McLevel*+1) to reply a MREP packet, thus it can increase manycast successful rate. However, in order to avoid too many reply messages for a client, we let a server located in level (*McLevel*+1) determines whether it needs to reply a MREP packet or not depending on its reply probability P_r. In the following section, we simulate the performance of our scheme according to different reply probability P_r.

When a server received the MREQ packet and determines to reply a MREP packet to the client c, it uses greedy routing method [9] in our grid-base manycast scheme. Furthermore, for some special requirements and applications, another kind of routing protocols such as GRID protocol [8] that is more resilient and less vulnerable by utilizing grid-by-gird manner can further provide long time communication and better service quality between a client and servers.

4 Simulation

In this section, our grid-based manycast (GBM) scheme is compared to scoped-flood scheme that is modified to perform manycast delivery in [1]. In our

Algorithm 2. Grid broadcast algorithm

When a gateway g with grid (x, y) receives a MREQ packet initialized from a client c with grid (x_0, y_0).
$(x', y') \leftarrow (x - x_0, y - y_0)$ /* the relative grid coordinate of g with respect to c */
$L \leftarrow max\{|x'|, |y'|\}$ /* the level of g with respect to c */
if (gateway g receives a MREQ packet for the first time) and ($L \leq McLevel$) **then**
  if ($x' = 0$ and $|y'| \bmod 4 \neq 1$) or ($y' = 0$ and $|x'| \bmod 4 \neq 1$) **then**
    gateway g broadcasts the MREQ packet
  else if ($x'_0 \times y'_0 \neq 0$) and ($|\,|x'| - |y'|\,| \bmod 4 = 0$) **then**
    gateway g broadcasts the MREQ packet
  else
    gateway g discards the MREQ packet
  end if
else
  gateway g discards the MREQ packet
end if

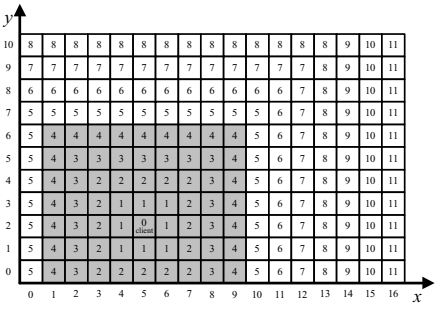


Fig. 6. An example of manycast level calculation

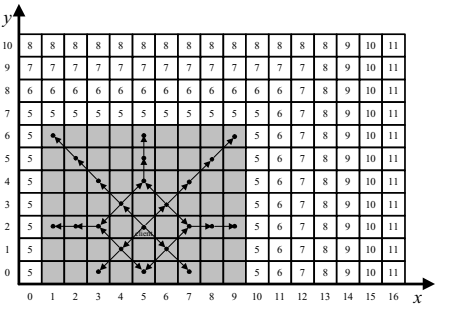


Fig. 7. An example of grid broadcast

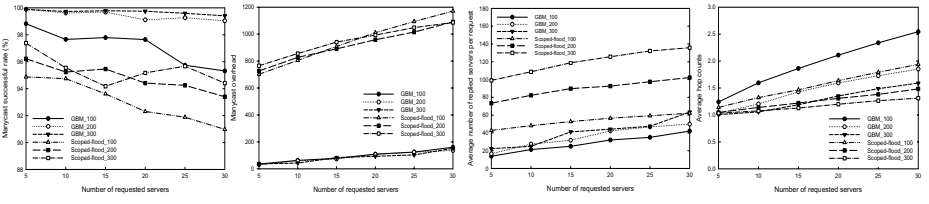
simulation environment, node movement uses the random way-point mobility model [10] with pause time 10 seconds for four simulation scenarios. All node with 250m transmission range are randomly placed in the network. A client is randomly chosen to initialize a manycast request in the simulation. Table 1 summarizes the parameters chosen for four simulation scenarios.

Four metrics are evaluated in the simulation to see how the performances of two compared schemes are. These metrics are described as follows.

- **Manycast successful rate:** The successful rate is presented by the percentage of the total satisfied manycast requests over the total number of manycast requests issued during simulation time. A manycast request is satisfied only when at lest k distinct server responses arrive back at a client.
- **Manycast overhead:** the average number of packet transmissions during server request and server reply phase per manycast delivery.

Table 1. Simulation parameters

Network size (grids)	17×17	Scenario 1,2,3
	10×10∼20×20	Scenario 4
Number of nodes	1500	Scenario 1,2,3
	550∼2200	Scenario 4
Number of servers	100,200,300	Scenario 1,2,3
	75∼290	Scenario 4
Number of requested servers (k)	5∼30	Scenario 1
	20	Scenario 2,3,4
Node mobility	10 m/s	Scenario 1,3,4
	5∼25 m/s	Scenario 2
Reply probability P_r	1	Scenario 1,2,4
	0∼1 with interval 0.1	Scenario 3
The side length of the grid	118 m	All scenarios



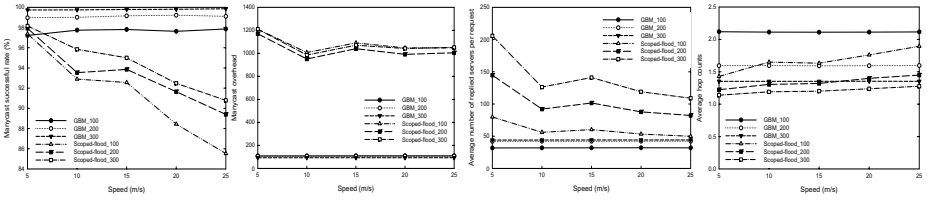
(a) (b) (c) (d)

Fig. 8. Scenario 1 : impact of different requested servers

- **Average number of replied servers per manycast request:** It is clear that the lower bound of this number must be at least k for a satisfied request. This metric can show reliability and quality of a manycast delivery if the replied servers are at least k and as closer to k as possible.
- **Average hop counts:** the average number of the total hop counts in the first k replying paths from servers to a client per satisfied request.

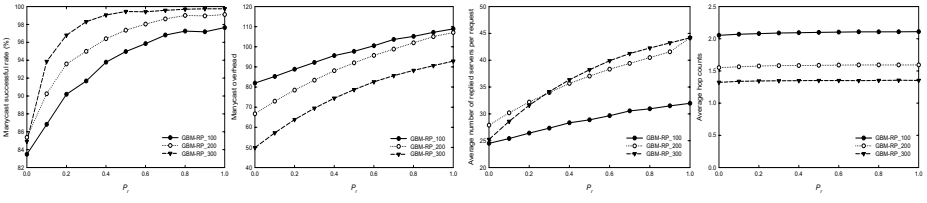
Fig. 8 shows the impact of different requested servers. Our scheme predicts a manycast region and uses an efficient grid broadcast method for finding at least k servers, thus it achieves higher manycast successful rate, much lower manycast overhead and less average number of replied servers than scoped-flood scheme. However, scoped-flood scheme using shortest paths has slightly less average hop counts than that of our scheme which utilizes greedy routing method.

Fig. 9 illustrates the impact of different node mobility. When node mobility increases, scoped-flood scheme shows poor manycast successful rate because of invalid TTL value which is obtained from previous manycast delivery. Our scheme still has better performance than scoped-flood scheme except average hop counts.



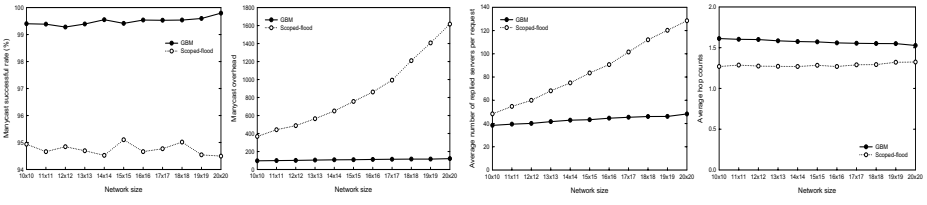
(a) (b) (c) (d)

Fig. 9. Scenario 2 : impact of different node mobility



(a) (b) (c) (d)

Fig. 10. Scenario 3 : impact of different reply probability P_r



(a) (b) (c) (d)

Fig. 11. Scenario 4 : impact of different network size (scalability)

In the Fig. 10, we evaluate our grid-based manycast scheme with reply probability (GBM-RP) under different P_r value. We can observe that reply probabilities P_r of servers located in level ($Mclevel$+1) must increase to keep higher manycast successful rate when the number of servers is smaller(e.g., 100 servers). But larger P_r also generates more manycast overhead and the number of replied servers. It is obvious that there is a tradeoff between these three metrics. In Fig. 10(c), the network with 100 servers has less average number of replied servers than that of the network with 300 servers because its average number of servers is fewer in each grid. In Fig. 10(b)(d), because the smaller number of servers results a larger manycast region, i.e. the value of GN in Algorithm 1, and more gateways need to rebroadcast MREQ packets, manycast overhead and the average hop counts of 100 servers are higher than those of 200 and 300 servers.

Finally, Fig. 11 shows the impact of networks sizes with different number of nodes and servers, i.e. scalability. Our scheme uses efficient grid broadcast

method instead of using experienced information such as a flooding in advance. Therefore, our scheme is scalable than scoped-flood scheme, especially in terms of manycast overhead and the average number of replied servers.

5 Conclusion

In this paper, we proposed a grid-based manycast scheme for large mobile ad hoc networks. In the server request phase, a manycast level calculation algorithm is presented to predict the number of grids that approximately contains k servers and form a manycast region, then an efficient grid broadcast method is used to find at least k servers that client specified. In the server reply phase, we use a simple reply probability method to reduce the number of replied servers. Simulation results indicate that our proposed scheme outperforms than scoped-flood scheme in most of the simulation metrics such as manycast successful rate, manycast overhead and the average number of replied servers under different scenarios.

References

1. C. Carter, S. Yi, P. Ratanchandani, R. Kravets, "Manycast: Exploring the Space Between Anycast and Multicast in Ad Hoc Networks," Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pp. 273–285, 2003.
2. Network Time Protocol (NTP) Version 4, http://www.ntp.org.
3. T. Wu, M. Malkin and D. Boneh, "Building Intrusion Tolerant Applications," Proceedings of the 8th USENIX Security Symposium, pp. 79–91, 1999.
4. L. Zhou, F.B. Schneider and R. van Renesse, "COCA: A Secure Distributed Online Certification Authority," ACM Transactions on Computer Systems, vol. 20, no. 4, pp. 329–368, 2002.
5. S. Yi and R. Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks," The 2nd Annual PKI Research Workshop Program (PKI03), 2003.
6. Y.C. Tseng, S.Y. Ni, Y.S. Chen and J.P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," Wireless Networks, vol. 8, no. 2/3, pp. 153–167, 2002.
7. C. Carter, S. Yi and R. Kravets, "ARP Considered Harmful: Manycast Transactions in Ad Hoc Networks," Proceedings of the IEEE Wireless Communications and Networking Conference, pp. 1801–1806, 2003.
8. W.H. Liao, Y.C. Tseng and J.P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," Telecommunication Systems, vol. 18, no. 1, pp. 37–60 ,2001.
9. G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," ISI Research Report ISI/RR-87-180, 1987.
10. D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in Mobile Computing, Kluwere Academic Publishers, pp. 153–181, 1996.

A Grid-Based Tracking Mechanism with Satisfaction of Energy Conservation and Guaranteed QoS in Wireless Sensor Networks

Sung-Min Lee and Hojung Cha

Department of Computer Science, Yonsei University
Seodaemum-gu, Shinchon-dong 134, Seoul 120-749, Korea
{sulee, hjcha}@cs.yonsei.ac.kr

Abstract. A power conservation technique is one of the essential solutions for sensor networks, which are for the most part not able to refresh their own power source. A sleep or active time control method is an effective power conserving solution because of its ability, in tracking applications, to turn off the radio of nodes that are not involved in packet transmission. However, most current research neglects the importance of QoS. In this paper, we present a sleep-time control mechanism based on a two-tier grid structure: communication grid and wakeup and sleep grid. The hybrid grid structure allows power conservation as well as preservation of QoS. Additionally, we present a framework for a mobile event tacking system, which consists of three phases: grid construction, surveillance, and tracking. The simulation result shows that the proposed method almost guarantees QoS while conserving a significant amount of energy.

1 Introduction

The recent advance of available technology on wireless sensor networks (WSN) has accelerated active research on various issues of many specific applications. Issues are usually related to the limitations of WSN, which involve restrictive computational power, memory use and power consumption [1]. Object surveillance and target tracking applications, which is one of the active research areas, also faces the above problems. The power conservation issue is the most critical since sensor nodes throughout the entire network field must frequently wake up to survey mobile objects and transmit packets to a designated node.

In an effort to find effective solutions for the early power depletion problem, a significant amount of research has already been conducted on tracking applications. The object tracking method with two modes, a surveillance mode, and a tracking mode; represent a sampling of the optimal power conservation solutions. Sensor nodes are working under the tracking mode whenever an event occurs in the sensor field and if not, they remain under surveillance mode. With the surveillance mode, the active duration of sensor nodes are lessened by increasing their sleeping intervals. On the other hand, when a mobile object is detected within the sensor field, sensors must

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 286 – 295, 2006.

reduce its sleeping time for accurate sensing and subsequently transmit data within necessary areas. Therefore, switching mode depends on the existence of events to lead to the power conservation effect.

Current existing object tracking methods tend to neglect consideration of quality of service (QoS, hereafter), which is the number of received data messages. There are many other factors to define the QoS, but location data is the most important one in the tracking application. Most of the previous research focuses on a mobile sink moving along a mobile event; so relaying sensed data to a sink is not an issue. Nevertheless, if the sink is located far from the event, relaying data will be a critical problem because of the inability to transmit data via sleeping nodes. In this case, power conservation is expected, whereas QoS is not guaranteed. If data cannot be delivered to the target, power conservation is meaningless, and therefore it is necessary to develop a mechanism satisfying both QoS and power issues.

In this study, we propose a sleep-time control mechanism based on the two-tier grid structure (SCM2G, hereafter), which conserves power while guaranteeing QoS. It uses a two-tier grid to organize sensor nodes: the first tier grid is for power conservation and the second one is for QoS. The first tier grid is the superset of the second tier grid. The second tier grid is constructed on the basic concept of GAF [2]. However, GAF alone cannot precisely predict movement of a mobile event. Thus, the first tier grid is required to predict and conduct an efficient node sleep and wakeup management by switching from surveillance mode to tracking mode.

The rest of the paper is organized as follows. Related research is discussed in Section 2. The proposed mechanism is explained in Section 3. The evaluation and conclusions are discussed in Sections 4 and 5, respectively.

2 Related Work

Research on power conservation focusing specifically on mobile target tracking applications has also been conducted. PEAS [3] and Mobicast [4] are well known solutions for the mobile target based on surveillance and tracking modes. Mobicast provides mobility of target to supplement the problem of Geocasting [5]. The work focuses on the "Just-in-Time," which means that the data messages arrive at designated nodes before their time expiration, so nodes do not have to wake up for a while. PECAS [6] is an optimized version of PEAS, which has a weakness at load balancing due to operating an activated node continuously without switching its role. PECAS rotates the role of the activated node, so it maintains network integrity. STEM [7] uses a separate radio for the paging channel to avoid interference with regular data forwarding, but it requires special hardware support. IDSQ [8] is a similar solution to others in which each sensor node performs detection by comparing measurement with a threshold. If nodes detect an event, and select a leader, then the leader suppresses the other nodes to prevent multiple tracks for the same target. The latest research on this issue, DTM and OTC [9], is designed to reduce communication overhead and to increase accuracy for prediction of mobile nodes' movement. DTM uses a moving

tree to wake up sleeping nodes before the moving event arrives, but it uses a motion profile, which does not have sufficient accuracy for prediction of the mobile event's movement. Thus, OTC is proposed as a way to overcome the prediction problem, but it also encounters problems with the large amount of communication overhead. Two separate protocols, which have opposite advantages to each other, may be combined; however, when merging two protocols, the increment of communication overhead is inevitable. The above studies propose reasonable solutions for their own settings and assumption, but overlooking the QoS issue leads to a lack of generality of the power conservation solution in the mobile target tracking environment.

3 Sleep-Time Control Mechanism Based on Two-Tier Grid

3.1 Overview

For a mobile target tracking application, we propose a sleep-time control mechanism based on the two-tier grid structure (SCM2G) operating on the virtual grid structure, which guarantees QoS and conserves power consumption simultaneously. There are several advantages of using the grid structure on this particular application. First, it helps to predict the movement of a mobile target by local time synchronization and neighborhood management. Most tracking applications use a time stamp to distinguish the order of data. Accomplishing the time synchronization is an expensive process, but using a grid to organize nodes and relay time synchronization data may reduce network overhead. In this paper, time synchronization is not discussed and it is assumed to be supported by the underlying system. Another advantage is the ease of node groupings that will wake up or sleep soon. Errors generated in the process of prediction are expected, so waking up nodes within a certain range is required to anticipate sensing the mobile event. Of course, after the event moves away, the activated group of nodes should go back to sleep. A grid simplifies grouping nodes and makes it easy to manage the neighborhood in the mobile target tracking application. Moreover, it makes it possible to create nested groups of nodes, which, in order to guarantee QoS, are organized to connect the whole sensor field.

Assumptions for our research include the following.

- Each node is aware of its own location
- Each node knows the amount of power consumption
- A single static sink which has unlimited computation power and resources is used
- The event can move to any direction and at any speed
- A user knows the maximum speed of the mobile event, but does not know its current speed
- Time synchronization is already provided

Fig. 1 explains the general framework of SCM2G based on above the assumptions. Deployed nodes are organized by grid construction mode, and then groups of nodes work under either the surveillance or tracking mode according to existence of the mobile object.

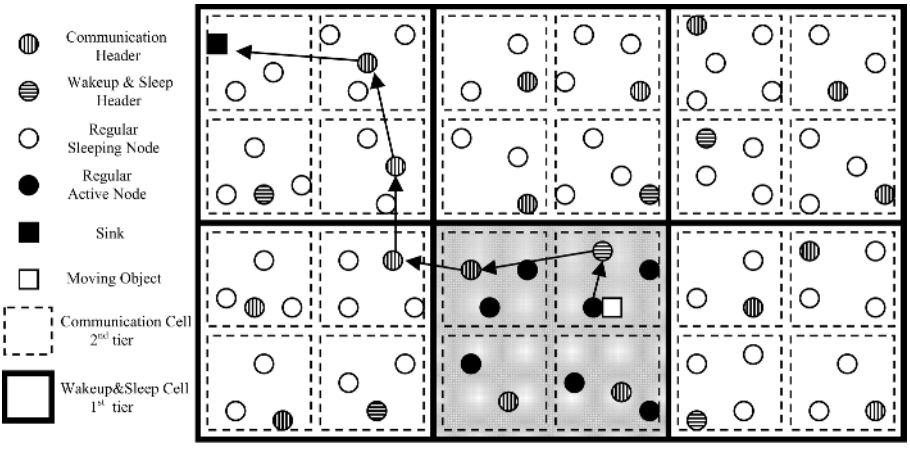


Fig. 1. An example of SCM2G framework

As illustrated in the above figure, the entire sensor field is divided into wakeup and sleep cells and each of them is again divided into communication cells. The rationale for using a two-tier gird is in satisfying both QoS and power conservation. The best case scenario is in using a single communication grid to save power consumption and guarantee QoS. However, the communication grid is constructed by the maximum or optimal radio range, so it does not consider the speed of the event and prediction frequency. For example, if the maximum speed of an object is 20m/s and the maximum radio range is 10m, the tank enters into and leaves from the cell in less than one second. This means that there is no sufficient time to predict the next destination of the event or even relay wakeup messages to the next cell. Moreover, for precise calculation of the destination of the event, several operations of prediction are required, but in this case, the size of the communication cell is too small. Conclusively, without the accurate prediction, nodes cannot sense the event correctly and the entire tracking system is useless. Therefore, a wakeup and sleep grid is required to solve that problem. In each of the communication cells, there is a single communication head node. One communication node (in this example, it is four) is set as the master node, called wakeup and sleep head node in a wakeup and sleep cell. In tracking mode, the head node predicts the next destination of the moving object and sends a wakeup message to that designated wakeup and sleep cell. As soon as the designated head node receives the message, it starts to broadcast a wakeup message within the cell to turn on their radios and sensors. In the shaded area, nodes receive the wakeup message and switch their mode to wake-up for preparing and sensing the event. When the master node does not receive sensing data within a certain period of time, it sends sleep messages within the cell to conserve power consumption. If there is no event, every node except the headers goes into the sleep mode, which is called the surveillance mode. Communication headers relay location data to the sink, but do not perform any computations. The initial mode is a grid construction and the control moves to one of two states, depending on the existence of an event. In the initial mode, if the grid size is too big, the energy consumption increases and if the opposite situation occurs,

prediction accuracy decreases. The following sections explain how to construct the two-tier grid structure considering both the power issue and accuracy. The surveillance mode and tracking mode are both explained in Section 3.3.

3.2 Two-Tier Grid Construction

The grid construction is an important process of SCM2G. In a dynamic environment, finding the right size of grid to save power consumption and maintain QoS is a difficult task. In order to find the optimal size of a grid l, the maximum speed v of the mobile event, the maximum or optimal radio range a of two nodes, prediction frequency f, delay time d, and the size xy of the sensor field are all required. Delay time is the elapsed time for wakeup messages to reach the designated cell. Using the above factors, we set the range of f as shown in Equation 1. v and xy are given numbers and d is too small to apply because the speed of a radio signal is close to light speed. According to the equation, a user can select at least one prediction time in which the accuracy should be low and it can go up to a certain number. That number calculation is based on the minimum number of cells, which is two. If the number of cells is two, the reduction rate of power consumption should be close to 50%. Thus, the frequency can increase up to the cell being divided into two cells. The user chooses f depending on the purpose within the range; a higher f gives higher accuracy and lower power reduction.

$$d \times v \le (d \times f) \times v \le \left\lceil \frac{\sqrt{xy}}{2} \right\rceil, \quad 1 \le f \le \left\lceil \frac{\sqrt{xy}}{2 \times d \times v} \right\rceil \tag{1}$$

With d, v, and f, we can calculate the temporal length l of the wakeup and sleep cell as shown in Equation 2. When f increases, l increases as well, so careful selection of f is required.

$$l = f \times d \times v \tag{2}$$

Due to the lack of considering the communication factor for QoS, l is not the final length of the wakeup and sleep cell. First, the length of the communication cell (b) is calculated by Equation 3. This equation is created by the diagonal length of the two combined communication cells, which equals to the maximum or optimal radio range a as the given number. Using this type of construction method allows any nodes in two communication cells to transmit and receive data. Therefore, only a single node in the communication cell is required to wakeup to connect the entire sensor field and that leads to a reduction of power consumption. This idea is adopted from GAF.

$$a = \sqrt{b^2 + 4b^2}, b = \left\lfloor \frac{a}{\sqrt{5}} \right\rfloor \tag{3}$$

Finally, the calculation for the length l_f of the wakeup and sleep cell reaches to the end when comparing l and b. The detailed process of the last comparison and calculation is shown in Equation 4. The temporal length becomes the final one only if l is

smaller than or equal to b, because it means that using l still satisfies QoS. If the condition is the opposite, the resized length is required.

$$l_f = \begin{cases} l & , \ l \le b \\ \left\lceil \frac{l}{b} \right\rceil \times b & , \ l > b \end{cases} \tag{4}$$

After the grid construction is finished, the surveillance mode proceeds. When an event occurs, the current mode switches from surveillance to tracking mode. The next section describes detailed processes for two modes.

3.3 Surveillance and Tracking

In surveillance mode, except for the header nodes, sensor nodes operate on the sleeping mode at most times. The sleeping nodes wake up to maintain their time synchronization or switch the role of header, but the wakeup period is long enough to save significantly on power consumption. On the other hand, the wakeup period for the header nodes is relatively short in order to preserve network connections and reduce sensing failure at all times. However, using fixed header nodes impedes the integrity of network connections due to the increment of malfunctioned nodes while time elapses. Therefore, a dynamic swap header technique is required to solve the critical problem.

There are two possible mechanisms for swapping header nodes; one uses a timetable to schedule sleeping periods like TDMA and the other checks residual amounts of power. The first method calculates sleeping time t_n for the n-th node, as shown in Equation 5. t_e is the entire time slot in a single wakeup and sleep cell and t_a is the activation time slot of the first head node. N is the number of nodes in the cell. Thus, each node works as the header for the same amount of time. The other solution is to simply check the amount of power left. It is a simple and effective solution, but must have hardware support.

$$t_a = \frac{t_e}{N}, \quad t_n = t_a \times n \tag{5}$$

The system remains in the surveillance mode as long as an event is not detected. When the event occurs, however, it switches from the surveillance mode to the tracking mode. In the tracking mode, each of the wakeup and cell head node, which is a communication head node, receives location data of sensor nodes and then predicts the next destination of the event. The method of prediction is with a simple calculation of the two coordinates of the sensed nodes. However, a single calculation deteriorates the accuracy of prediction, so an exponentially weighted moving average (EWMA) may be needed. Afterwards, the head node sends the wakeup messages to the destination cell so that sleeping nodes can prepare to sense the event. When the system works on this mode, there is no swapping operation for head nodes. When the head node does not receive location data in a certain length of time, it will send sleep messages within the current cell, so sensor nodes go into the surveillance node again. In future research, the surveillance and tracking mode will be minutely examined and discussed in regard to both problems and solutions.

4 Evaluation

The evaluation of SCM2G, which focuses on both energy conservation and QoS, is discussed as based on simulation data. TOSSIM [10], which is specifically designed for a sensor network environment, is used. Two distinct sizes of sensor fields (200m X 200m, 80m x 80m) are selected for evaluation; the larger is for grid size since several different sizes of cells are required to observe their tendency. The other one is used for the QoS and power conservation test. In this test, 80 sensor nodes and one sink node are deployed with a methodical and proportionally distributed manner. In addition, energy consumption is calculated based on the Crossbow's MICA2 specification and QoS is defined as a number of received messages over a number of sent messages.

4.1 Energy Consumption Depending on Grid Size

The grid construction method is verified by making comparisons with existing work. As explained above, the size of the sensor field is 200m X 200m, which suggests that the range of *f* is from 1 to 7, according to Equation 1. Increasing *f* means enlarging the size of the grid cell. It leads to the augmentation of power consumption, because of the increment of a number of wakeup control messages, which must disseminate throughout a designated cell.

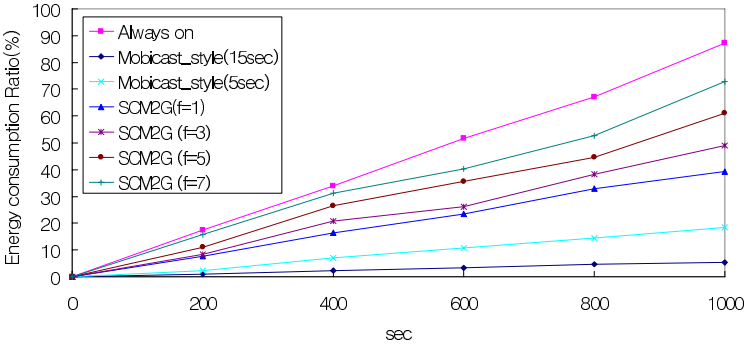


Fig. 2. Comparison of energy consumption ratio while time elapses

In Fig. 2, we tested four SCM2G methods with different sizes of cells, a method that always turns on radios of nodes, and two Mobicast methods with different lengths of sleep time. The graph shows that Mobicast is superior to others, and the proposed method is in between the 'always on' method and Mobicast. It means that SCM2G (f=7), which has the largest cell size for the particular sensor field, still reduces power consumption in comparison to the 'always on' method. Therefore, proper operation for grid construction is substantiated. Mobicast outperforms the proposed method; nonetheless, power conservation without considering QoS is inadequate.

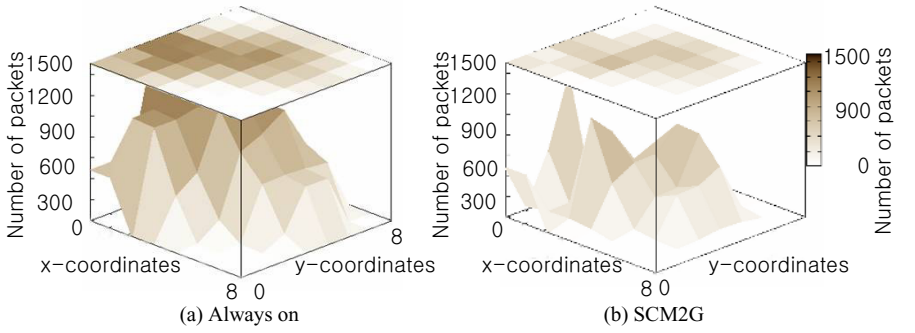


Fig. 3. Comparison of energy consumption ratio while time elapses

A significant amount of energy is conserved by applying SCM2G instead of the 'always on' method when packets relay from the middle of a node to a sink as shown in Fig. 3. SCM2G only uses a small number of communication cell headers to transmit data. Moreover, most nodes, except headers, turn off their radios, so preserving energy consumption when receiving packets is additionally expected.

4.2 Energy Conservation and QoS

The following graphs show the result of various experiments conducted to observe the precise effects on power conservation and QoS.

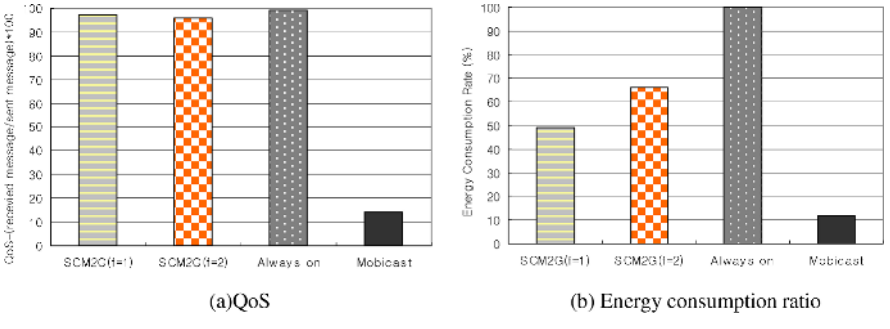
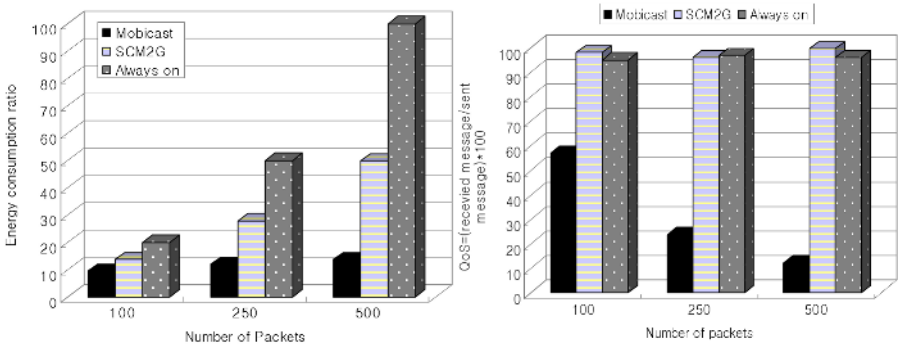


Fig. 4. Comparison of the energy consumption ratio and QoS

For Fig.4, we used 80m X 80m, so the biggest *f* a user can select is 2. Mobicast shows inadequateness for preserving QoS, although it conserves a lot of energy in comparison to others. The result on QoS of Mobicast is obvious because of insufficient preparation of data replaying mechanism. On the other hand, even if SCM2G sacrifices some amount of energy consumption in comparison to Mobicast, it guarantees QoS using communication headers. Nodes working on the 'always on' method

never go to sleep mode, so its QoS is guaranteed. However, its energy consumption is much higher than SCM2G because of lack of sleep-time control mechanism. In addition, even larger grid still conserves about 35% of energy comparing to the 'Always on' method.



(a) Energy consumption ratio (b) QoS

Fig. 5. Comparison of the energy consumption ratio and QoS with various numbers of packets

The above graphs are the result of energy conservation and QoS while the number of data packets increases. More packets within the fixed time duration, 500sec, mean more accurate location data for the user. As shown in Fig. 5, more packets degrade QoS of Mobicast; meanwhile SCM2G preserves QoS close to 100% in any condition. Mobicast only wakes up and shortens the sleep-time of nodes when located around a mobile event, so increasing the rate of packet loss is inevitable while the number of packets increases. Of course, the regular method consumes enormous amounts of energy because of the increasing packet transmission. In Fig. 5(b), QoS of Mobicast is about 50% while the number of packet is 100. Nevertheless, it is not still sufficient to satisfy for tracking applications. For example, if speed of a moving object is 5m/s and one packet is generated per 5sec (100 packets in 500sec), the moving object moves about 25m after 5sec. In this case, Mobicast only guarantees 50% of QoS. The performance on this particular environment cannot be accepted by most of tracking applications for lack of accuracy. Therefore, the conclusion that SCM2G conserves energy consumption with guaranteed QoS is reached.

5 Conclusion

Current work on tracking mobile objects mainly considers power conservation. However, the major role of networks is in transmitting data to designated locations with a reliable and accurate manner. As illustrated by the experiment, less than 50% of QoS is not satisfied with tracking mobile object applications in sensor networks. The proposed method, SCM2G, supplements the weakness of Mobicast while conserving sufficient amounts of the energy consumed because the sleep time control mechanism prevents useless radio transmission and provides a longer sleep time. The result of

simulation, which our method provides is close to 100% of QoS while it conserves about 50% of energy consumption, and proves an improvement of performance.

For future research, accurate predictions for mobile object direction and efficient time sync solutions remain. A small number of data messages will be generated by an accurate prediction and efficient time sync solutions may reduce flooding messages. Consequently, SCM2G, including the above future work, is expected to yield better performance.

Acknowledgement

This work was supported in part by the National Research Laboratory (NRL) program of the Korea Science and Engineering Foundation (2005-01352) and the ITRC programs (MMRC) of IITA, Korea.

References

1. I. F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayrici: A survey on sensor networks. IEEE Commun. Mag., Vol. 40. (2002) 102-114.
2. Xu, S. Bien, Y. Mori, J. Heidemann and D. Estrin: Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks. Technical Report 6, Center for Embedded Networked Computing, Los Angeles (2003).
3. F. Ye, G. Zhong, S. Lu, and L. Zhang: Peas: A robust energy conserving protocol for long-lived sensor networks, in Proc. The 23rd International Conference on Distributed Computing Systems, Rhode Island (2003) 169-177.
4. C. L. Tan and S. Pink: Mobicast: a multicast scheme for wireless networks, ACM Mobile Networks and Applications, (2000) 259-271.
5. Y.-B. Ko and N. H. Vaidya: Geocasting in mobile ad hoc networks: Location-based multicast algorithms, in Proc. IEEE Workshop on Mobile Computer Systems and Applications, New Orleans (1999) 101-110.
6. C. Gui, P. Mohapatra: Power conservation and quality of surveillance in target tracking sensor networks, in Proc. 10th annual international conference on Mobile computing and networking, Philadelphia (2004) 129-143.
7. C. Schurgers, V. Tsiatsis, M. B. Srivastava: STEM: Topology Management for Energy Efficient, in Proc. IEEE. Aerospace Conf, Montana (2002) 135-145.
8. M. Chu, H. Haussecker, and F. Zhao: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, International Journal of High Performance Computing Applications, Vol. 16. (2002) 90-110.
9. S. Bhattacharya, G. Xing, C. Lu, G. Roman, O. Chipara, B. Harris: Dynamic Wake-up and Topology Maintenance Protocols with Spatiotemporal Guarantees, in Proc. Information Processing in Sensor Networks, Los Angeles (2005) 28-34.
10. P. Levis, N. Lee, M. Welch, D. Culler: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications, in Proc. ACM SenSys'03, Los Angeles (2003) 126-137.

Frame Size Adaptive MAC Protocol in Low-Rate Wireless Personal Area Networks

Eun -Chang Choi[1], Jae-Doo Huh[1], Kwang-Sik Kim[2], Moo-Ho Cho[3], and Soo-Joong Kim[4]

[1] Sensor Networking Research Team
Electronics and Telecommunication Research Institute
161 Gajeong-Dong, Yuseong-Gu, Daejeon 305-350, Korea
{ecchoi, jdhuh}@etri.re.kr
[2] The Korean Intellectual Property Office
Government Complex-Daejeon Bldg. 4 920, Dunsan-Dong, Seo-gu,
Daejeon 302-701, Korea
kskim@kipo.or.kr
[3] School of Computer and Multimedia Engineering, KyongJu University
San 42-1, Hyohyun-Dong, KyongJu 780-712, Korea
mhcho@kju.ac.kr
[4] School of Electronic and Electrical Engineering, Kyungpook National University
161 Sankyuk-Dong, Buk-Gu, Daegu 702-701, Korea
sjkim@knu.ac.kr

Abstract. Based on IEEE 802.15.4 standard for Low-Rate Wireless Personal Area Network (LR WPAN), this paper proposes a frame size adaptive Medium Access Control (MAC) protocol. Mean Burst length and mean Gap between bursts are measured by using training frames between two communication de-vices. With the mean Burst and Gap values, an optimal MAC frame size to meet the required FER is calculated. During communication period, FER is periodi-cally measured in a transmitting device. If the measured FER is lower than the required FER, MAC frame size is increased, else vice versa. Thus, channel resources can be more effectively utilized by sending optimal bits into one trans-mission frame. The proposed scheme is evaluated under various wireless chan-nel conditions in terms of the achieved throughput. Simulation results show that the proposed scheme achieves a much higher throughput than a non-frame size adaptive MAC protocol in LR WPAN does.

1 Introduction

The IEEE 802.15.4 Task Group (TG) has been chartered with creating Low-Rate (LR) WPAN standard and has published a final draft standard recently [1]. The main fea-tures of the standard are network flexibility, low cost, and low power consumption. One of the largest application opportunities for IEEE 802.15.4 is home automation and networking [2]. Home networks support communication between PCs, laptops, PDAs, cordless phones, smart appliances, security and

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 296–305, 2006.

monitoring systems, consumer electronics, and entertainment systems anywhere in and around the home. The use of wireless technology is the reduction in installation cost, because new wiring is not needed. Wireless networking conveys information exchange with minimal installation effort. This trend follows from the wider availability of cheaper and highly integrated wireless components and the success of other wireless communication technologies [2]. One of the design difficulties is to support the various QoS (quality of service) requirements for different home networking applications [3]. QoS in this context re-fers to the requirements of a particular application, typically data rates and delay constraints, which can be quite stringent for home entertainment systems. Since a typical wireless channel is time-varying, an efficient communication system can be achieved by cross-layer adaptation [3]. High Rate (HR) WPAN supports five different data rates by selecting a data rate according to the wireless channel condition [4]. But LR WPAN supports only one data rate of 250 kbps in the 2.4GHz band, and the rate adaptation can not be applied [1]. Under time-correlated fading channel com-munication environments, as frame size increases, frame error rate also increases [5]. Thus, MAC frame error rate depends on the frame size. There is little research on MAC fame size adaptation in wireless communication. We believe an efficient com-munication system can be achieved by selecting a MAC frame size according to the wireless channel condition. Thus, the frame adaptation scheme presented in this paper will be better than the rate adaptation scheme in some wireless channel conditions. In this paper we focus on the analysis of frame size adaptation protocol in LR WPAN for achieving a better performance and an efficient use of channel time. The proposed frame size adaptive MAC protocol is described in Section 2. In Section 3, the per-formance of our proposed scheme is evaluated. In Section 4, we provide conclusions.

2 The Proposed MAC Protocol

2.1 Motivation

IEEE Std 802.15.4 MAC provides the creation of two wireless network topologies: star topology and peer-to-peer. In the star topology, device communicates with each others via the coordinator that operates as a network master. The peer-to-peer topol-ogy enables the creation of the ad hoc, self-organizing wireless network. Two devices directly communicate through peer-to-peer connectivity without the intervention of the coordinator. The channel condition in IEEE 802.15.4 is estimated based on the results of attempted transfers of data frames between two full function devices(FFDs) that are actively participating in a data transfer. Thus in peer-to-peer topology, the channel estimation and the MAC frame size selection have to be done by a pair of FFDs par-ticipating in a communication. However, the coordinator needs to know the selected MAC frame size in order to allocate an optimal channel time for the communication of the pair of FFDs. In addition, a source FFD can transmit multiple frames to one target FFD during allocated channel time [1]. In this situation, the

frames within a GTS(Guaranteed time slot) may experience different channel quality that leads to a MAC frame size change. Thus, a mechanism needs to monitor the channel quality change. In order to solve such problems mentioned above and to enhance the through-put performance, we propose a frame size adaptive MAC protocol for IEEE 802.15.4. The main application for our proposed protocol is file transfer of asynchronous burst data such as Music files (average 3Mbytes) and image file (average 500 Kbytes) de-fined in [4]. In addition, this application uses fully acknowledged protocol for transfer reliability as acknowledgements.

2.2 Frame Size Adaptive MAC Protocol

We present a frame size adaptive MAC protocol for LR WPAN. Once the initial data rate is chosen using mean SNR (Signal to Noise Ratio), a data rate is fixed until a communication between two FFDs finishes. At first, Average Burst length (ABL) and Average Gap (AGL) between bursts are measured by using training frames between two communication FFDs or fading channel model. Then, maximum MAC frame size(L) is obtained from the ABL and AGL, while Initial FER (L) must be lower than required FER. Periodically, FER is measured after two FFDs starting communication. If calculated FER is lower than the required FER, frame size is increased to $L+\triangle$, where $\triangle$ is a group of bits (64bits or 128 bits) and depends on channel conditions. Otherwise, if calculated FER is higher than required FER, frame size is decreased to $L+\triangle$. In this way, the channel can be more effectively utilized by squeezing optimal bits into one transmission frame.

2.3 MAC Frame Error Modeling

The LR WPAN system uses OQPSK schemes to achieve 250 kbps transfer rates with-out specified error correction coding. It is assumed that convolutional codes are used [6]. Most of approaches assume that error occurs independently between bit streams in a frame [5]. However, error event happens as bursty type in real communication sys-tems that use Viterbi decoding as a channel coding scheme [5]. In this paper, frame error rate (FER) model is derived based on probability distribution of burst.

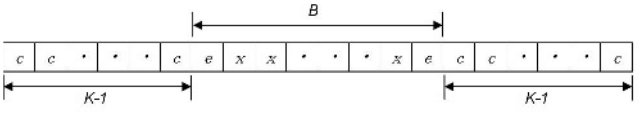


Fig. 1. Concept of burst and gap

At first, let's describe the probability distributions of Burst and Gap between consecutive Bursts. Fig. 1 shows output bit stream of Viterbi algorithm for the

convolutional coder with code-rate of $1/n$, constraint length of K. Character 'c' means the decoded bit without error, 'e' does the bit with error, and 'x' does 'c' or 'e'. The bit stream of $xxx\cdots x$ do not include $K-1$ consecutive 'c' and bit stream of $exxx\cdots xe$ is defined as 'burst' of the length B. The bit stream of c's between two consecutive bursts is defined as 'gap' with the length G, which is larger than $K-1$.

Miler [7] used Geometric distribution for modeling probability distribution of burst and gap. Model parameters are AGL, ABL and mean bit error rate within a burst. The probability distribution of burst length is given by

$$P(B=l) = p(1-p)^l - 1, l > 0 \tag{1}$$

where $p = 1/\overline{B}$, $\overline{B}$ means ABL.

FER is mainly affected from probability distribution of gap which is defined as $P(G=g)$. $P(G=g)$ means g consecutive bits without error in output of Viterbi decoded bit stream. The probability distribution of gap length is given by [7]

$$P(G=g) = q(1-q)^{g-K+1}, \forall g \geq K-1 \tag{2}$$

where $q = \frac{1}{\overline{G}=K+2}$, $\overline{G}$ means AGL.

Ref. [8] performed computer simulation to show the agreement between the Geomet-ric distribution proposed by Miler [7] and the result of simulation about PDF of burst length. FER is calculated based on probability model of Burst. Fig. 2 shows concept of Burst event. As shown in the figure, burst event can occur more than zero in a frame interval.

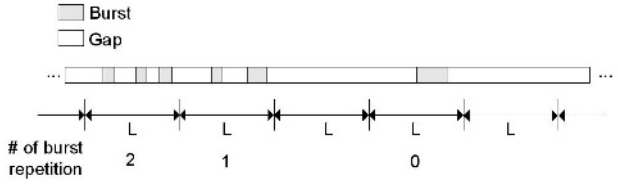


Fig. 2. Concept of Burst events

Define $FER(L)$ as Frame error rate in MAC frame with the length L. By some manipu-lation, $FER(L)$ can be given by

$$FER(L) = \frac{\overline{B}+L-1}{\overline{B}+\overline{G}}(1-q)^{L/2-K+1-\overline{B}/2} \tag{3}$$

where the range of frame size L is , $4K\text{-}4+\overline{B} \leq L \leq \overline{G}+1$,$q = \frac{1}{\overline{G}-K+2}$,and K constraint length of convolution coder.

Channel model in LR WPAN is different from COST-207 model used in terrestrial DAB mobile communication [9]. However two systems will present the similar FER in the same ABL and AGL because the same convolutional coding scheme is used for both systems.

2.4 MMAC in IEEE 802.15.4

The LR-WPAN standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. The superframe is bounded by net-work beacons, is sent by the coordinator (see Fig. 3) and is divided into 16 equally sized slots. The beacon frame is transmitted in the first slot

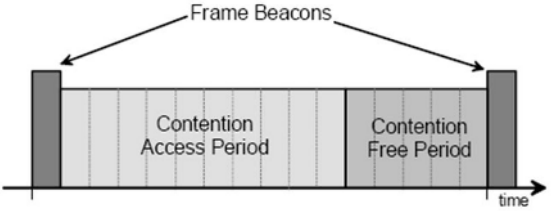


Fig. 3. A superframe structure in IEEE 802.15.4 MAC

of each superframe. If a coordinator does not wish to use a superframe structure, it may turn off the beacon transmissions. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes. Any device wishing to communicate during the contention access period (CAP) between two beacons shall compete with other devices using a slotted CSMA (Carrier Sense Multiple Access)/CA(Collision Avoid-ance)mechanism. All transactions shall be completed by the time of the next network beacon. For low-latency applications or applications requiring specific data bandwidth, the PAN coordinator may dedicate portions of the active superframe to that applica-tion. These portions are called guaranteed time slots (GTSs). The GTSs form the contention-free period (CFP), which always appears at the end of the active super-frame starting at a slot boundary immediately following the CAP, as shown in Fig. 3. The PAN coordinator may allocate up to seven of these GTSs, and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP shall remain for contention-based access of other networked devices or new devices wishing to join the network.

2.5 Throughput of the Proposed MAC

In this section, throughput in a GTS is derived. In our proposed scheme, the channel time for each GTS are evenly divided for all FFDs in a superframe, if traffic types of all FFDs assume the same priority. The channel time T_{GTS} of a GTS in a superframe is obtained by

$$T_{GTS} = T_{CFP}/N_{FFD} \tag{4}$$

where T_{CFP} is a duration of the CFP in the superframe and N_{FFD} is a number of FFDs in a piconet. When a FFD requests a GTS to the coordinator, the data rate for the frame transmission is informed to the coordinator. Therefore, coordinator can estimate the time duration for one frame transmission [4]. The throughput TP_{GTS} of the proposed MAC is given by

$$TP_{GTS}(L) = (T_{CFP}/T_{STP})(R/N_{FFD})[(1 - FFR(L))T_{frame}(L) + T_{overhead}] \quad (5)$$

where T_{STP}, $T_{frame}(L)$ and $T_{overhead}$ are the superframe transmission time, the data frame transmission time and overhead time consisting of the preamble, PHY/MAC headers, two SIFSs, and ACK frame produced by one data frame transmission. $T_{frame}(L)$ is changed according to L with fixed $T_{overhead}$ value. Overhead may be 5∼25% compared to $T_{frame}(1024bits)$ in constant MAC frame size [4]. In considering MAC frame size adaptation, overhead will be longer. $T_{frame}(L)$ is derived from

$$T_{frame}(L) = L/R \quad (6)$$

where L is the MAC frame payload size with the FCS and R is the date rate. The over-head time among a frame transmission time is given by

$$T_{overhead} = \alpha T_{frame} \quad (7)$$

where α is the ratio of overhead time to frame time in maximum MAC frame size. α is proportional to data rate where 0.05∼0.25 is reasonable for 250kbps.

3 Performance Analysis

3.1 Simulation Setting

We assume that all nodes are uniformly distributed in the coverage area of a POS, which is 10 meter radius in [10], and within the radio range of each other. For simplic-ity, we assume that the headers of all types of packets are always reliably received. Since the control and command frames are much shorter than data frames, no trans-mission failure of these frames are considered for simplicity. The parameters used in this simulation study are chosen based on the IEEE 802.15.4 standards [1]. We com-pare the throughput achieved by the following two different configurations: Case 1: protocol with variable MAC frame size (VMFS); Case 2: protocol with constant MAC frame size (CMFS) In Case 1, once the initial data rate is chosen to result in maximum throughput by some calculation of Eq. (5), a rate change is not allowed until a communication be-tween two FFDs finishes and frame size is adjusted to satisfy the required FER (8%) in an interval. In Case 2, once the initial data rate is chosen, a rate change is also not allowed until a communication between two FFDs finishes. In Case 2, but the data rate is chosen to satisfy maximum frame size. Once the initial MAC frame size is chosen, a MAC frame sizechange is not allowed until a communication between two FFDs finishes.

We evaluate the performance of the proposed scheme in a time-correlated fading channel with 8Hz Doppler frequency, which corresponds to a pedestrian speed (1m/s). The fading gain is generated according to the modified Clarke and Gans fad-ing model [11]. We set the path loss exponent to 2 in the log-distance path loss model,which is obtained from measurements in an office environment at 2.4 GHz in [12]. The transmit power is 0dBm that complies with FCC rules [10].

3.2 Simulation Results

Several ABL and AGL value are considered to cope with various channel conditions. Fig. 4 shows the throughput according to MAC frame size L without

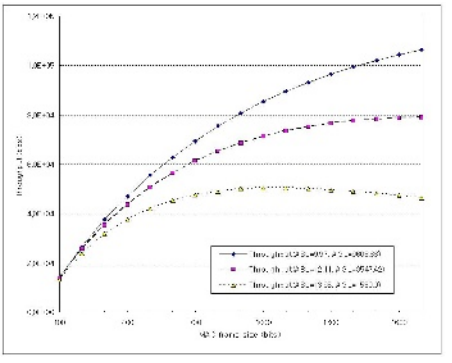


Fig. 4. Throughput according to MAC frame size L without considering required FER

considering re-quired FER. We assume that overhead ratio α is 20% in constant MAC frame size of 1024bits and constraint length K of Viterbi decoder is 3. And ABL and AGL values used are obtained from the results of channel model of Ref. [8]. Date rate is 250kbps. As shown in Fig. 4, MAC frame sizes that make maximum throughput are shown depending on the ABL and AGL values used. With ABL=13.95, AGL=1560.0, Maximum throughput is obtained in MAC frame size near 1000 bits. In better channel conditions such as ABL=9.97 and AGL=9606.66, maximum throughput is obtained in MAC frame size of more than 1000bits as proposed in Ref. [4].

Fig. 5 shows the throughput according to required FER with considering required FER. As required FER increases, throughput transmitted increases. As AGL increases, also throughput transmitted increases. In good channel conditions such as ABL=9.97 and AGL=9606.66, the throughput approaches near 100 kbps. In worse channel condition such as ABL=13.95, AGL=1560.0, but maximum throughput is 23.7kbps. That is because the available MAC frame size is lower than 200 bits.

Fig. 6 shows the throughput according to variable AGL. We assume that ABL is 8.27, data rate is 250kbps and frame sizes are 100, 300, 500, and 1000

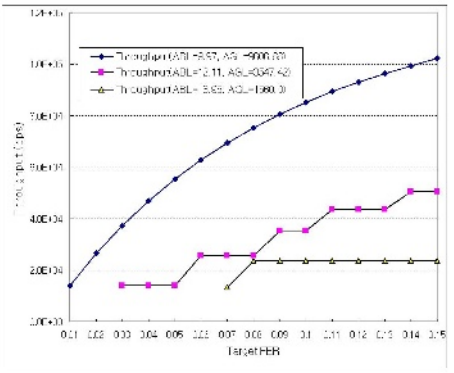


Fig. 5. Throughput according to required FER

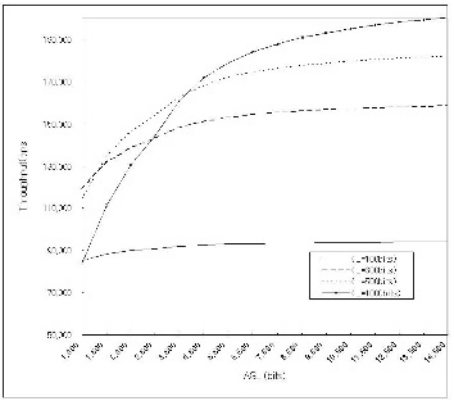


Fig. 6. Throughput according to variable AGL

bits. The fame size that presents the highest throughput depends on wireless channel conditions. In low AGL (under 1,000bits) which means bad wireless channel condition, MAC frame size of 200bits results in the best throughput than others. In high AGL (over 4,500bits) which means good wireless channel condition, MAC frame size of 1000bits results in the best throughput than others.

Fig. 7 shows the throughput according to variable ABL. We assume that AGL is 1560.0, data rate is 250kbps and frame sizes are 100, 300, 500, and 1000 bits. The fame size which presents the highest throughput is 500bits not 1000bits and is not changed. From the result of Fig. 7, we know that the best frame size in VMFS scheme may be mainly selected by AGL not ABL.

Fig. 8 shows the throughput of VMFS and CMFS scheme according to AGL. We assume that ABL is 8.27, data rate is 250kbps. In VMFS scheme, frame sizes are changed according to wireless channel conditions, but in CMFS scheme, the

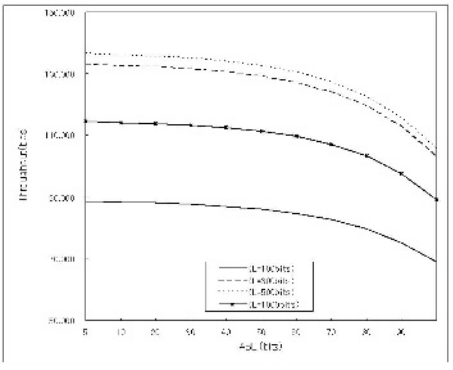


Fig. 7. Throughput according to variable ABL

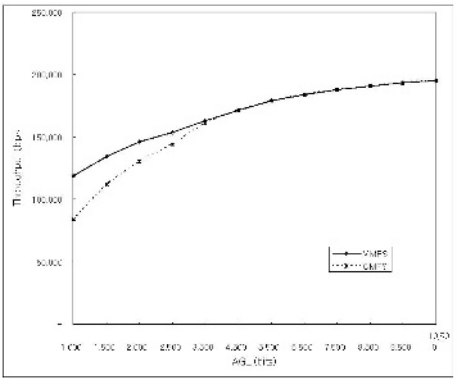


Fig. 8. Throughput of VMFS and CMFS according to variable AGL

frame size is fixed as 8192bits. In low AGL (under 3,500bits), VMFS results in the better throughput than conventional CMFS. In high AGL (over 3,500bits), both scheme presents the same throughput. Therefore, the proposed VMFS scheme is better than the CMFS scheme for all data rate.

4 Conclusions

In this paper, we proposed a frame size-adaptive MAC protocol for LR WPAN to support various QoS requirements for different home networking applications. Under time-correlated fading channel communication environments, frame error rate in-creases as frame size increases. By using training frames between two communication devices, mean Burst length and mean Gap between bursts are measured. With the mean Burst and Gap values, an optimal MAC frame size to

meet the required FER is calculated. During the communication period, FER in a transmitting FFD is periodi-cally measured. And the MAC frame size is adjusted according to the measured re-sults. In this way, channel resources can be more effectively utilized by sending opti-mal bits into one transmission frame according to the wireless channel conditions. As expected, simulation results show that the proposed VMFS scheme achieves a much higher throughput than a non-frame size adaptive MAC protocol in LR WPAN does.

References

1. Institute of Electrical and Electronic Engineers : Standard for Part 15.4: Wireless Medium Access Control Layer (MAC) and Physical Layer (PHY) Specifications for Low Rate Wire-less Personal Area Networks (LR-WPANs), 802.15.4. (2003)
2. Ed Callaway et al. : Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks, IEEE Commun. Mag., Vol. 40, No. 8. (2002) 70-77
3. AJ Goldsmith and SB Wicker : Design challenges for energy-constrained Ad Hoc wireless networks. IEEE Wireless Commun. Mag. (2002) 8-27
4. Byung-Seo Kim, Yuguang Fang and Tan F. Wong : Rate-Adaptive MAC Protocol in High-Rate Personal Area Networks. Proc. of WCNC 2004, Vol. 2. (2004) 1395-1399
5. A. Franchi and R. A. Harris : On the error burst properties of Viterbi decoding. Proc. IEEE Int. Cont. Commun., Geneva, Switzerland (1993) 1086-1091
6. Pilsoon Choi, et al. : An experimental coin-sized radio for extremely low-power WPAN (IEEE 802.15.4) application at 2.4GHz. IEEE J. pf solid state circuits, Vol. 28, No.12. (2003)
7. R. L. Miller, L. J. Deutsch, and S. A. Butman : On the error statistics of Viterbi decoding and the performance of concatenated codes. JPL Publication 81-9, Jet Propulsion Labora-tory, Pasadena, CA (1981)
8. Hyun Lee, Sammo Jo, Bong ho Lee and Soo In Lee : Packet Error Rate Model for DAB system. THE KOREAN SOCIETY OF BROADCAST ENGINEERS conference (2002) 201-206
9. L. Thibanh and M. ThienLe : Performance Evaluation of COFDM for Digital Audio Broad-casting Part I: Parametric Study. IEEE Trans. On Broadcasting, Vol. 43. No 1. (1997) 64-75
10. J. Karaoguz : High-Rate Wireless Personal Area Networks. Commun. Mag. (2001) 96-102
11. R. J. Punnoose, P. V. Nikitin, and D. D. Stancil : Efficient simulation of ricean fading within a packet simulator. Proc. IEEE VTC'00 (2000) 764-767
12. G. G. M. Janssen and R. Prasad : Propagation measurements in indoor radio environments at 2.4 GHz, 4.75 GHz and 11.5 GHz. Proc. IEEE VTC'92 (1992) 617-620

FLEXOR: A Flexible Localization Scheme Based on RFID

Kuen-Liang Sue[1], Chung-Hsien Tsai[1], and Ming-Hua Lin[2]

[1] Department of Information Management, National Central University, 300, Jhongda Rd., Jhongli City, Taoyuan County 32001, Taiwan
klsue@mgt.ncu.edu.tw, 93423002@cc.ncu.edu.tw
[2] Department of Information Management, Shih Chien University, 70 Ta-Chih Street, Taipei 10462, Taiwan
mhlin@mail.usc.edu.tw

Abstract. There are many localization schemes used for the indoor or outdoor applications, such as the GPS, RADAR etc. However, the accuracy of indoor localization scheme is easy to be influenced because of the obstacles and the environment interference. LANDMARC which is proposed by Lionel et al. uses the RFID tags to reduce the influence of the interference. This paper proposes an improved localization scheme, FLEXOR, which divides the localization area into cells to reduce computational overhead and provide two localization modes: region mode and coordinate mode. In the performance evaluations, FLEXOR has been proved to have the advantages of fast localization, flexibility, and it also provides the high localization accuracy as LANDMARC.

1 Introduction

Recently, the localization of mobile target becomes a popular issue in wireless communication. The location information can be more useful as it combined with geographic information or road map, such as the applications of GPS (Global Positioning System)[1,2]. The combination of the location information and wireless network create great convenience in life.

However, GPS is not suitable for the indoor localization because it is based on outer space satellites. For that reason, some mechanisms such as RADAR, a positioning system based on radio waves, is developed for indoor localization[3]. Although it has the advantage of easily set up and less base station requirement, the accuracy of localization is still need to be improved since the signal strength of radio wave is usually interfered by the obstacles of environment.

The LANDMARC localization scheme which is proposed by Lionel et al. uses RFID tags as localization reference to reduce the interference from outside environment and improve the indoor localization accuracy[4]. In this paper, we propose a more flexible localization mechanism - FLEXOR (Flexible Localization EXplOits Rfid). The second part of this paper introduces related works and RFID briefly. The third part demonstrates the FLEXOR scheme and discusses the localization procedure into more detail. In the fourth part, we analysis and

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 306–316, 2006.

evaluate the proposed mechanism by simulations. The final part gives conclusions and future research.

2 Related Works

There are many techniques used for the indoor localization, such as IR, IEEE 802.11, and ultrasound[5,6]. However, because of the restrictions of these techniques, the indoor localization schemes based on them can not satisfy some requirements of localization services, such as localization accuracy or cost. Therefore, many people try to find other techniques for the indoor localization and study their feasibilities. In recent years, the RFID (Radio Frequency Identification) is more and more popular, and it also provides a new attempt at localization[7].

A RFID system which is composed of RFID readers and RFID tags is originally designed for data identification[8]. Because the price of RFID tag is reduced substantially, and the technology of RFID system is more mature, some people propose the indoor localization schemes which are based on the RFID technique, such as SpotON and LANDMARC.

The objective of SpotON is to design a new RFID tag which can measure the received signal strength and be used for further localization[9]. However, it relies on the design of hardware. The related researches still stay on the evaluation stage. LANDMARC use the existing RFID system for indoor localization instead of the design of new hardware[4]. The main idea of LANDMARC is using the readers to detect the signal of tags from power level 1 to power level 8 gradually. Each reference tag will be detected in some degree of power levels of the readers. The degrees of power levels which reference tags belong to are collected to find out which reference tags are near to the tracking tag. When k nearest reference tags was determined, LANDMARC computes the location of tracking tag as follows:

$$(x, y) = \sum_{i=1}^{k} W_i(x_i, y_i) \tag{1}$$

The (x,y) is the estimated coordinate of tracking tag; (x_i, y_i) is the coordinate of the ith tag of k nearest reference tags, and W_i is the weight of ith nearest reference tags. LANDMARC has two main advantages:

1. LANDMARC does not need to deploy many RFID readers. It uses RFID tags as references, and they are much cheaper than readers. So, LANDMARC is cost effective.
2. LANDMARC can reduce the localization errors caused by environmental interference by using dynamically adjusted power level degree.

However, LANDMARC is lack of flexibility as it used for some applications. In fact, we often need to known the roughly region instead of the coordinates. For example, the navigation system in the museum only needs to provide the region information to the consumers. Therefore, we propose an improved localization

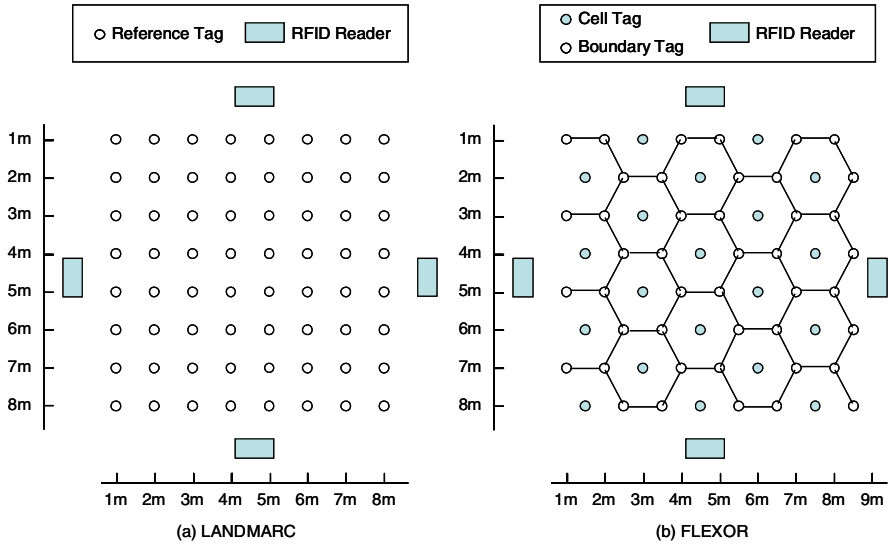


Fig. 1. Layout of (a) LANDMARC and (b) FLEXOR

schemew FLEXOR. It can reduce the computational load of localization by classifying the reference tags. Besides, FLEXOR has almost equivalent localization accuracy with LANDMARC, and it is also not easy to be influenced by the interferences. The detail of FLEXOR is demonstrated in the next section.

3 FLEXOR Localization Scheme

3.1 Layout of FLEXOR

In LANDMARC, the reference tag is arranged in quadrangle in the localization area, which is illustrated in Fig. 1(a). When locating according to the nearest four reference tags, LANDMARC gets the minimum localization error. Actually, the layout of the tags affects the result most. Therefore, we rearrange the layout of the location system in FLEXOR.

Firstly we divide the localization area into several hexagons, each hexagon called a cell. Moreover, the reference tag is separated into two categories: Cell Tag (CT) and Boundary Tag (BT).

Cell tags: The reference tags located in the center of each cell.

Boundary tags: The reference tags besides the cell tags, which lies in the boundary of each cellular region.

The localization area consists of cell tags and boundary tags. Each cell tag has six boundary tags around it in maximum. The target must equipped with a RFID tag, which is called tracking tag.

The main concept of FLEOR localization scheme is to find the reference tags nearest to the tracking tag, and further forecast the region of target. Therefore

we need the information of relative distance between tags to find the nearest tags. In FLEXOR, this is completed by simulating the power level setting of RFID readers.

In FLEXOR, we place several RFID readers around the localization area as shown in Fig. 1(b), and the detection range of each reader ranges from power level 1 to power level 8. During the locating period, the RFID readers proceed to detect the signals diffused by all tags and record the corresponding minimum power level which detects the signal of each tag. After finishing the signal detection of power level, the computation center then summarize the detected information for further localization.

3.2 Localization Modes

In many practical applications, we don't need to know the exact coordinates, but the region of the target. Take the supermarket application for example; the consumer only needs the information of what region he/she is and the product classification of every region. Therefore, the FLEXOR scheme provides two localization modes: region mode and coordinates mode.

3.2.1 Region Mode

In region mode, the information we need for localization is only cell tags. We can use the information of power level detected by RFID readers to find the cell tag nearest to the tracking tag, and set the cellular area which the cell tag locates as the region of tracking tag to provide region localization service.

Let there are k RFID readers in the localization area, m cell tags and n boundary tags. When the localization target along with the tracking tag enters the localization area, the RFID readers can detect the signal transmitted from the tracking tag in certain power level range. Therefore, we can define the signal vector of tracking tag as $\vec{S} = (S_1, S_2, ..., S_k)$,S_i represents the power level of ith RFID reader which first detects the signal of tracking tag, and $1 <= i <= k$. Meanwhile, for each cell tag j, we can define the signal vector of cell tag j as $\overrightarrow{\theta_j} = (\theta_{j1}, \theta_{j2}, ..., \theta_{jk}), \theta_{ji}$represents the power level of ith RFID reader which first detects the signal of cell tag j.

Therefore, computation center can calculate the relative distance between each cell tag and the tracking tag as follows:

$$E_j = \sqrt{\sum_{i=1}^{k} (\theta_{ji} - S_i)^2} \tag{2}$$

Among the formula (2), E_j is the distance between tracking tag and the jth cell tag. When we find the nearest cell tag according to formula (2), we can set the located cell of the nearest cell tag as the region of tracking tag.

From the description above, we find the computation load of region mode can be greatly reduced compared to LANDMARC. The main reason is that the data of tags needed to be calculated in region mode of FLEXOR is far less than LANDMARC. Therefore, the reduction of computational load helps FLEXOR to provide faster localization service.

3.2.2 Coordinates Mode

FLEXOR provides an accurate coordinates localization mode when we need to know the coordinate of tracking tag. It continues with the procedure of region mode and further use the data of boundary tags to estimate the coordinate. The coordinates mode of FLEXOR includes the following steps:

Step1: Like the region mode, FLEXOR finds the cell tag nearest to the tracking tag from all cell tags.

Step2: Each cell can find out the surrounded six boundary tags from the pre-arranged cell tag and boundary tags. Step 2 is to find the boundary tag, l, nearest to the tracking tag from six boundary tags around the region; then, from the two boundary tags adjacent to the nearest boundary tag l, find next one nearest to the tracking tag.

Step3: We can get the coordinate information of three reference tags from Step 1 and Step 2; it includes one cell tag and two adjacent boundary tags. Further, by using 3-nearest neighbor algorithm we can calculate the coordinate (x,y) of tracking tag as follows:

$$(x, y) = \sum_{i=1}^{3} W_i(x_i, y_i) \tag{3}$$

Formula (3) gives weight (W_i) to the coordinates of three localization reference tags (x_i, y_i), and sums up the weighted coordinates to calculate the coordinates of tracking tag (x,y). The weight is determined by considering the distance between reference tag and tracking tag, which is illustrated in formula (4):

$$W_i = \frac{\frac{1}{E_i^2}}{\sum_{i=1}^{3} \frac{1}{E_i^2}} \tag{4}$$

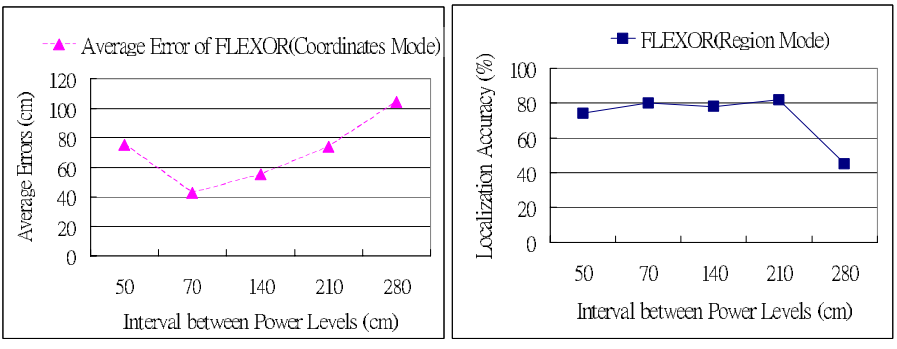
E_i in formula (4) represents the distance between the ith reference tag and tracking tag. Though there are more steps in coordinates mode, the calculation is still greatly reduced compared to LANDMARC. It only needs the data of all cell tags and six boundary tags to complete the calculation, but the localization accuracy can meet the requirements of several practical applications.

4 Performance Evaluations

4.1 Simulation Configurations

We use simulation tool written in JAVA to evaluate the localization error, computational time, and regional estimation accuracy of FLEXOR. The RFID tag layout in FLEXOR is deployed into the shape of cells (hexagons), while it is the shape of squares in LANDMARC. As for the number of tags, the simulations are executed with the different number of 16 tags and 64 tags which covers $9m^2$ and$49m^2$regions respectively. That is to say that there are four tags in region of one square meter. In the comparison of FLEXOR and LANDMARC, the coverage of localization area is the same.

In the simulation, we observe the effect of certain factors on the FLEXOR; those factors include the interval between power levels, the layout of the readers, and the number of reference tags. The evaluation of localization accuracy is based on the average error which is calculated from 30,000 localization simulation results.



(a) Coordinates mode (b) Region mode

Fig. 2. The effect of interval between power levels to FLEXOR

4.2 Effect of Interval Between Power Levels

FLEXOR finds the nearest tag to the tracking tag by referring the degree of power levels of detected tags. In other words, the 8 degrees of power levels divide the localization area into several sub-areas. And we can determine relative distance between reference tags and tracking tag by using the detected degrees of power levels. Therefore, the interval between power levels affects the localization accuracy to a certain extent. In the simulations, we assume that the interval between power levels is fixed. For example, if the interval between power levels is 1 meter, the detection range from power level 1 to 8 is 1 meter to 8 meters.

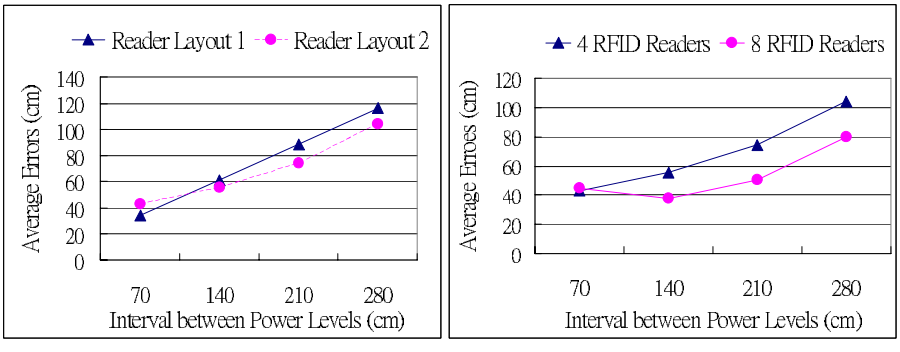
We evaluated the localization accuracy of FLEXOR under different intervals between power levels. The simulation configurations are as follows: There are 64 reference tags deployed in the localization area which is spread in 8 square meters. And there are 4 readers deployed around the localization area. The distance between each reader and the boundary of the localization area is different. We observed the variation of the average localization error under the intervals of power level from 50cm to 280cm. The simulation result is illustrated below:

When the interval between power levels is equal to 70 centimeters, the coordinates mode of FLEXOR has the smallest localization error, as shown in Fig. 2(a). While the interval between power levels increases, the localization error will also increase. When the interval between power levels is small, the localization area can be divided into more sub-areas. This makes relative distance between tags can be determined more correctly. Hence, the localization error can be reduced while the interval was smaller.

But, when the interval between power levels is equal to 50 centimeters, the localization error is increased. The reason of the contrary result is related to the coverage

of detection range. When the interval between power levels is equal to 50 centimeters, the maximum detection range of readers will be 4 meters. That is to say that there are nearly 50% of the reference tags are outside the detection range of readers. Therefore, the low coverage of detection range increases the localization error.

In Fig. 2(b), we find that the variation of interval between power levels from 50 centimeters to 210 centimeters causes little effect to the localization accuracy. Because the region mode uses only cell tags to locate the tracking tag, and the distance between cell tags is longer than that between any adjacent tags. Hence, the increasing of interval causes little effect to localization accuracy if the interval was not too long.



(a) Effect of reader layout (b) Effect of number of readers

Fig. 3. The effects of readers in coordinates mode of FLEXOR

Because the longer the interval is, the more cell tags is covered in the same detection range of power levels. And that makes the relative distance between cell tags difficult to determine. According to the analysis above, we find that if the interval between power levels is shorter and the coverage of detection range is still wide enough, the FLEXOR will have better precision.

4.3 Effect of RFID Readers

As mentioned in the previous analysis, if the different detection ranges of the readers can divide the localization area into more sub-areas, the distance between tags will be determined more correctly to provide precise localization. The factors that influence the division of localization area may include the layout and the number of the readers. In the following simulations, we observe the influences of the different layout and the different number of readers.

4.3.1 Distance of Reader Layout

We construct two layouts of the readers in the simulations.

Layout 1: There are 4 readers deployed around the localization area (the north, east, south, and west side respectively). The distance between each reader and the localization area is the same.

Layout 2: There are also 4 readers around the localization area. But the distance between every reader and localization area is different. The distances from the boundary of localization area to the readers are 0, 1, 2, and 3 meters respectively.

Similarly, we observe the variation of localization accuracy of FELXOR under two layouts of readers. The comparison is illustrated in Fig. 3(a). In general, the average errors of layout 2 are smaller than that of layout 1. Only when the interval between power levels was equal to 70 centimeters the layout 1 has smaller average error than layout 2. When the interval was small, the detection range of readers can still divide the localization area equally, even the readers are deployed as layout 1. But when the interval increases, the average error of layout 1 will increase greater than that of layout 2. The detection ranges of layout 2 can divide the localization area symmetrically, so the localization error affected by the variation of interval does not increase significant. Therefore, the FLEXOR has more precise accuracy if the distances between readers and the localization area are not the same.

4.3.2 Number of Readers

We also evaluate the influence of the number of readers on the accuracy of FLEXOR. The layout of the readers adopts layout 2 mention in the previous section. We deploy different number of readers around the localization area. In the simulation of 4 readers, there is one reader deployed in each side of the localization area. While in the simulation of 8 readers, there are two readers deployed in each side.

When the number of readers increases, the localization accuracy will be improved. Because the number of the degrees of power level in which the tag is detected by the readers (as θ_{ji} in formula (2)) is increased, the relative distance between tags can be determined precisely. Therefore, the nearest reference tags used for localization can be chosen appropriately. The Fig. 3(b) shows the simulation results with different number of readers.

4.3.3 Positions of Readers

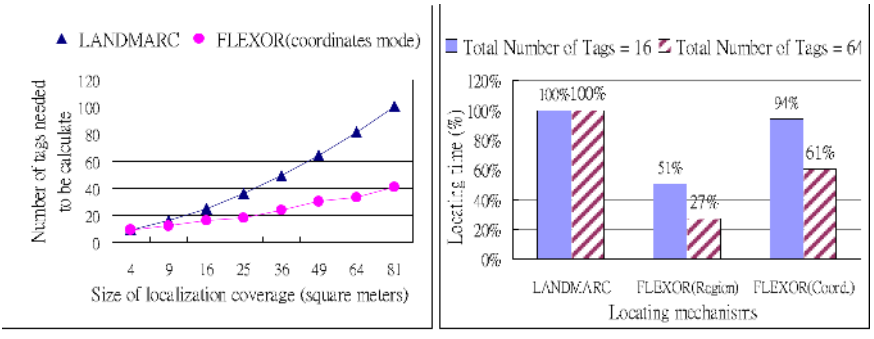
In this section, we evaluate the effect of the position of the readers to the FLEXOR. We compare the accuracy of simulation results which are executed with 4 readers of different position.

Table 1. The comparison of localization accuracy under different positions of readers

Reader positions	Comparison of localization accuracy
4 tags around the localization area	Better
2 tags in both north and south sides	Worse
2 tags in both north and east sides	Worse

From the comparison results shown in Table 1, we find that when 4 readers are distributed over each side of localization area, the FLEXOR will have better localization precision. If the readers deployed only in several sides of localization area, the localization accuracy will reduced.

The simulation results provide the important considerations for the implementation of FLEXOR. In principle, while the interval between power levels is smaller, FLEXOR will have higher accuracy. But if the interval of power levels is too small to cover the most part of localization area, the localization error will increase. That is because most tags will be outside the detection ranges of the readers.



(a) Analysis in computational loa (b) Analysis in localization time

Fig. 4. Comparison of efficiency for LANDMARC and FLEXOR

In the aspect of the readers, when the distances between the readers and localization area are different, and the positions of readers are distributedly deployed around the localization area, and more readers are used, the FLEXOR will has better precision.

4.4 Comparison of Computational Load

The FLEXOR also decreases the computation load for localization. In LANDMARC, the degrees of power levels of all tags are needed to be gathered for localization. While in FLEXOR, it mainly concerns about the degrees of power levels of cell tags. FLEXOR classifies the reference tags into cell tags and boundary tags, and constructs the topology of tags into shape of cells. It only needs to take the information of all cell tags and 6 boundary tags into calculation for the localization service If the number of total reference tags and the size of localization area were the same, the data of tags need to be calculated of FLEXOR is fewer than that of LANDMARC, as illustrate in Fig. 4(a). For this reason, the FLEXOR saves more time for localization as the coverage of localization area was wider. The simulation result of localization time is shown in Fig. 4(b). The advantage of faster localization is more important for many location-based services which emphasize the quality of service, such as response time.

4.5 Comparison of Localization Accuracy

In the paper of LANDMARC the accuracy has been verified to be superior to all existing locating mechanisms. FLEXOR can reach the same accuracy with

LANDMARC. We also compare the locating accuracy between FLEXOR and LANDMARC, and make comparison based on average locating errors.

The stimulation environment includes 64 tags, and spread in 49 square meters area. Each side of the localization area equipped with 2 RFID readers; totally there are eight readers and each of them is set with different distance to localization area. The simulation results show that they have almost the same localization error.

5 Conclusion

In this paper, we present an indoor localization mechanism w FLEXOR, which is based on RFID. FLEXOR improves the LANDMARC by dividing the localization area into many cell regions and provides the following advantages:

Flexible localization service: FLEXOR provides two kinds of localization modes including region mode and coordinates mode. According to the different requirement of application, FLEXOR can be used with flexibility and satisfy various situations of localization services.

Reduce the computational load and speed up the localization service: FLEXOR classifies the reference tag of LANDMARC into cell tag and boundary tag. When locating the target, it only needs to take the data of all cell tags and six boundary tags into calculation. Comparing with LANDMARC, FLEXOR can save a lot of time in calculating the formula (2). Therefore, FLEXOR can provide localization service in shorter time. This advantage is more important for many application services which emphasize on the response time of QoS.

Provide high localization accuracy: In the simulations, the coordinates mode of FLEXOR has high localization accuracy equivalent to LANDMARC. This advantages can satisfy some m-services which require the precise localization.

In the future studies, we will focus on the improvement of the localization accuracy of region mode. And we expect that the accuracy of the coordinates mode can also be improved by this way.

Acknowledgment

The research was supported by the National Science Council, Taiwan, R.O.C., under the contract NSC94-2416-H-008-014.

References

1. Cecchini, J, "Next generation GPS-aided space navigation systems", Aerospace and Electronic Systems Magazine, IEEE, vol. 17, no. 12, pp.7-10, 2002.
2. Bajaj R., Ranaweera S.L. and Agrawal D.P., "GPS: location-tracking technology", Computer, vol. 35, no. 4, pp.92-94, 2002.
3. Bahl P.and Padmanabhan V.N., "RADAR: An in-building RF-based user location and tracking system", Proceedings of IEEE INFOCOM, vol. 2, pp.775-784, 2000.

4. Ni L.M., Yunhao Liu, Yiu Cho Lau and Patil A.P, "LANDMARC: indoor location sensing using active RFID", Pervasive Computing and Communications, pp.407-415, 2003.
5. Want R., Hopper A., "Active badges and personal interactive computing objects", Consumer Electronics, IEEE Transactions on, vol. 38, no. 1, pp.10-20, 1992.
6. Fukuju Y., Minami M., Morikawa H. and Aoyama T, "DOLPHIN: an autonomous indoor positioning system in ubiquitous computing environment", Software Technologies for Future Embedded Systems, pp.53-56, 2003.
7. Hahnel D., Burgard W., Fox D., Fishkin K. and Philipose M., "Mapping and localization with RFID technology", Robitics and Automation, IEEE International Conference on, vol. 1, pp.1015-1020, 2004.
8. Philipose M., Smith J.R., Jiang B., Mamishev A., Sumit Roy and Sundara-Rajan K., "Battery-free wireless identification and sensing", Pervasive Computing, IEEE vol. 4, no. 1, pp.37-45, 2005.
9. Hightower J., Vakili C., Borriello G. and Want R., "Design and Calibration of the SpotON Ad-Hoc Location Sensing System", CSE, University of Washington, Seattle, WA, pp.1-18, August 2001. http://portolano.cs.washington.edu/projects/spoton/

Security Enhancement Mechanism for Ad-Hoc OLSR Protocol*

Inshil Doh[1], Kijoon Chae[1], Howon Kim[2], and Kyoil Chung[2]

[1] Dept. of Computer Science and Engineering, Ewha Womans University, Korea
isdoh@ewhain.net, kjchae@ewha.ac.kr
[2] Electronics and Telecommunications Research Institute, Korea
khw@etri.re.kr, kyoil@etri.re.kr

Abstract. In this paper, we propose security mechanism for Ad-hoc OLSR protocol. We consider the security vulnerabilities for OLSR and apply one-time digital signature for lightweight authentication of the routing messages. We also propose compromised node management mechanism in which normal nodes collaborate to report malfunctioning nodes and to isolate them from network. We finally analyze the security and overhead of our mechanism.

1 Introduction

Mobile Ad-hoc network is a collection of mobile nodes which communicate with each other and provide routing function without any fixed infrastructure. Many routing protocols have been proposed for efficient communication and their main function is to discover the dynamic network topology. Ad-hoc routing protocols are classified in two groups. One of them is on-demand, i.e. reactive protocol, such as AODV[1] or DSR[2] in which routing paths are discovered when communication is needed. The other one is table-driven or proactive protocol in which routing tables for recent network topology are managed by periodic routing message exchanges. Recently, OLSR(Optimized Link State Routing)[3] and TBRPF(Topology dissemination Based on Reverse-Path Forwarding)[4] were adopted as RFCs for ad-hoc proactive routing protocols. However, security researches of proactive protocols are deficient compared with those of on-demand fashion. Several security mechanisms have been proposed, but they are based on PKI or symmetric keys and can be costly. In this paper, we propose security mechanism which can substantially reduce the overhead of using digital signature. Our approach applies one-time digital signature for authenticating the routing messages. It also detects not only external attacks but also internal attacks and isolates the compromised nodes and enhances network routing security efficiently.

* This research was partially supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) and supported by ETRI(Electronics and Telecommunications Research Institute), Korea.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 317–326, 2006.

The rest of this paper is organized as follows. In section 2, we present the motivation of our security mechanism. Previous work is briefly discussed in section 3. In section 4, we present the assumptions and one-time digital signatures, and we describe our proposed security mechanism in Section 5. Section 6 analyzes our proposed mechanism, and section 7 concludes our work and points out some future research directions.

2 Motivation

2.1 OLSR Protocol

The Optimized Link State Routing protocol(OLSR)[3] is a proactive link state routing protocol for mobile ad hoc networks. It operates as a table driven method, and every node exchanges topology information with other nodes of the network regularly. Each node selects a set of its neighbor nodes as "multipoint relays" (MPR). In OLSR, only nodes, selected as such MPRs, are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required. Control traffic in OLSR is exchanged through two different types of messages: HELLO and TC messages. HELLO messages are emitted periodically by a node and contain a list of neighbors from which control traffic has been heard, a list of neighbors with which bidirectional communication has been established, and a list of neighbors that have been selected to act as a Multipoint Relay for the originator the HELLO message. Upon receiving a HELLO message, a node examines the lists of addresses. If its own address is included in the addresses, bi-directional communication is possible between the originator and the recipient of the HELLO message. In addition to information about neighbor nodes, periodic exchange of HELLO messages allows each node to maintain information describing the links between its neighbor nodes and nodes which are two hops away. This information is recorded in a nodes 2-hop neighbor set and is utilized for MPR optimization. Link state between two nodes changes step by step, which means symmetric-neighbor or MPR-neighbor state cannot jump from asymmetric link or no link state. This idea is used for our malfunctioning node detection mechanism. Notations for link state are as follows.

- ASYM-LINK : an asymmetrical link
- SYM-LINK : a symmetrical link
- SYM-NEIGH : the node is a symmetric neighbor
- MPR-NEIGH : the node has been selected as an MPR by the sender

TC messages, just like HELLO messages, are emitted periodically. TC message contains a set of bi-directional links between a node and some of its neighbors. TC messages are flooded to the entire network, exploiting the MPR optimization. Only nodes which have been selected as an MPR generate TC messages. In route calculation, the MPRs are used to form the route from a given node to any destination in the network.

2.2 Security Vulnerabilities of OLSR

A misbehaving node can send control messages while pretending to be another node. Hello or TC message from a spoofed originator address results in conflicting routes to a node or causes incorrect links to be advertised in the network, respectively. Especially, because OLSR optimizes flooding through MPR nodes and forms tree-like structure, when an MPR node is compromised, quite a large area is affected as in Fig. 1. Incorrect control messages can be broadcasted by this compromised nodes with legitimate signatures. This kind of internal attacks are more serious and tricky.

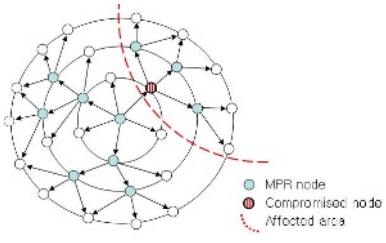


Fig. 1. Security vulnerabilities of OLSR Protocol

3 Previous Work

An important issue for securing ad-hoc network is the integrity of routing information. If some of the nodes are compromised and malfunctioning in routing process, the integrity of the network fails. Several researches have been proposed for reactive routing protocols, such as AODV and DSR. Ariadne[5] uses TESLA[6] for authenticating routing information, and SAODV[7] applies digital signature for routing messages. SRP[8] assumes SA(Security Association) between sender and receiver and adds MAC to the routing messages. ARAN[9] uses digital signature and every node is authenticated by next node. For proactive routing protocols, DSDV[14] and SEAD[10] use hash chains for message authentication. And for OLSR, PKI-based authentication mechanism[15] and symmetric key-based authentication mechanism[16] are proposed. Both apply digital signature for integrity of routing information. Most of the researches can block external attack which can be detected easily by applying signature mechanism. However, more serious and subtle attack to detect is the internal attack which is done by compromised nodes with legitimate keys.

4 Assumptions and One Time Digital Signatures

4.1 Assumptions

– PKI-based key pairs are available for every node. However, in order to reduce the high computational costs, the PKI is not used to sign routing messages directly, but used when broadcasting public key components of hash table.

- Colluding attack by neighboring nodes is not considered.
- There is no compromised node at the beginning stage when nodes are deployed and find their neighbors for the first time.
- Timestamp is deployed to defend replay attack, but the timestamp setup process is not considered in this work.

4.2 One-Time Digital Signature

One-time signature scheme is based on a public function f that is easy to compute but computationally infeasible to invert. The scheme was first introduced by Lamport[11], and Merkle[12] proposed an improvement in which the signer can generate only one x and one y for each bit of the message to be signed under a one way function $y = f(x)$. These x's and y's are called secret key components and public key components, respectively. When one of the bits in the message to be signed is '1', the signer releases the corresponding value of x; but when the bit to be signed is '0', the signer releases nothing. Because this allows the receiver to pretend that he did not receive some of the x's, and therefore to pretend that some of the '1' bits in the signed message were '0', the signer must also sign count of the '0' bits in the message. Now, when the receiver pretends that a '1' bit was actually a '0' bit, he must also increase the value of the count field, which can't be done. Because the count field has only $\log_2 n$ bits in it, the signature size is decreased by almost a factor of two, i.e. from $2n$ to $n+\lfloor \log_2 n \rfloor +1$.

5 Security Enhancement Mechanism

5.1 Routing Message Authentication

In this subsection, we propose OLSR security mechanism applying one-time digital signature utilizing one-way hash function for secure efficient routing. In our mechanism, hash tables for one-time digital signature are generated before routing protocol process. After the hash table is created, each node broadcasts the one-time public key components signed by public key cryptography. During the routing process, all the control packets are signed by one-time digital signature. When a node uses up the hash table components, it regenerates a new table and broadcasts one-time public key component to the entire network with PKI signature.

Let Mi be the íth routing message to be sent and two hash functions f and h are known to all nodes. we applied MD5[13] for getting hashed information of routing messages. Hash function f is applied to the message Mi to obtain its hash $f(Mi)$. This hash value $f(Mi)$ is to be signed to provide authenticity and integrity of message Mi. Suppose the output of hash function f is l-bit long. Using Merkle's scheme, we need $n(= l + \lfloor \log_2 l \rfloor + 1)$ one-time public key components to sign $f(Mi)$. In order to sign more than one message, we need multiple sets of these one-time public key components. We derive multiple sets of public key components from hash chains by repeated hashing of the public key components in the first set to decrease the storage overhead. Fig. 2 shows signing and verifying process using one-time digital signature.

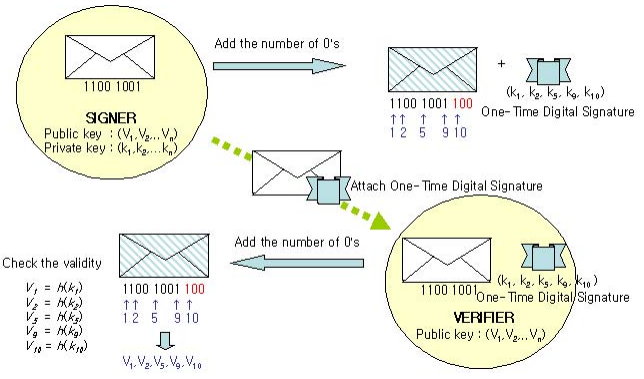


Fig. 2. Signing and verification of message with one-time digital signature

Hash Table creation. Hash table creation process is as follows. we adopted MD5 for generating hash chain and hash tables.
1. Each node randomly chooses private key components x_j when $j = 1,..., n$.

$$n = l + \lfloor \log_2 l \rfloor + 1, \; l : \text{length of } f(Mi)$$

2. Each node creates a table of n hash chains of length k as in Fig. 3. k is hash table length.
3. Each node broadcasts k'th row signed using PKI private key.
4. When receiving this message, every node verifies the message from other nodes using public-key cryptography and stores them as v_j, $j = 1, ..., n$. These v_js are the one-time public key components of the corresponding node.

0	$h^0(x_1)$	$h^0(x_2)$	...	$h^0(x_n)$
1	$h^1(x_1)$	$h^1(x_2)$	...	$h^1(x_n)$
...	...	...	...	...
k	$h^k(x_1)$	$h^k(x_2)$	...	$h^k(x_n)$

Fig. 3. Hash Table generated by each node

Signing the control messages. To detect identity spoofing, TC and Hello messages are transformed into one-time signature added messages.
1. $f(Mi)$ is concatenated with a count field of 0's using Merkle's method.
2. One-time digital signature is attached to $f(Mi)$. One-time digital signature is the hash values in the $(k-i)$th row of the hash table where the bit-value is 1.
3. After each control message is signed, the source node attaches global one-time signature in the same manner.

Verifying the control message. When a node receives signed control message, it verifies the message.
1. It first obtains the bit string g by concatenating $f(Mi)$ with counter field, using Merkle's method.

2. For all j such that g_j =1, it checks if one-time digital signature is correct by repeated hashing of received signature values.
3. If the global signature is correct, it repeats step 2 for every control message.
4. If respective control message is true, it updates its own information.

5.2 Compromised Node Attack Detection

Identity spoofing can be detected easily by signing and verifying the control message. But when nodes are compromised and they forge control message with correct signature, it is not easy to detect the attack. We propose internal attack detection and reporting mechanism using state transition information.

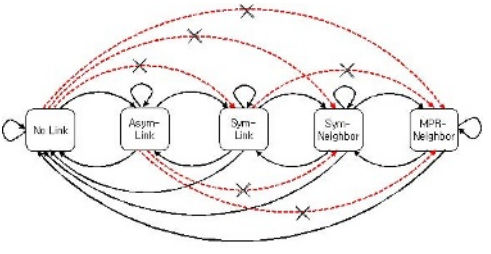


Fig. 4. Link State Transition Diagram between two nodes

Attack detection: When a node wants to notify symmetric link or symmetric neighbor with its neighbors, it needs to pass through previous states as in Fig. 4. Attacks can be made in a way that compromised node notifies non-neighboring node as neighboring node, and vice versa. In the latter, the attacker does not damage network seriously because, in that case, the neighboring nodes cannot chosen as an MPR even if it has enough links to neighbor nodes. Network performance can be just degraded in some degree. However, the former attack can disrupt network routing and should be detected. For detection, every node keeps recent state with its neighbor nodes and if incorrect link state information with itself is broadcasted, it recognizes the neighbor node malfunctioning. When an MPR node generates and broadcasts TC messages, the neighboring nodes operate in the same way.

Attack report: In OLSR, nodes use 2-hop neighbor information for MPR selection, and when a node detects malfunctioning node, it broadcasts a report message with signature in every 2-hop neighbors. If more than one node detect an attack, there will be several mal-report messages broadcasted in local area. The node first reporting the attack gathers the reports and generates one mal-report message to decrease traffic overhead. In the mal-report, there are as many local reports with respective one-time digital signature as the number of reporting nodes and one global one time digital signature signed by the initiator. When only one node reports malfunctioning node, there will be one signature in the mal-report packet. Fig. 5 shows local reporting by 3 neighbor nodes near the malfunctioning node. When reporting malfunctioning node, pure broadcast is used to prevent malicious node from blocking reporting packet.

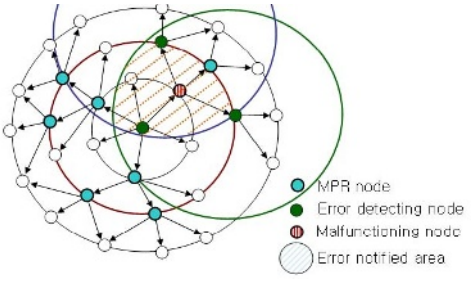


Fig. 5. Local reporting by three neighbor nodes

Attack prevention: When normal nodes receive mal-report, they verify the packet first, and when the verification is true, they decrease the credit of reported node. When credit of a node is under a threshold, the node is classified as malicious node and cannot be selected as an MPR and eventually isolated from the network. Usually, when the malicious node tries to disrupt the routing, it is reported by many normal nodes and isolated. When a malicious node falsely reports a normal node, the normal node also makes a mal-report. And in the worst case, if a malicious node attacks just one normal node repeatedly, both the malicious node and the normal node can be excluded from the network. But even in this case, the malicious node cannot damage the network seriously.

6 Evaluation

In this section, we evaluate security aspects of our mechanism and analyze the overhead.

6.1 Security Analysis

Identity spoofing. Identity spoofing can be efficiently detected by using one-time digital signature. Because every control message is signed respectively, every other node can verify by repeated hashing of the key components of the source node.

Incorrect link state notifying. Incorrect link state notification can be efficiently detected by proposed mechanism.

- Notifying false MPR nodes : Through attack detection, report, prevention process, wrong notification can be filtered.
- No reply : When compromised node hears the signal from its neighbors and doesn't reply, the node can listen several messages passing by, but cannot disrupt routing.
- Compromised nodes not getting the previous state message, but broadcasting the next state by guessing : Compromised node is not detected at that moment if its neighbor node had broadcasted the previous state. However in later phase, it will be detected when it tries to make another attack and be isolated from the network.

TC message updating. TC message is digitally signed and cannot be updated during flooding process.

Reporting normal nodes as malicious nodes. Falsely reporting node is also reported. In the worst case, real malicious node can isolate just one normal node, but malicious node itself is also excluded.

6.2 Overhead Analysis

Time overhead. For signature generation, our mechanism needs, on the average, 1.75ms when applying MD5 on a Pentium 4(2.4GHz) PC running Red Hat Linux 9.0 with 512MB RAM. For verification, it needs about 0.1ms. This is less than one-tenth of the simplest PKI-based digital signature generation and verification. We need PKI-based computation only when new hash table is generated and one-time public key components of the table have to be broadcasted throughout the network. This decreases the time overhead considerably.

Additional time overhead needed is for generation of new hash table when all the key components are used up. The time for new table generation depends on the length of the table, and with the similar computer specification, we can compute millions of MD5 hashing per second. It means that table creation does not take much time for each node when notebooks with regular specification are used.

- Time for signing and verifying control packets = (number of control messages +1) x generation time for one-time digital signature
- Time for signing and verifying mal-report packets = (number of reporting messages from neighborhood +1) x generation time for one-time digital signature

Storage overhead. The storage overhead for one-time public key components is $(l + \lfloor \log_2 l \rfloor + 1)$ x m bits for each node where l and m are the output length of hash function f and h, respectively. When we apply MD5, it requires 2KB and each node has $2k$KB storage where k is the length of hash table. The length of the hash table can be properly determined considering the number and the movement of nodes. If compared with the symmetric key based digital signature, our proposal has greater advantage. In symmetric key based approach, every node has to have $n(n-1)/2$ keys and the overhead rises as the number of nodes increases. Our mechanism does not require memory as much as symmetric key based mechanisms do even when the number of nodes drastically increases. Additional storage overhead needed for a node is the storage for keeping the final link state with its neighbors and for credits of other nodes, and this can be neglected.

Computation overhead. In our mechanism, PKI computation is only needed when hash table is newly generated and public-key components need to be broadcasted. One-time digital signature has much less computational complexity than PKI-based key computation and decreases computation overhead drastically. In addition, using the characteristics of hash function, receiver node can further

decrease hashing by keeping recently delivered signature values. When detecting malfunctioning node, reporting node needs to compute additional one-time digital signature and the overhead will increase if more nodes are getting compromised. However, it is worth applying because malicious nodes can be controlled efficiently through our mechanism.

Traffic Overhead. Our mechanism has additional traffic for reporting malfunctioning nodes and the traffic overhead rises as malicious nodes increases. However, the initiating node collects the reports from neighborhood and broadcasts one mal-report, and the additional traffic volume is controlled.

7 Conclusion and Future Work

To enhance the security of ad-hoc OLSR protocol, we applied one-time digital signature for authenticating control packets. Our mechanism can decrease time, storage, computation overhead considerably. In our mechanism, we also propose detecting internal attacks by compromised nodes which have legitimate keys and generate false information. When detecting malicious nodes, neighbor nodes collaborate to report to the entire network and eventually isolate the nodes from routing. For future research, we would like to consider other possible attacks including malicious neighbor nodes colluding attack, and to research efficient managing schemes for those attacks.

References

1. C.Perkins, E. Belding-Royer, Ad hoc on-demand distance vector (AODV) routing, July 2003. RFC 3561, Experimental.
2. David B. Johnson, David A. Maltz, and Yih-Chun Hu, The dynamic source routing protocol for mobile ad hoc networks (DSR), February 24 2003. In ternet-Draft, draft-ietf-manet-dsr-08.txt.
3. T. Clausen (ed) and P.Jacquet (ed). Optimized link state routing protocol (OLSR), October 2003. RFC 3626, Experimental.
4. R. Ogier, F.Templin, and M.Lewis, Topology dissemination based on reverse-path forwarding (TBRPF), February 2004. RFC 3684, Experimental.
5. Yih-Chun Hu, Adrian Perrig, and David B.Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proc. the 8th ACM International Conference on Mobile Computing and Networking, September 2002.
6. A. Perrig et al., Efficient Authentication and Signing of Multicast Streams over Lossy Channels, In Proc. IEEE Symp. Security and Privacy, IEEE Press, 2000, pp. 56-73.
7. Manel Guerrero Zapata, Secure ad hoc on-demand distance vector (SAODV) routing, October 2002. In ternet-Draft, draft-guerrero-manet-saodv-00.txt.
8. P. Papadimitratos and Z. J. Haas, Secure Routing for Mobile ad-hoc Networks, SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January pp. 27-31, 2002.

9. Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M.B elding-Royer, A secure routing protocol for ad hoc networks. In Proceedings of the 10th IEEE International Conference on Network Protocols, pages 78-89. IEEE Computer Society, 2002.
10. Yih-Chun Hu, David B.Johnson, and Adrian Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), Calicoon, NY, USA, June 2002.
11. L. Lamport, Constructing digital signatures from oneway function, Technical Report SRI-CSL-98, SRI International, October 1979.
12. R.C. Merkle, A Digital Signature Based on a Conventional Encryption Function, Proc. CRYPTO87, LNCS 293, Springer Verlag, 1987, pp 369-378.
13. R.L. Rivest, The MD5 Message Digest Algorithm, RFC1321, Apr 1992.
14. C. E. Perkins and P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, In Proc. SIGCOMM 94 Conf. Communications Architectures, Protocols and Applications, ACM Press, 1994, pp. 234-244.
15. D.Raffo, C. Adjih, T. Clausen, P.Muhlethaler, An Advanced Signature System for OLSR, Proc. SASN'04 October, 2004.
16. A. Hafslund, A. Tonnesen, R.B. Rotvik, J. Andersson and O. Kure, Secure Extension to the OLSR protocol, OLSR Interop and Workshop, 2004.

Mitigating Route Request Flooding Attacks in Mobile Ad Hoc Networks

Zhi Ang Eu[1,2] and Winston Khoon Guan Seah[1,2]

[1] School of Computing, National University of Singapore
euzhiang@comp.nus.edu.sg
[2] Networking Department, Institute for Infocomm Research, A*STAR
winston@i2r.a-star.edu.sg, winston@comp.nus.edu.sg

Abstract. A mobile ad hoc network is set up with a group of mobile wireless nodes without the use of any dedicated routers or base stations. Each node acts as an end node as well as a router for other nodes. There are generally two types of ad hoc routing protocols, reactive and proactive routing protocols. The focus of this paper centers on reactive routing protocols which establish routes between communicating nodes when needed using a route discovery process involving Route Requests and Route Replies, a process which can be easily misused for denial-of-service attacks. In this paper, we will describe one such attack, the Route Request Flooding Attack (RRFA) targeted at reactive routing protocols used in mobile ad hoc networks. Then, we propose the Route Request Flooding Defence (RRFD) mechanism that is designed to reduce the impact of RRFA. Finally, we present simulation results to show the detrimental effects of RRFA and the effectiveness of RRFD.

1 Introduction

A mobile ad hoc network (MANET) consists of a group of mobile wireless nodes that allow data communications beyond direct radio transmission through the use of intermediate nodes that will help to forward data packets. In MANETs, there is no central entity to coordinate the operations of the network, therefore there are more security challenges as compared to wired networks. Due to the nature of the wireless medium, malicious nodes or trusted nodes infected by viruses or worms can disrupt the operations of ad hoc networks by injecting wrong routing information or forging data packets. In this paper, we aim to address the problem of malicious flooding of route requests and present our solution using a popular reactive MANET routing protocol, the Ad hoc On-Demand Distance Vector (AODV) [1], as an example.

This paper is organized as follows. Section 0 gives an introduction to the basic features of AODV and describes a particular denial-of-service attack against MANETs that use reactive routing protocols – Route Request Flooding Attack (RRFA). Section 0 gives related work on security for AODV and lists some of the existing measures that can reduce the impact of RRFA. Section 0 describes our solution, the Route Request Flooding Defence (RRFD) that aims to reduce the impact of RRFA. We present some simulation results in Section 0, which demonstrates the impact of RRFA and the performance improvement when RRFD is used. Section 0 concludes this paper.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 327 – 336, 2006.

2 AODV and RRFA

The Ad hoc On-Demand Distance Vector (AODV) is a reactive MANET routing protocol. If a node wants to send data packets to a destination that is not in its routing table, it will buffer the data packets and broadcast a Route Request (RREQ) into the network. The RREQ packet will be forwarded by other intermediate AODV nodes to the intended destination node. The destination node, upon receiving the RREQ, will send a Route Reply (RREP) on the reverse route back to the source node.

After a RREQ is sent, the node will wait for a period of time for the RREP, namely, NET_TRAVERSAL_TIME milliseconds. If a RREP is not received within that time period, another RREQ will be sent, and for every RREQ sent, the waiting time for the RREP would be doubled. If no RREP is received after a fixed number of attempts, the data packets from the buffer will be dropped. If more data packets are received from the application layer, a new route discovery process will be initiated. In this paper, for the purpose of analysis, we assume that expanding ring search is not being utilized. If expanding ring search is used, the impact on the network will be similar since the reduction in transmission radius of the RREQ is offset by the increased rate at which RREQs are broadcast.

The Route Request Flooding Attack (RRFA) is a denial-of-service attack which aims to flood the network with a large number of RREQs to non-existent destinations in the network. In this attack, the malicious node will generate a large number of RREQs, possibly in the region of hundreds or thousands of RREQs, into the network until the network is saturated with RREQs and unable to transmit data packets. RRFA can be classified into two types: breadth-RRFA and depth-RRFA.

In *breadth-RRFA*, an attacker would initiate route discoveries to a large number of unreachable destinations. There are two possible ways to implement this attack. The first method is from the application layer. If the maximum allowable number of RREQs sent per second (RREQ_RATELMIT) is set to a very large value, the attacker can send a large number of data packets (e.g. PING packets) continuously into the network layer, resulting in a large number of RREQs originating from the node. In the second method, the attacker can modify the routing protocol or use malicious code to send out large numbers of RREQs to unreachable destinations. In *depth-RRFA*, the attacker will send out a large number of RREQs repeatedly. Only one unreachable destination is needed as compared to many for breadth-RRFA.

Reactive routing protocols generally require less routing overhead than proactive routing protocols because routes need not be maintained through periodic route updates even when there is no data traffic. Proactive routing protocols are not affected by RRFA as data packets are dropped at the source or any intermediate nodes if the destinations are not found. However, RRFA can seriously degrade the performance of reactive routing protocols and affect a node in the following ways:

1. The buffer used by the routing protocol may overflow since a reactive protocol has to buffer data packets during the route discovery process. Furthermore, if a large number of data packets originating from the application layer are actually unreachable, genuine data packets in the buffer may be replaced by these unreachable data packets, depending on the buffer management scheme used.
2. Depending on the design of the wireless interface, the buffer used by the wireless network interface card may overflow due to the large number of RREQs to be sent.

Similarly, genuine data packets may be dropped if routing packets have priority over data packets.

3. Since RREQ packets are broadcast into the entire network, the increased number of RREQ packets in the network results in more MAC layer collisions and consequently, congestion in the network as well as delays for the data packets. Higher level protocols like TCP which is sensitive to round trip times and congestion in the network will be affected.
4. Since MANET nodes are likely to be power and bandwidth constrained, RRFA can reduce the lifetime of the network through useless RREQ transmissions as well as additional overheads of authenticating a large number of RREQs, if used.

3 Related Work

The salient features of MANETs make them extremely vulnerable to malicious node behaviour resulting in performance degradation [2]. A summary of some security measures for MANETs in general is provided in [3] while security problems in AODV and specific attacks against AODV are addressed in [4] and [5]. A secure AODV that makes use of digital signatures and hash chains to protect routing packets is described in [6]. Intrusion detection systems described in [7], [8] and [9] are able to detect RRFA since there would be a significant increase in the number of RREQ packets. Once nodes are detected to be malicious, they are usually excluded from the network. However, this does not work well against worms or viruses that are able to spread through the network to infect other nodes. We do not propose to exclude nodes from the network once they are detected to be malicious. Instead, we try to reduce their negative impact on the network since they may still be willing to forward data packets. We now examine five current approaches used against RRFA:

1. Address filtering for RREQs. If the nodes know the IP addresses of all the nodes in the network, they can drop RREQs for destinations that are not in the network. However, this approach requires the nodes to have global knowledge of the network. Since nodes may fail, leave or join the network at any time, it is difficult to know all the valid IP addresses of the nodes at all times. The impact of depth-RRFA will not be affected as long as there exists at least one unused IP address.
2. The AODV RFC[1] specifies that a node should not originate more than RREQ_RATELIMIT RREQs per second. This can prevent attacks from the application layer but does not prevent the attacker from modifying the routing protocol to set RREQ_RATELIMIT to a very large value. Furthermore, genuine RREQ attempts to reachable destinations can be hindered since they may be dropped when RREQ_RATELIMIT is always reached due to excessive forged RREQs.
3. In Ariadne[10] which is designed for DSR networks, route discovery chains are used to rate-limit the number of route discoveries. Each route discovery needs a key from the route discovery chain and the release of keys can be regulated. This limits the impact of RRFA on the network but a fixed number of forged RREQs can still be injected into the entire network. Furthermore, genuine RREQ attempts from a compromised node to reachable destinations may never be sent if the number of forged RREQs generated by it is large.

4. In [11], an adaptive statistical packet dropping mechanism is proposed to defend against malicious control packet floods like RRFA but the mechanism does not distinguish between genuine and forged RREQs from the malicious or victim node. Furthermore, the effectiveness of the mechanism depends on the network conditions.
5. In [12], a priority system is used to determine the transmission priority of RREQs. When the malicious node broadcast excessive RREQs, the priorities of its RREQs are reduced. Similarly, this method does not distinguish between genuine and forged RREQs from the malicious or victim nodes.

4 Route Request Flooding Defence (RRFD)

In this paper, we assume that there exists a security mechanism, such as public key cryptography and digital signatures that enables a node to authenticate routing messages from any node in the network. Therefore, a malicious node cannot spoof the originator and destination IP addresses in a RREQ packet although the destination IP address may not be reachable in the network. To alleviate the impact of RRFA on the overall performance of the network, we propose the Route Request Flooding Defence (RRFD) mechanism that aims to do the following:

1. Minimize the impact of breadth-RRFA and depth-RRFA on the entire network. Neighbours of a malicious node will not rebroadcast most of the forged RREQs from that node, thus confining the impact to the neighbours of that malicious node.
2. Identify forged RREQs to a very high accuracy. This is achieved by observing the RREQs received. If there are many RREQs with the same source and destination sent within a short period of time, then it is highly likely to be forged.
3. Allow the malicious node to maintain or establish valid data communications to reachable destinations. Since the user of the malicious node may not know that the node has been compromised, it would be beneficial to allow the malicious node to continue its communication with other nodes in the network.
4. Subject the packets to two rounds of examination before they are retransmitted into the network. Firstly, the RREQ is examined to determine if it is forged using RRFA. If the RREQ is forged, it will be dropped with no further authentication. Otherwise, if the RREQ is deemed to be genuine, then it will be authenticated to ensure that the RREQ is not forged.

4.1 RRFD Design

RRFD consists of three components: RREQ binary exponential backoff, Route Discovery Cycle (RDC) binary exponential backoff and Fast Recovery. In RREQ binary exponential backoff, each node will ensure that its neighbour follows a binary exponential backoff when sending RREQs in a RDC. If RREQs are sent faster than what is allowed, excess RREQs are dropped. This ensures that the generation of RREQs in a RDC follows a binary exponential backoff as stated in the AODV specifications [1]. Fig. 1 illustrates the earliest times in which RREQs are allowed to be sent in a RDC if the number of RREQ retries is set to 2.

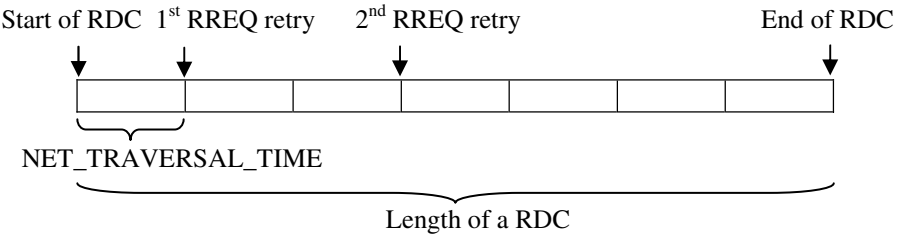


Fig. 1. Illustration of a Route Discovery Cycle (RDC)

In RDC binary exponential backoff, each node will ensure that its neighbour follows a binary exponential backoff when initiating another RDC. The waiting time between successive RDCs will be doubled after each successful RDC. Using RDC binary exponential backoff, the impact of RRFA would be reduced exponentially if the attacker persists in sending forged RREQs. Fig. 2 illustrates the earliest times in which a new RDC is allowed to proceed if a node keeps receiving RREQs with the same pair of source and destination node in every RDC.

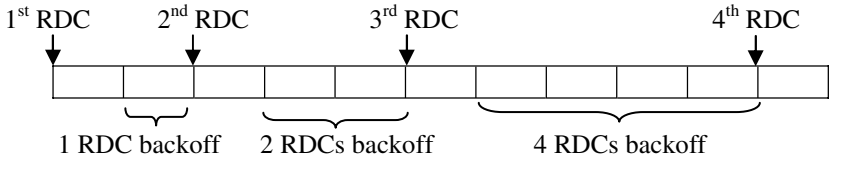


Fig. 2. Illustration of RDC binary exponential backoff

In Fast Recovery, the number of RDCs that a node will need to wait before initiating another RDC will be reduced exponentially if it does not initiate another RDC for at least one RDC period. Due to node movement, link breakages may occur after a period of time. Therefore, Fast Recovery ensures that nodes that send genuine RREQs do not get penalized by RRFD. By using RDC binary exponential backoff in conjunction with Fast Recovery, the aim of identifying forged RREQs to a very high accuracy is achieved since we drop most of the forged RREQs without affecting the transmission of genuine RREQs.

To implement RRFD, each node maintains an entry in a RREQ database for every unique pair of originator and destination nodes. It is not sufficient to maintain an entry for just every originator that is a neighbour of each node as collusion might occur in the case of two or more neighbouring malicious nodes. When a RREQ is received, we search the RREQ database for an entry. If an entry is not found, we create a new entry, start a new RDC and rebroadcast the RREQ. If an entry exists, this indicates that there was a previous RREQ with the same originator and destination. If the RREQ is in a current RDC, the backoff time will be determined based on the number of RREQs already forwarded by the node in the current RDC. If we have reached the end of the RDC after forwarding the RREQ, the RDC backoff would be set to twice the previous value subject to a predetermined maximum value (we set this value to 64 in our work). If the RREQ is not in a RDC, the backoff time will depend on the

number of RDCs to backoff, which may be reduced according to the Fast Recovery algorithm (if a node has not transmitted a similar RREQ for at least one RDC).

4.2 RRFD Analysis

To provide a simple mathematical analysis of the success of RRFD against breadth-RRFA, we define a metric *S* as:

$$S = 1 - \frac{\text{Number of forged RREQs forwarded by node if RRFD is used}}{\text{Total number of forged RREQs forwarded by node if RRFD is not used}}$$

which refers to the ratio of forged RREQs that are dropped by node A when RRFD is used against the number of forged RREQs forwarded by node A when RRFD is not used. Therefore, *S* measures the success ratio of RRFD against RRFA. The value of *S* depends on the type of attacks and changes over time. Therefore, we define S_n as the success rate before the $(n+1)^{th}$ successful RDC by an attacker, M. Let R_i be the success rate of the node in using RRFD to defend against RRFA by M during the interval between the start of the i^{th} and $(i+1)^{th}$ successful RDC. Also, let *x* be the number of RREQs transmitted in one RDC. Therefore,

$$R_i = 1 - \frac{x}{(1+\min(2^{i-1},64))x} = 1 - \frac{1}{(1+\min(2^{i-1},64))}$$

R_1 is $\frac{1}{2}$ since one RDC of RREQs is dropped during the interval between the start of the first and just before the start of the second successful RDC by M. The maximum value of R_i is $\frac{64}{65}$ since we set the maximum number of RDC backoff to be 64.

$$S_n = \frac{\sum_{i=1}^{n} R_i * (1+\min(2^{i-1},64)) * x}{\sum_{i=1}^{n} (1+\min(2^{i-1},64)) * x} = 1 - \frac{nx}{nx + \sum_{i=1}^{n} \min(2^{i-1},64) * x}$$

For values of $n \geq 7$,

$$S_n = 1 - \frac{nx}{(n+1+2+4+8+16+32+64+(n-7)64)x} = 1 - \frac{1}{65 - \frac{321}{n}}$$

$$S_{\max} = \lim_{n\to\infty} S_n = 1 - \frac{1}{65} = \frac{64}{65}$$

Therefore, the upper bound of *S* is about 98%. To provide a lower bound of *S*, we need to consider how M may reduce the impact of RRFD on his attack. To maximize the impact of the attack to counter RRFD, M will stop sending RREQs for at least one RDC after a successful RDC. This ensures that the number of RDCs that M has to wait before initiating another successful RDC will be 1.

$$S_{\min} = 1 - \frac{nx}{2nx} = \frac{1}{2}$$

Since M will not send any RREQ at least for one RDC after a RDC has ended, the lower bound of *S* is at least 50%. Therefore, we have proven that RRFD is at least 50% effective against a breadth-RRFA.

Similarly, to find out the success of RRFD against depth-RRFA, we assume that M broadcasts y RREQs per RDC with each RREQ spaced equally apart and x is the number of RREQs sent in a RDC. We then define S_n as:

$$S_n = 1 - \frac{nx}{ny + \sum_{i=1}^{n} \min(2^{i-1}, 64) * y}$$

This equation is the same as the equation in the breadth-RRFA analysis except that the number of RREQ transmitted in a RDC is y instead of x. For values of $n \geq 7$,

$$S_n = 1 - \frac{nx}{(n+1+2+4+8+16+32+64+(n-7)64)y} = 1 - \frac{x}{(65 - \frac{321}{n})y}$$

$$S_{\max} = \lim_{n \to \infty, y \to \infty} S_n = 1$$

When y is much larger than x, RRFD is very effective against RRFA and the success rate approaches 100%. To maximize the number of RREQs forwarded by A, M will stop sending RREQs for at least one RDC after a RDC. Therefore,

$$S_n = 1 - \frac{nx}{2ny} = 1 - \frac{x}{2y}$$

In a depth-RRFA, y will be at least equal to x. Therefore, the lower bound of S is 50%. If y is much larger than x, then

$$S_{\min} = \lim_{y \to \infty} S_n = 1$$

Therefore, RRFD is very effective against depth-RRFA as the success rate can approach 100% if the attacker is very aggressive in sending large number of RREQs.

5 Simulation Results

To determine the effectiveness of RRFD against depth-RRFA and breadth-RRFA, we simulated various scenarios of 100 nodes randomly placed in a 1600m×1600m area. The nodes use the random waypoint mobility model with speeds 0~20m/s and pause time of 10 secs. Constant Bit Rate (CBR) data sources are used with a rate of 2 pkts/sec and 512 bytes/pkt. The MAC layer used is based on the IEEE 802.11 with data rate of 2Mbps.

For comparison, we evaluated RRFD and RREQ-Rate-Limit subject to breadth-RRFA and depth-RRFA by one malicious node which is not restricted in the number of RREQs it can originate. In RREQ-Rate-Limit, a node will transmit at most n RREQs per second (we set n = 10 in our simulations). Each set of simulations is run for 5min each, and repeated using 5 different scenarios.

The *packet delivery ratio* (PDR) measures the number of successfully delivered data packets against the total number sent. The *routing overhead* includes all the routing control packets, namely, RREQs, RREPs and RERRs. The *average end-to-end delay* is the average time taken for data packets to travel from source to destination.

5.1 Breadth-RRFA Analysis

First, we investigate the impact of breadth-RRFA by one malicious node on the network performance. A total of 20 valid random connections are made from any source

to destination, excluding the malicious node. In each simulation, the number of unreachable destinations varies from 0 to 200.

The results in Fig. 3 show that both RRFD and RREQ-Rate-Limit outperform normal AODV with increasing number of forged RREQs. However, when there is no RRFA, using RRFD results in a loss of about 10% of the data packets. This is the result of genuine RREQs being dropped due to fast link breakages on some routes. The routing overhead is greatly reduced when RRFD and RREQ-Rate-Limit is used. RREQ-Rate-Limit performs better than RRFD in terms of the routing overhead in most cases as RREQ-Rate-Limit places an upper limit on the number RREQs being retransmitted while RRFD allows more RREQs to be retransmitted when the number of unreachable destinations increases. Nevertheless, RRFD still outperforms the normal AODV in terms of PDR and has lower routing overhead as the number of unreachable destinations increases. In all cases, the end-to-end delays for RREQ-Rate-Limit and RRFD are lower as compared to the normal AODV.

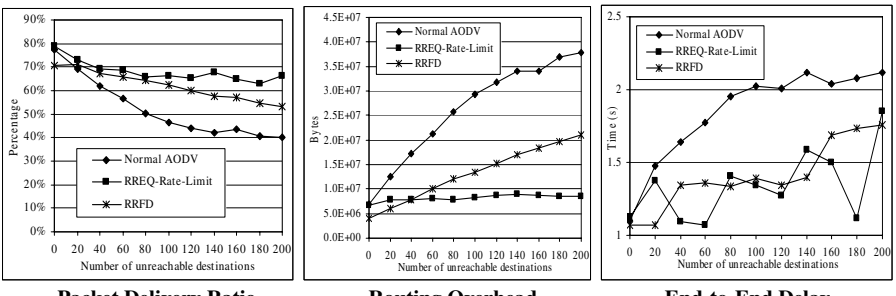
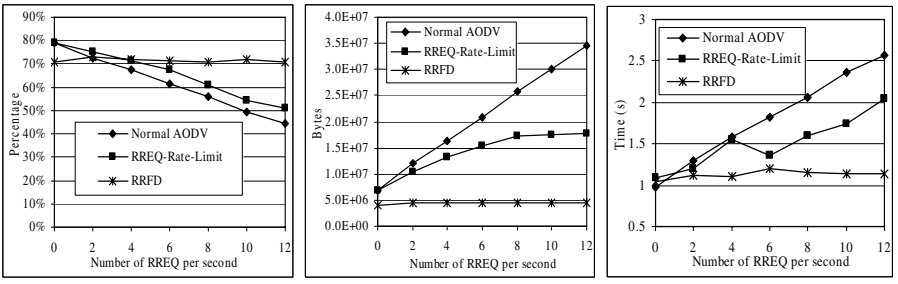


Fig. 3. Comparison of RRFD and RREQ-Rate-Limit against normal AODV in breadth-RRFA

5.2 Depth-RRFA Analysis

Next, we will investigate the impact of depth-RRFA from one malicious node. A total of 20 valid random connections are made from any source to destination, excluding the malicious node. In each simulation, the number of RREQs sent per second from the malicious node varies from 0 to 12.

The results in Fig. 4 show that RRFD performs much better than RREQ-Rate-Limit and normal AODV. The PDR remains consistently above 70% despite the transmission of more forged RREQs, while the performance of normal AODV and RREQ-Rate-Limit decreases with increasing forged RREQs. When there is no RRFA, using RRFD results in a loss of about 10% of the data packets, as in the previous case. For RRFD, the routing overhead incurred is significantly lower than RREQ-Rate-Limit as RRFD ensures that the surrounding nodes drop forged RREQs from the malicious node while RREQ-Rate-Limit will continue to rebroadcast them until the limit of 10 RREQs per second is reached. In most cases, the end-to-end delay experienced when using RRFD is also significantly lower than RREQ-Rate-Limit and normal AODV. Depth-RRFA can be considered to be more critical as typical methods like address filtering is not effective against it. Hence, there is a need for better protection against depth-RRFA, which RRFD can provide.



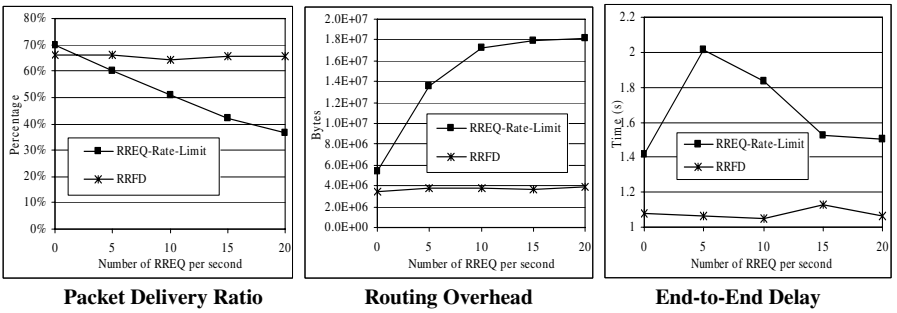
Packet Delivery Ratio **Routing Overhead** **End-to-End Delay**

Fig. 4. Comparison of RRFD and RREQ Rate Limit against normal AODV in a depth-RRFA

5.3 Increasing Good Throughput

The simulations in Section 0 and Section 0 do not include any data flow from the malicious node. It is desirable that valid data communications from the malicious node is allowed to continue so that it may receive information from other nodes if it is a compromised node. For example, if the node is infected by a Trojan horse, virus or a worm, it is desirable that the victim can obtain patches or warning messages from other nodes in order to recover from the compromised status.

Using depth-RRFA, we aim to determine whether RRFD or RREQ-Rate-Limit is more effective in allowing malicious nodes to send genuine RREQs while dropping forged RREQs originating from the malicious nodes. In each simulation, the number of forged RREQs sent per second from the malicious node varies from 0 to 20. The results in Fig. 5 show that RRFD is better than RREQ-Rate-Limit when the number of forged RREQs sent increases. For RRFD, the PDR remains constant despite an increase in the number of forged RREQs in the network. However, when there is no RRFA, using RRFD results in a loss of about 4% of the data packets. RRFD outperforms RREQ-Rate-Limit with lower routing overheads and end-to-end delays.



Packet Delivery Ratio **Routing Overhead** **End-to-End Delay**

Fig. 5. RRFD vs RREQ-Rate-Limit in depth-RRFA

6 Conclusion

In this paper, we have presented an effective mechanism, the RRFD, to mitigate the effect of denial-of-service attacks from malicious nodes by flooding the network with

RREQs to unreachable destinations. Although RREQ-Rate-Limit performs better than RRFD in breadth-RRFA, RRFD outperforms RREQ-Rate-Limit in depth-RRFA and allowing the victim node to communicate with other nodes. Breadth-RRFA is not as critical as depth-RRFA as it can be easily mitigated by schemes like address filtering. As the simulation time used in the current study is rather short (only 5 minutes), we believe that RRFD will perform even better as time increases because more forged RREQs will be dropped. In terms of scalability, RRFD works well regardless of the size of the network since the total number of RREQs in the network is not restricted while the effectiveness of RREQ-Rate-Limit will depend greatly on the size of the network since each node is allowed only to transmit a fixed number of RREQs.

References

1. C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC3561, July 2003
2. M. Hollick, J. Schmitt, C. Seipl and R. Steinmetz, "On the Effect of Node Misbehavior in Ad Hoc Networks", *Proceedings of the IEEE International Conference on Communications,* pages 3759-3763, June 2004
3. L. Buttyán and J.-P. Hubaux, "Report on a Working Session on Security in Wireless Ad Hoc Networks", *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 7, Issue 1, pages 74-94, January 2003
4. P. Ning and K. Sun, "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols", *Proceedings of the 2003 Annual IEEE Information Assurance Workshop,* pages 60-67, June 2003
5. W. Wang, Y. Lu and B. Bhargava, "On Security Study of Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks", *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 179-186, March 2003
6. M. G. Zapata and N. Asokan, "Securing Ad hoc Routing Protocols", *Proceedings of the ACM Workshop on Wireless Security (WiSe 2002),* pages 1-10, September 2002
7. Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks", *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 135-147, October 2003
8. A. Mishra, K. Nadkarni and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks", *IEEE Wireless Communications*, Vol. 11, Issue 1, pages 48-60, February 2004
9. G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer and R. A. Kemmerer, "An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks", *Proceedings of the 20th Annual Computer Security Applications Conference,* December 2004
10. Y. Hu, A. Perrig and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks", *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, pages 12-23, September 2002
11. S. Desilva and R.V. Boppana, "Mitigating Malicious Control Packet Floods in Ad Hoc Networks", *Proceedings of 2005 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2112-2117, March 2005
12. P. Yi, Z. Dai, Y. Zhong and S. Zhang, "Resisting Flooding Attacks in Ad Hoc Networks", *Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 657-662, April 2005

Advanced Networking

Performance Analysis of an Efficient Network Transition Mechanism Supporting Mobile IPv6*

Su-Jin Lee[1], Jungjin Park[1], Hyun-Kook Kahng[1], and Ilyoung Chong[2]

[1] Dept. of Electronics and Information Engineering, Korea University, Anam-dong, Sungbuk-Gu, Seoul 136-701, Korea
{aza97, pjj, kahng}@korea.ac.kr
[2] Dept. of Information and communications Engineering, Hankuk University of Foreign Studies, Imun-dong, Dongdaemun-Gu, Seoul 130-790, Korea
iychong@hufs.ac.kr

Abstract. IPv6 will replace IPv4 as the widely used protocol in the Internet because this protocol can solve the inherent problems in IPv4, i.e., large address space, better mobility support, easier security integration than IPv4. Since a huge number of sub-networks are already installed using IPv4, the transition phase where IPv4 and IPv6 coexistence is inevitable. There are many transition mechanisms to support communication between incompatible nodes in this situation. However, they scarcely ever consider mobility of the nodes. When a mobile IPv6 node wishes to communicate with an IPv4 node, they can communicate through MIPv6 and any translation mechanism.However, the communication is inefficient because of tunneling and reverse tunneling via a home agent. To support efficient routing, we propose a simple efficient transition mechanism extending NAT-PT, which adds functions to be able to handle packets for communications on behalf of the home agent. Implementation results indicate that the proposed mechanism extending NAT-PT, named NAT-PTm, provides more efficient mobility than NAT-PT in the mobile environment.

1 Introduction

Internet users want to use a service of the high quality anywhere and anytime. Mobile of users are gradually increasing because of the development of wireless communication technologies and the resulting improvement in performance for portable computing devices such as labtops and PDAs. Insufficient address space would occur if we keep using IPv4 addresses. IPv6 was developed to provide sufficient address space so that next generation networking needs such as network security, improved mobility and Quality of Service (QoS) could be fulfilled. Mobile IPv6 can support mobility more efficiently than mobile IPv4, Mobile IPv6 provides mobility using IPv6s functions. So, it is desirable that the IPv4 network changes to the IPv6 network soon. However, IPv6 deployment is going to be a gradual process since a huge amount of sub-networks are already installed

* This research was supported by University IT Research Center Project.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 339–348, 2006.

for IPv4. Therefore, the transition phase of IPv4 and IPv6, the coexistence environment, is inevitable. In this environment, there are some sets of transition mechanisms such as 6to4, NAT-PT, DSTM, and etc., to insure transparent communications between the two protocols. When mobility in this environment is considered, there are various scenarios depending on the protocol version of mobile node, correspondent node and the visited network. Also, in order to support mobility in each possible scenario, the associated transition mechanisms should be modified according to some considerations. From these various scenarios using transition mechanisms, we consider one scenario using NAT-PT when a mobile IPv6 node communicating with IPv4 node moves to a different network in the same IPv6 island. Even though it is said that NAT-PT may not be efficient, still NAT-PT is one of solutions for interconnection between an IPv6 network and an IPv4 network. In the real market, NAT-PT is deployed as a transition mechanism. Our work is to propose, implement, and analyze a new transition mechanism, NAT-PTm, for supporting mobility efficiently. The scenario considered in this paper is similar to one of scenarios in portable Internet that is called WiBro. So, the proposed mechanism could be considered as deployment of portable Internet. The remainder of this paper is organized as follows. In section 2, we describe the proposal of an efficient mechanism which extends NAT-PT. Section 3 presents implementation and test of NATPTm. Section 4 describes the performance analysis of the test with discussions of the analysis. Finally brief conclusions will be drawn.

2 Proposal of Efficient Transition Mechanism

We propose the transition mechanism extending NAT-PT for supporting mobility efficiently between a mobile IPv6 node and a correspondent IPv4 node over a limited transition environment. In this section, we show a problem that happens in the Figure 1 environment when applying NAT-PT and we propose an efficient transition mechanism which extends NAT-PT for the mobile environment.

2.1 Inefficient Routing of NAT-PT

Figure 1 shows a testbed with NAT-PT installed. We analyze the traffic flows on the interface of each node when a mobile IPv6 node communicating with a correspondent IPv4 node moves to a visited network in the same IPv6 island. It is a simple model using one NAT-PT since all requests and responses pertaining to a session are routed via the same NAT-PT router. For this test, we use NAT-PT and Mobile IPv6 sources from KAME kit[7] that is based on Free BSD. In this environment, we analyze traffic related to the communications. Figure 2 depicts the traffic flows which are request and response messages to continue the communication between two incompatible nodes. The traffic flows are processed as follows: a mobile node (MN) moving to a different network in the same IPv6 island obtains a new prefix from the router advertisement message. Then, the MN creates a care-of address and registers care-of address by sending a Binding Update (BU) message to its home agent. The home agent that has

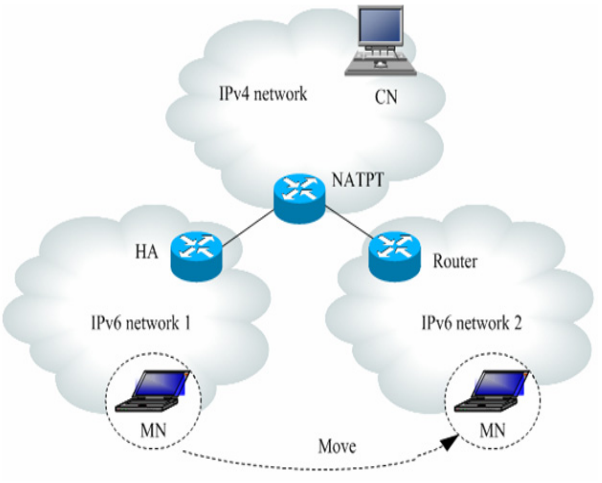


Fig. 1. Testbed in our work

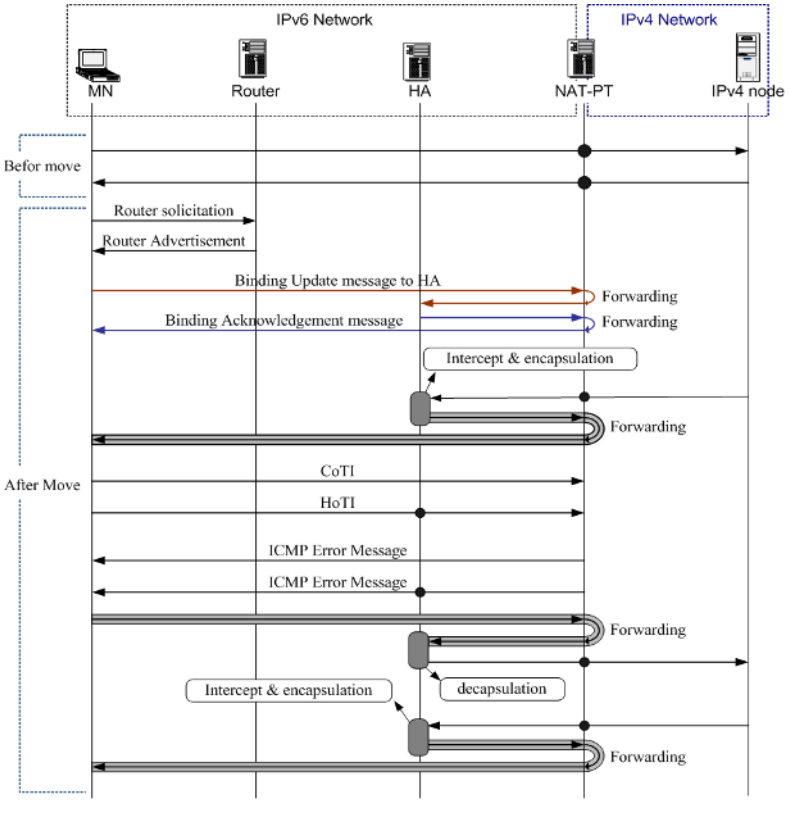


Fig. 2. Traffic flows in the environment installing NAT-PT

received the BU message intercepts a packet destined to MNs home address from a correspondent node (CN). After encapsulating the packet, the home agent forwards it to the MNs care-of address via NAT-PT as shown in Figure 1. The MN that has received the encapsulated packet sends HOTI and COTI messages to the CN for the Return Routability procedure before sending the BU message to the CN. Since NAT-PT cannot understand these messages, it returns an ICMP error message to the MN because there are no IPv4 messages corresponding to the MIPv6 messages. Therefore, packets toward the CN are tunneled from the MN to the home agent (i.e., reverse tunneling) and then routed normally from the home agent to the CN. Packets to the MN are intercepted and tunneled by the home agent and then routed normally from the home network to the MN(i.e., tunneling). Routing optimization could not be achieved using tunneled and reverse tunneled packets via the home agent. This kind of use also causes overhead in the home agent unnecessarily which, in turn, decreases performance. So, we propose this efficient transition mechanism to avoid overhead of the home network and to achieve efficient routing.

2.2 Design of the Transition Mechanism

The main consideration of this mechanism is to distribute the process overhead of the home agent to other devices. To avoid overhead caused by reverse tunneled or tunneled packets in the home network, NAT-PT should be able to encapsulate or decapsulate packets via the home agent for reverse tunneling or tunneling of packets on behalf of the home agent. To do so, NAT-PT should be modified to be able to know the MNs care-of address and the home agents address. This is information used to handle tunneled(or reverse tunneled) packets. In order to obtain and manage this information, a proposed mechanism, NAT-PTm, extends entry of the mapping table in NAT-PT to manage mobility information of mobile nodes. Also, this mechanism is modified to understand the binding update message before forwarding the binding update message to the MNs home network. The information is stored into an associated entry of the extended mapping table accompanied by checking a flag to indicate whether mobility is on or not. To store the information into a related entry at the mapping table, we can find its associated entry by searching for entries that have their address matched with the address of the home address destination option from the binding update message

3 Implementation of Extended NAT-PT, NAT-PTm

This section describes a procedure of an implemented module for the extended NAT-PT. We are based on the KAME kit[7], that runs in free BSD, to implement the extended NAT-PT.

3.1 The IPv6 Translation Module of the NAT-PTm

Figure 3 shows the procedure of the IPv6 translation module which translates IPv6 packets into IPv4 packets as we described in section 2.2. First of all, on receiving an IPv6 packet, the IPv6 translation module checks whether the IPv6

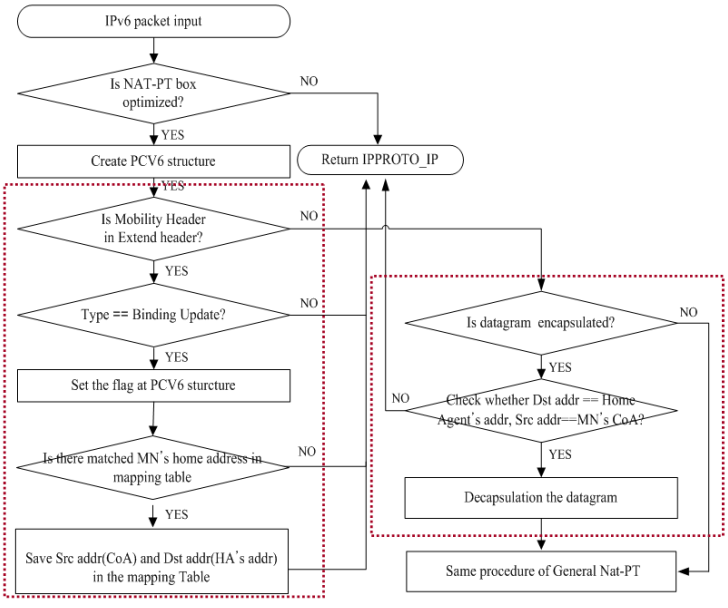


Fig. 3. The procedure of IPv6 translation module

packet includes a mobility header or not. If the IPv6 packet is a BU message, it sets a flag in the PCV6 structure (i.e., the protocol control block in NAT-PT for the IPv6 datagram) extending it to indicate whether the IPv6 node related to the communication has moved or not. If not a BU message, the extended NAT-PT has same operation as NAT-PT. If a BU message, the source address(i.e., MNs care-of address) and destination address(i.e., home agents address) from the BU message are stored as a suitable entry in the mapping table which has stored the MNs home address matching the home address destination option of the BU message. This address information is used to de-tunnel the packet in the extended NAT-PT on behalf of the home agent.

3.2 The IPv4 Translation Module of the NAT-PTm

Figure 4 shows the procedure of the IPv4 translation module. The module translates IPv4 packets to IPv6 packets as we described in section 2.2. The IPv4 translation module translates IPv4 packets to IPv6 packets whose destination is MNs home address. Then, if the mobility flag field in the suitable mapping table is set to one, the module returns the MNs care-of address and the home agents address registered into a suitable entry of mapping table. These addresses are used to encapsulate the translated IPv6 packet on behalf of the home agent.

3.3 Test of the NAT-PTm

Figure 5 shows the packet flows for communication between two incompatible nodes in Figure 1s environment implemented on NAT-PTm. When the MN

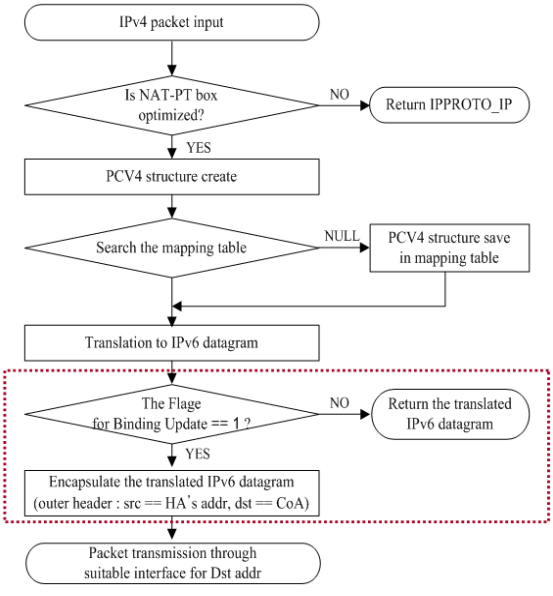


Fig. 4. Procedure of the IPv4 translation module

has moved to a different network in the same IPv6 island, the MN sends the home agent a binding update message. On receiving the binding message at the NAT-PTm of the boundary router, the NAT-PTm forwards messages to the home agent after the NAT-PTm obtains the MNs care-of address and the home agents address needing to handle the packets on behalf of home agent. It then stores the information into an associated address entry at the mapping table. Thereafter, when the NAT-PTm receives an IPv4 packet from the correspondent IPv4 node, it translates it to an IPv6 packet. Then, the extended NAT-PT encapsulates the translated IPv6 packet using the extended information, that is the associated entry has the information of the MNs care-of address and the home agents address with the setting of the mobility flag related to mobility. Identically, when the boundary router receives a reverse tunneled IPv6 packet, it checks the destination address of the outer header before the extended NAT-PT handles the packet. If the destination address is the home agents address, the packet is decapsulated and translated to an IPv4 packet.

4 Performance Analysis

To compare the newer NAT-PTm with an original NAT-PT, we measured traffic on the interface of each node in the environment shown in Figure 1. The results show that use of the proposed mechanism provided the desired routing and caused lower overhead than use of the original NAT-PT.

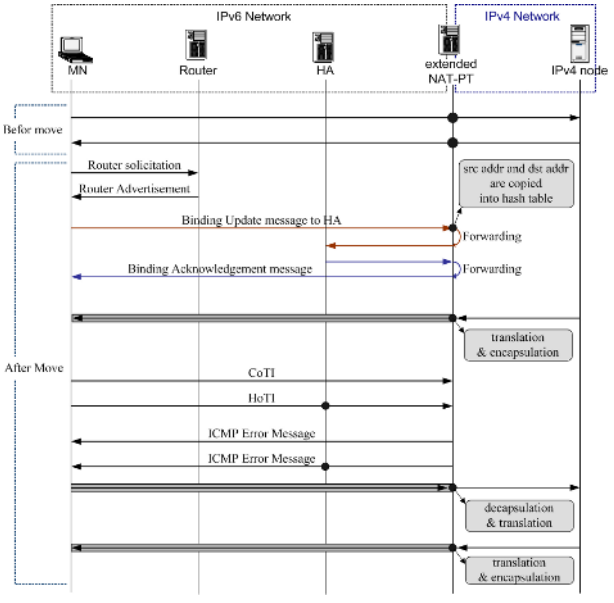


Fig. 5. Packet flows in the environment that has installed the extended NAT-PT

4.1 Comparison of the Two Mechanisms

When we tested in the environment installing NAT-PT, the boundary router could not forward packets due to overflow during a short period of time. Most packets were concentrated on the interface connected to the home network of the boundary router since packets were forwarded through tunneling via the home agent. However, in the environment installing the proposed mechanism, it did not happen since packets were not forwarded via the home agent. Figure 6 and Figure 7 show the receiving rate(byte/sec) and downloading end time for 300MB data on the interface of the MN. The unit of the X-axis and the Y-axis are seconds and bytes, respectively. In these figures, the traffic between 0 and 15 seconds depicts the time during which the MN stays in the home network up until the MN moves to a different network. The next time gap, 10 seconds between 15 seconds and 25 seconds, is the MNs handoff time. As you can see in Figure 6 and Figure 7, there are differences in the rate(byte/sec) and in the downloading end time. In Figure 6, the rate of data receipt to the MN is lower than them that of Figure7, however it is nearly constant. On the other hand, the rate of data is not constant in the Figure7. The reason is described as follows. In the environment having installed NAT-PT, the time(i.e.,Round Trip Time) receiving ACK from MNv6 to CNv4 is longer than the case of applying NAT-PTm, since traffics between the MN and the CN are always forwarded via the home agent. Therefore, because packets are stored constantly in the sending buffer of the CN until receiving an ACK, the CN can send packets constantly and the MNv6 also can receive packets constantly. On the other hand,

in the environment that installed the proposed mechanism, since the traffic from the CN is sent directly to the MN without forwarding to the home agent, the time(i.e., RTT) receiving an ACK from the MNv6 to the CNv4 is shorter than the RTT applying NAT-PT. The CNv4 sends directly packets to the MNv6 whenever the CNv4 receives an ACK. So, the rate of data is fluctuating because the CNv4 sends, directly, packets to MNv6 although the sending buffer of the CN is not full. As shown from the data receiving rate and receiving end time in Figure 6, the average receiving rate with NAT-PT is about 3.3Mbyte and the average receiving rate with the proposed mechanism is about 4.6Mbyte. Use of the proposed mechanism extending NAT-PT shows that the performance efficiency is increased by about 39%.

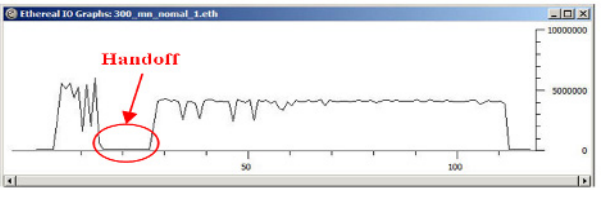


Fig. 6. Results using NAT-PT

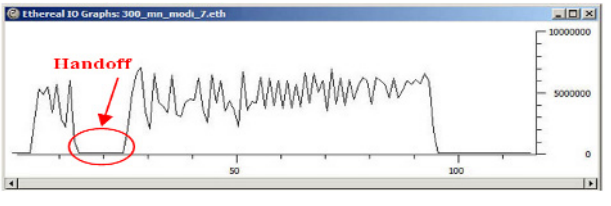


Fig. 7. Results using proposed mechanism

4.2 Analysis Based on Traffic Incoming to Each Interface of Boundary Router

Table 1 shows traffic rates that are measured on the interfaces of each boundary router. In the case of applying NAT-PT, half of the traffic incoming to the boundary router is concentrated on the interface connected to the MNs home network. Because the destination of almost all traffics is the MNs home address or CN via home agent by reverse tunneling. On the other hand, in the case of applying the proposed mechanism, traffic is not concentrated on the interface connected to the home network. A border router with the installed proposed mechanism does not forward the translated packets to the home network, but instead, it encapsulates the packet on behalf of home agent and forwards directly it to the MN. It can reduce overhead due to the unnecessary concentration on the interface connected to the home network. It can also achieve packet forwarding more efficiently. Figure 8 and Figure 9 show traffic flows on each interface of routers applying NAT-PT and NAT-PTm. In the case of applying the proposed

Table 1. Traffic rate concentrated on each interface

interface	NAT-PT	Proposed mechanism
interface 0 (connected to IPv4 network)	26.7%	50.0%
interface 2 (connected to visited network)	23.4%	44.0%
interface 3 (connected to home network)	49.9%	6.0%

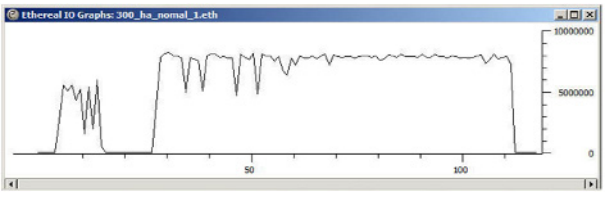


Fig. 8. Traffics on the interface connected to home network at NAT-PT

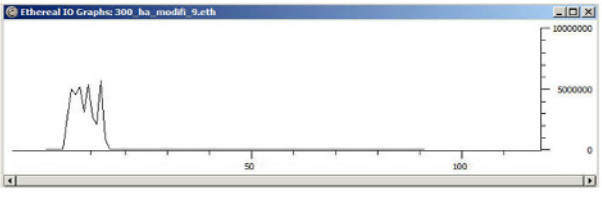


Fig. 9. Traffics on the interface connected to home network at NAT-PTm

mechanism, there are no packets related to the communications on the interface connected home network at the boundary router. While, in the case of applying NAT-PT, you can see that traffic is concentrated on the interface connected to the home network due to tunneling or reverse tunneling of packets. As a result, there is a severe performance reduction at the home network

5 Conclusion

In the IPv4/IPv6 coexistence environment, to support mobility between mobile IPv6 and the IPv4 node, the Mobile IPv6 and NAT-PT mechanism can work together. However, when they are applied in this situation, some problems develop. There becomes inefficient routing and low performance in the home network. In this paper, we considered the above problems and improved performance in the limited environment such as testbed. In the environment where there is a mobile IPv6 node in a IPv6 network communicating with IPv4 node in an IPv4 network that moves to a different IPv6 network in the same island, inefficient routing happens since all packets should communicate through tunneling and

reverse tunneling via a home agent. Due to the tunneling or reverse tunneling packets incoming to home network, low performance of the home network results. To improve it, we proposed an efficient mechanism extending NAT-PT, NAT-PTm. The proposed mechanism extended the mapping table to handle the related mobile IPv6 message on behalf of the home agent. Also, the proposed mechanism modified some parts of translation modules to handle encapsulation or decapsulation of packets on behalf of the home agent. As a results performance related to the communication is improved. In the case of applying the proposed mechanism, the receiving rate of data is higher than in the case of applying NAT-PT, about 39%. Since the proposed mechanism reduces packets toward the home network that should be forwarded to another network, performance in the home network is also improved. Also, NAT-PTm can help deployment of the portable Internet called WiBro that is going to use IPv6. However, we tested under a limited environment and did not consider any security issues. So, there is required additional work to apply this mechanism over real networks.

References

1. Perkins, C., Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002
2. D.Johnson, C. Perkins, J.Arkko, Mobility Support in IPv6, RFC 3775. June 2004
3. S.Deering and R. Hinden, Internet Protocol, Version6(IPv6) Specification, RFC 2460. December 1998
4. G. Tsirtsis, P.Srisuresh, Network Address Translation - Protocol Translation (NAT-PT), RFC 2766. February 2000
5. KAME Project, "http://www.kame.net"

LAID: Load-Adaptive Internet Gateway Discovery for Ubiquitous Wireless Internet Access Networks*

Bok-Nyong Park[1], Wonjun Lee[1,**], Choonhwa Lee[2], Jin Pyo Hong[3], and Joonmo Kim[4]

[1] Dept. of Computer Science and Engineering, Korea University, Seoul, Republic of Korea
wlee@korea.ac.kr
[2] College of Information and Communications, Hanyang University, Seoul, Republic of Korea
[3] Dept. of Information and Communications Engineering, Hankuk University of Foreign Studies, Republic of Korea
[4] School of Electrical, Electronics, and Computer Engineering, Dankook University, Seoul, Republic of Korea

Abstract. Ubiquitous Internet connectivity is to connect all devices to the Internet at any time and any place. To achieve this ubiquitous Internet connectivity, we consider integrating the Internet and mobile ad-hoc networks. One of the most important issues in the ubiquitous Internet connectivity is to find an efficient and reliable Internet gateway. We propose a load-adaptive Internet gateway discovery approach that can exploit network conditions. The load-adaptive Internet gateway discovery scheme dynamically adjusts a proactive area according to network traffic. Among the candidates, a serving gateway is selected based on offered load. The simulation results show that our discovery scheme outperforms existing discovery schemes.

1 Introduction

With the increase of potable devices as well as progress in wireless broadband communications, an integration of different heterogeneous wireless networks will be one of the areas for next generation wireless/mobile networks. To realize seamless heterogeneous wireless/mobile networks, we focus on ad-hoc networks providing Internet connection. This is referred as a ubiquitous wireless Internet access network. Ad-hoc networks are considered complementary to IP networks in a sense that Internet connectivity can be extended into the ad-hoc networks,

* This work was supported by grant No. B1220-0501-0205 from the University fundamental Research Program of the Ministry of Information & Communication in Republic of Korea.
** Corresponding Author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 349–358, 2006.

making them part of the Internet. The ubiquitous wireless Internet access network architecture is highly scalable and cost effective, offering a solution to the easy deployment of ubiquitous wireless Internet.

Main issue in the ubiquitous wireless Internet access network is to discover an Internet gateway (IG). When an ad-hoc mobile node (AMN) wants to connect the Internet, it should be able to connect an appropriate IG. To achieve efficient integration and ubiquitous Internet access, we need efficient IG discovery scheme which determines the quality of the Internet connectivity. There have been three proposed approaches in the IG discovery: proactive, reactive, and hybrid [6,7,8]. A proactive approach [10] enables good connectivity and low latency, but requires considerable overheads. In contrast, a reactive approach [9] achieves low routing overhead at the expense of increased latency. A hybrid approach [1,6,8] uses a proactive approach within a gateway's advertisement range, while it uses a reactive approach outside the coverage. One of primary challenges to design a hybrid discovery scheme is to determine the optimal proactive area. However, existing hybrid schemes set their proactive area once and do not dynamically adjust it [7], which may not be an appropriate range any more for changing network conditions. To improve existing hybrid discovery schemes, we propose a new ***L**oad-**A**daptive hybrid **I**nternet gateway **D**iscovery* (LAID) scheme that dynamically adjusts its proactive area according to changing network conditions. Once routes are discovered to IGs, AMNs should be able to select one IG providing the best Internet connection. In the Internet gateway selection, our selection method distributes data packets into different IGs while keeping low offered load. It decreases the average delay and the packet drop rate.

The rest of this paper is organized as follows. Section 2 discusses an overview of the Internet connectivity in ubiquitous wireless Internet access networks, and Section 3 proposes a load-adaptive Internet gateway discovery. Section 4 presents our performance evaluation. Finally, we draw out conclusion in Section 5.

2 Overview of Ubiquitous Wireless Internet Access Networks

Ad-hoc networks can be applied anywhere with low cost and share data between device, where there is little or no communication infrastructure [5]. For access to ubiquitous Internet services, an *Internet gateway* (IG) in access networks provides Internet connectivity for *ad-hoc mobile nodes* (AMNs). AMNs are usually connected to the Internet through one or more IGs. This IG is part of both networks and acts as a bridge between an ad-hoc network and the Internet. First, packets from an AMN are forwarded to an IG and then transmitted to their destination in the Internet. IG is equipped with both interfaces: the first wired interface for the Internet and the second radio interface for ad-hoc networks. Thus, IGs run ad-hoc routing protocols to act as an AMN, and it operates as a member of a fixed subnet connected to the Internet. Fig. 1 shows an operation of ubiquitous wireless Internet access through ad-hoc networks.

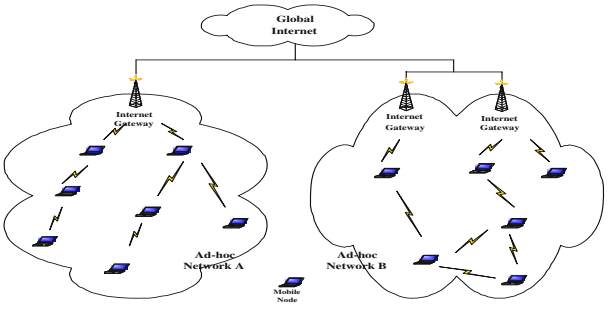


Fig. 1. Ubiquitous Internet connectivity through ad-hoc networks

When an AMN needs a connection to the Internet, it should connect the nearest or best Internet gateway. Key issue for supporting Internet connectivity is IG discovery, for which three approaches have been proposed: proactive, reactive, and hybrid approaches [6,7,8]. According to the proactive approach, IGs advertise their presence by sending an advertisement message on the ad-hoc network. It provides for good connectivity and low delay via frequent broadcasts of current IG information with the expense of high control message overheads. By the reactive approach, AMNs broadcast a route request message to discover IGs. On receipt of this request, IGs sends reply messages back to the requesters. Although it achieves low discovery overhead, the reactive scheme may increase route discovery delay. The hybrid approach combines the proactive and reactive schemes, reaping the best of both schemes: good connectivity and low delay. After finding multiple relay routes, AMNs select the best IG to communicate with Internet hosts outside the ad-hoc networks.

3 LAID: Load-Adaptive Internet Gateway Discovery

In dynamic network environment, existing IG discovery schemes are only suitable for certain network configurations. Performance and scalability problems may come to the surface because of the fixed proactive areas in hybrid scheme that do not reflect dynamic network conditions [7]. The primary challenge in the design of a hybrid approach is how to determine the optimal proactive area. The loss rate and delay are decreased by increasing the area, but it will pay more in packet overhead to maintain routes in a larger area. The routing overhead is reduced by decreasing the area, but it may pay more in delay and experience higher loss rates [8]. Thus, fixed value of proactive area is not the best choice for all levels of network conditions. To achieve optimal performance, we propose a Load-Adaptive hybrid IG Discovery (LAID) scheme which dynamically resizes the range of proactive IG advertisements. Our protocol adapts its behavior to current network situations such as the number of IGs or the number of AMNs that need global communication. In this section, we compute the proactive area and describe IG selection method.

3.1 Proactive Area Measurement

Internet gateways (IGs) periodically announce their presence in an ad-hoc network by broadcasting Internet Gateway Advertisement Messages (IGAMs) with their information within periodic intervals. To prevent the flooding of the advertisements, these advertisements are limited within n-hop neighborhood using a time-to-live (TTL) field. This range determines the IG's discovery scope, called a proactive area, which is dynamically adjusted by our adaptive Internet gateway discovery protocol. The initial value of the proactive area is computed as follows:

$$Proactive\_area(\Psi) = r \cdot \frac{N}{N_{IG} \cdot N_i} \tag{1}$$

where Ψ is a proactive area by TTL, r is a given radius $r = \frac{N}{N_i}$, N is the total number of nods, N_i is the number of nodes assigned to an IG, and N_{IG} is the number of Internet gateways. For example, if the number of total nodes is 50, the number of nodes assigned to an IG is 25, the number of IGs is 2, and r is 2, then from equation (1), the proactive area (Ψ) is 2.

The proactive range expands or shirks according to network traffic which is estimated by IGs during the time interval $(\Delta_{t1}, \Delta_{t2})$. To compute the offered load, we suppose that the average traffic arrival rate is λ and the average traffic duration is τ per time interval, and we consider a periodic time interval of length $\Delta_t\{>1\}$ between two successive estimations. The number of path connected to the IG over this interval is $n(\Delta_t)$ and the amounts to be generated are $\lambda_1 \cdot \tau_1, \lambda_2 \cdot \tau_2, \cdots, \lambda_n(\Delta_t) \cdot \tau_n(\Delta_t)$. For simplicity, let us also assume that the packet sizes are independent. Then over the interval (Δ_{t1},Δ_{t2}), the offered load is given by

$$\rho = \sum_{i=1}^{n(\Delta_t)} \lambda_i \cdot \sum_{i=1}^{n(\Delta_t)} \tau_i = \sum_{i=1}^{n(\Delta_{t1}, \Delta_{t2})} \lambda_i \tau_i \tag{2}$$

To obtain an up-to-date route from IGs, it is desirable to reduce the time interval. However, short interval will increase the overhead of the protocol in terms of bandwidth waste and power consumption at AMNs [8]. We dynamically adjust the beacon interval according to the network conditions, e.g., node mobility and traffic. It allows our protocol to react to the changes in the network on time.

To avoid unnecessarily frequent resizing of the proactive area, we introduce two threshold: max threshold ($\gamma_{\max}$) and min threshold ($\gamma_{\min}$), which are based on the traffic load given by (2) and always $\gamma_{\max} > \gamma_{\min}$. If the estimated value is larger than the max threshold ($\rho > \gamma_{\max}$), the size is incremented by 1. Similarly, if the estimation is less than the min threshold ($\rho < \gamma_{\min}$), the size is decremented by 1. In other words, if $\Psi(now)$ is the current proactive area, the next proactive area becomes $\Psi(now + \Delta_t) = \Psi(now)$ or $\Psi(now + \Delta_t) = \Psi(now) \pm 1$. The $\gamma_{\max}$ and $\gamma_{\min}$ are $\rho + \rho \cdot 0.05$ and $\rho + \rho \cdot (-0.05)$, respectively.

As shown in Fig. 2, AMNs within the TTL (e.g. the proactive range) receive the periodic IGAM messages from IGs. If they are out of the range, the AMNs

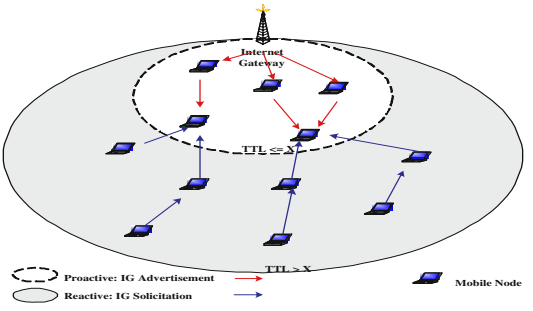


Fig. 2. Illustration of load-adaptive Internet gateway discovery

broadcast Internet gateway Request messages (IGRQ). AMNs inside the proactive area of an IG respond with Internet gateway Response messages (IGRP) to the soliciting AMNs or relay to IGs. On receipt of IGRQ messages, IGs send an IGRP message which has the IGs' prefix and information back to the soliciting AMNs. Data packets within the proactive area are routed by means of proactive routing protocols. Routes from a source node to the edge of the proactive area are reactively maintained. The load-adaptive hybrid IG discovery scheme provides efficient and fast discovery of IGs by the integration of three traditional Internet gateway discovery schemes.

3.2 Internet Gateway Selection Method

After finding multiple IGs, AMNs should select the best IG to communicate with Internet hosts outside the ad-hoc networks. The selection of the IG can be categorized into two cases: when an ad-hoc node is entered into the ad-hoc network at the first time and when a node performs a handover to new IG. The handover occurs when a moving ad-hoc node receives the IGAMs or when the ad-hoc node is disconnected from the previously registered IG. Although there exist several IG discovery schemes in ad-hoc networks with Internet connectivity, most of them regard the shortest path with minimum hop counts as a major IG selection metric. Also, they did hardly concern multiple IGs. When AMNs are available some IGs and the selection of IG is only based on the shortest-path, the shortest-path algorithm does not perform very well. The poor performance of the shortest-path algorithm is not surprising, since the metric do not consider load of IGs and/or quality of a path during route setup. Hence they cannot fairly distribute the load on the different IGs and may lead to higher packet dropping rate. That is because the ad-hoc nodes want to get the qualified various services from the Internet. To reach this goal, we consider load of IG to guarantee the quality of network connection for user. This information is used by the source node to select the proper IGs. In our proposed load-adaptive discovery approach, Internet gateway selection is regulated by a distributed redirecting selection mechanism based on load of IG, which redirects the selected IG with heavy offered load into different IGs with less offered load to reduce and distribute data traffic over the network.

4 Simulation Model and Performance Evaluation

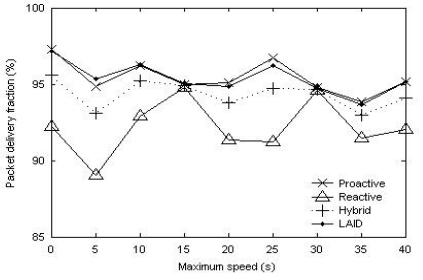
In this section, we evaluate the proposed scheme, compare it with existing IG discovery schemes, and analyze the analytical overhead of the discovery approaches.

4.1 Simulation Model

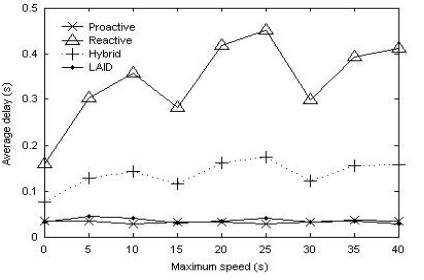
The simulations were performed using ns-2 [4]. In order to support wireless LAN in the simulator, the Distributed Coordination Function (DCF) of IEEE 802.11 is adopted as MAC layer protocol. As a mobility model, we use the random waypoint model in rectangular field where a node starts its journey from a random location to a random destination with a randomly chosen speed. The size of network is 800 m $\times$ 800 m and the number of mobile nodes is 50 in simulations. Constant bit rate (CBR) traffic with four packets per second and packet size of 512 bytes are used. We use the number of source nodes of 10 and 20. Simulations are run for 300 seconds. For fair comparisons, all discovery protocols use the same set of mobility and traffic. On stationary, one IG is located in the middle of the grid [i.e., coordinate (400, 400)] for the first three simulation scenarios. In the second simulation scenario, two IGs are located in the coordinates (1, 400) and (799, 400), respectively. An AMN uses modified AODV protocol [2] to communicate with its peers and to access wired networks through an IG. To manage AMNs' mobility between ad-hoc networks, AMNs as well as IGs run MIP [3], where MIP FA and HA are hosted in the IGs.

4.2 Simulation Results

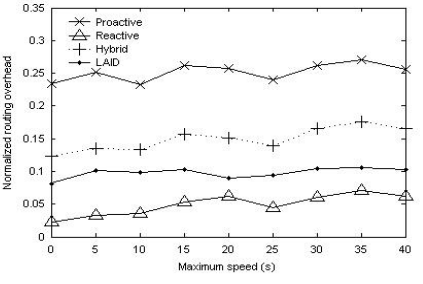
To compare IG discovery approaches in the case of a single IG, a set of simulations has been performed in terms of three metrics: packet delivery fraction, average end-to-end delay, and normalized routing overhead. Various mobility scenarios have been simulated to understand their effects. Fig. 3 shows the simulation results for the proactive, reactive, hybrid, and load-adaptive approaches. Both proactive and reactive approaches have specific advantages and disadvantages that make them suitable for certain types of scenarios. In the proactive approach, the overhead for Internet connectivity increases as IGs broadcast periodic IGAM messages during the intervals that are flooded through the whole. The proactive scheme costs more overhead, but allows for good connections and low delay because it instantly knows better paths to IGs. In contrast, the reactive scheme incurs fewer overhead than the proactive approach, because AMNs request IG information by sending out IGRQ messages only when necessary. However, whenever there is a need for sending a packet, AMNs must find IGs if the IGs are not already known. This IG discovery process may result in considerable delay. Thus, it causes longer packet delay and lower packet delivery fraction. Fig. 3 shows that the hybrid and load-adaptive schemes are a compromise of proactive and reactive schemes. The hybrid and load-adaptive approaches minimized the disadvantages, and maximized the advantages of the two combined approaches. The load-adaptive IG discovery (LAID) scheme enables lower packet



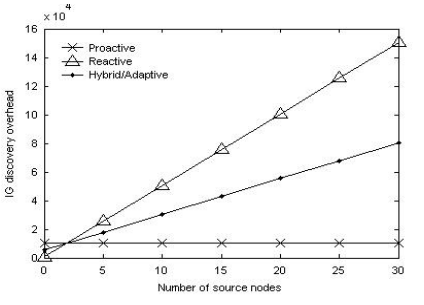
(a) Packet delivery fraction



(b) Average delay



(c) Normalized routing overhead

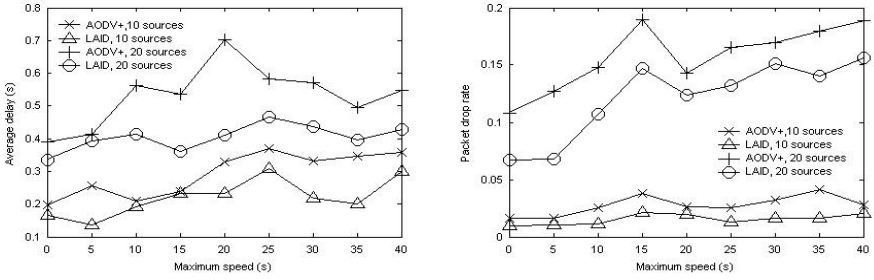


(d) Discovery overhead for analytic model

Fig. 3. Simulation results in IG discovery scenario: a) Packet delivery fraction, b) Average delay, c) Normalized routing overhead, and d) IG discovery overhead about analytic model

delay compared to the reactive and hybrid approaches, and less overhead compared to the proactive and hybrid approaches. The dynamic resizing of proactive ranges can help to reduce excessive traffic otherwise by proactive approach during low mobility and traffic periods, by confining the advertisement traffic to a limited area. Under high traffic and mobility, our LAID will extend the proactive area to farther disseminate information about available IGs. Increased proactive area ends up with reduced route acquisition time and bandwidth loss. The LAID will scale well with network size and mobility.

For IG selection, we compare the performance of proposed LAID using load-based selection scheme and the AODV+ [1] using shortest-path selection algorithm in terms of average end-to-end delay and packet drop probability. For a scenario involving burst traffic, we assume a set of AMNs is initially requesting connections to IGs. The second experiment (Fig. 4) reports average delay and packet drop rate under various speeds. The average delay is defined by delay from the source node to the IG. Fig. 4(a) shows that our LAID achieves lower average delay than AODV+. Under higher traffic load, the average delay is further improved compared to the AODV+ using shortest path selection algorithm. A new connection is blocked, if there is no IG bandwidth available when it is



(a) Average delay (b) Packet drop rate

Fig. 4. Effects of varying mobility in IG selection: a) Average delay and b) Packet drop rate

needed. Fig. 4(b) plots the packet drop probability versus mobility at IGs. As the number of source nodes increases, the AODV+ and the LAID drop a large fraction of the packets. Simulation result has shown that the LAID gives a lower packet drop rate than AODV+. That is because our redirecting selection mechanism in LAID can redirect the IG with heavy traffic to the IGs with light traffic. When the number of source nodes increase, the AODV+ does not perform well, since the metric simply selects an IG with shortest-path without regard to their density. The LAID considers offered load of IG, and uses redirecting selection mechanism. Thus, it performs better performance than AODV+. Our selection method might increase the throughput because of redirecting the requests originated by the AMNs at the boundary of radius through the neighborhood IGs.

4.3 Comparison of Internet Gateway Discovery Approaches by Analytic Model

We analyze the three IG discovery approaches: proactive, reactive, and hybrid/adaptive. Our analysis model assumes that new traffic generated by the hosts connected to mobile nodes follows Poisson distribution and is generated independently of each other. All hosts have the same traffic generation pattern.

When an ad-hoc source tries to discover a route towards a fixed node, it should find an IG. In a proactive approach, IGs will periodically broadcast IGAM messages to an ad-hoc network to advertise their presence. Therefore, the overhead of proactive schemes includes hello messages for route update plus the messages sent out by IGs themselves. Total overhead in the number of messages required by the proactive approach can be expressed as follows:

$$\Theta_P = N\left(N_{IG}\cdot\lambda_{IGAM}\cdot\Delta_t + P_{Ph}(\Delta_t) + \frac{1}{\mu}\alpha_N\right) \tag{3}$$

where Θ_P is the overhead of proactive approach, N is the number of nodes, N_{IG} is the number of IGs, λ_{IGAM} is the rate at which IGAM messages are emitted by

IGs, $P_{Ph}(\Delta_t)$ is the number of the hellos packets by a AMN per a time interval, and $\frac{1}{\mu}\alpha_N$ is a route maintenance cost which is called $\hat{\beta}_P$, where μ is average communication link lifetime and α_N is the number of active neighbor nodes. We assume that link lifetime is independent of each other and are exponentially distributed. The discovery overhead of the proactive approach is independent of the number of sources sending data packets to the same IG.

Similarly, in the reactive approach a source willing to communicate with a host in the fixed network will first attempt to contact it within the ad-hoc network. If no answer is received after a network-wide search, then the source tries to find a route towards the Internet. The source wants to reactively discover an IG there is an overhead which includes the IGRQ broadcast messages, plus IGRP reply messages from every IG to the source. The overhead of the reactive IG discovery by one source can be computed as follows:

$$\Theta_R = N\left(N_S \cdot \lambda_{IGRQ} \cdot \Delta_t(R) + P_{Rh}(\Delta_t(R)) + \frac{1}{\mu}\alpha_L h\right) \tag{4}$$

where Θ_R is the overhead by the reactive approach, N_S is the number of source nodes communicating with a host in the Internet, λ_{IGRQ} is the sum of route requests and replies during the time interval $(\Delta_t(R))$ for reactive requests, $P_{Rh}(\Delta_t(R))$ is the number of hello packets emitted by a AMN for $\Delta_t(R)$ second, and $\frac{1}{\mu}\alpha_L h$ is route maintenance overhead which is called $\hat{\beta}_R$, where α_L is the number of active links and h is a hop count. If link layer is used to detect link failures, P_{Rh} is 0. Route lifetime follows an exponential distribution with a mean route lifetime of μ/h. The average rate of route failures is given by h/μ. The discovery overhead of the reactive approach is proportional to the number of active routes in the network. Therefore, reactive overhead increases with the number of sources and destinations in the network.

By a hybrid/adaptive approach, IGs periodically send IGAM messages within a certain range which is determined by a proactive area. Sources in that range behave as in a proactive approach, and those beyond that range behave as in a reactive approach. The hybrid/adaptive IG discovery scheme has the constituent overhead of proactive and reactive approaches. For sources outside the area covered by the IGAM messages, the overhead will be similar to that of the reactive approach. Thus, the overhead of the hybrid/adaptive approach is computed as follows:

$$\begin{aligned}\Theta_H = N_{TTL}^{IG}\left(N_{IG} \cdot \lambda_{IGAM} \cdot \Delta_t + P_{Ph}(\Delta_t) + \hat{\beta}_P\right) + \\ N_{N-TTL}\left(N_S \cdot \lambda_{IGRQ} \cdot \Delta_t(R) + P_{Rh}(\Delta_t(R)) + \hat{\beta}_R\right)\end{aligned} \tag{5}$$

where Θ_H is the overhead of hybrid/adaptive approach, N_{TTL}^{IG} is the number of nodes in the TTL range from an IG, and N_{N-TTL} is the number of nodes for each source outside the proactive area. Hence only $N - TTL$ nodes in the path revert to reactive discovery. In this scheme, used hello packets are not part of the discovery overhead. That is because update packets are generated, transmitted, and received by the link layer (in that case, $P_{Ph} = 0$ and $P_{Rh} = 0$).

Fig. 3(d) shows a graph for this analytic model. We note that different proactive range leads to different performance of the hybrid/adaptive scheme, and the optimal TTL is dependent on several network conditions. As our analytical model has estimated, our adaptive approach can achieve a good trade-off between the efficiency of the protocol in terms of signaling overhead.

5 Conclusions

We have proposed a load-adaptive hybrid Internet gateway discovery approach named LAID. Our load-adaptive hybrid Internet gateway discovery approach dynamically adjusts the proactive area based on the offered load. We investigate the performance of LAID under various network conditions. Our simulation study shows that the proposed load-adaptive discovery approach outperforms other existing approaches. Also, the load-based redirecting selection scheme provides load-balancing and reduces average delay and packet drop rate.

Acknowledgment

This work was supported by grant No. R01-2005-000-10267-0 from Korea Science and Engineering Foundation in Ministry of Science and Technology.

References

1. A.Hamidian, "A Study of Internet Connectivity for Mobile Ad Hoc Networks in NS 2," Master's thesis. Department of Communication Systems, Lund Institute of Technology, Lund University. January 2003.
2. C.E. Perkins, E.M. Belding-Royer, and S. Das, "Ad-Hoc On-Demand Distance Vector Routing," RFC 3561 in IETF, July 2003.
3. C.E. Perkins, "IP Mobility Support," RFC 3311 in IETF, August 2002.
4. K. Fall and K. Varadhan, Eds., "ns Notes and Documentation," 2003; available from *http://www.isi.edu/nsnam/ns/*.
5. M. Ilyas, The Handbook of Ad-Hoc Wireless Networks, CRC PRESS, 2002.
6. M. Ghassemian, P. Hofmann, C. Prehofer, V. Friderikos, A.H. Aghvami, "Performance Analysis of Internet Gateway Discovery Protocols in Ad Hoc Networks," WCNC2004.
7. P. M. Ruiz and A. F. Gomez-Skarmeta, "Maximal Source Coverage Adaptive Gateway Dis-covery for Hybrid Ad Hoc Networks," ADHOC-NOW 2004.
8. P. Ratanchandani and R. Kravets, "A hybrid approach to internet connectivity for mobile ad hoc networks," Proceedings of WCNC 2003, March 2003.
9. R. Wakikawa, J.T. Maline, C.E. Perkins, A. Nilsson, and A.H. Tuominen, "Global Connec-tivity for IPv6 Mobile Ad-hoc Networks," IEFT Internet-Draft, draft-wakikawa-manet-globalv6-03.txt, October 2003.
10. U. Jonsson, F Alriksson, T. Larsson, P. Johnasson, and G.Q. Maguire, "MIP-MANET: Mo-bile IP for Mobile Ad-hoc Networks," Mobihoc, Aug. 2000.

Fast Restoration of Resilience-Guaranteed Segments Under Multiple Link Failures in a General Mesh-Type MPLS/GMPLS Network

Jong-Tae Park, Min-Hee Kwon, and Jung-Ho Kwon

School of Electrical Engineering and Computer Science
Kyungpook National Univeristy
702-701 Daegu, Korea
{jtpark, minhi, jkwon}@ee.knu.ac.kr

Abstract. Network resilience in MPLS/GMPLS networks has been receiving considerable attention. Most of the previous research on GMPLS recovery management has focused on efficient routing or signaling methods from single failures. Multiple simultaneous failures, however, may occur in a large-scale complex GMPLS network infrastructure. In this article, we derived the condition to test the existence of backup segments which satisfy the resilience constraint under multiple link failures, in a general mesh-type MPLS/GMPLS network. A decomposition theorem and a backup segment construction algorithm have been developed for the fast restoration of resilience-guaranteed backup segments, for the primary path with an arbitrary configuration. Finally, the simulation has been to show the efficiency of the proposed approach.

1 Introduction

Network resilience in multi-protocol label switching (MPLS) and generalized MPLS (GMPLS) networks has been receiving considerable attention as of late, in research and standardization communities [1]. Resilience is the ability to recover from network component failures. Most of the previous research on MPLS/GMPLS recovery man-agement has focused on efficient routing or signaling methods from single failures [2]. Multiple simultaneous failures often occur in large-scale GMPLS network infrastructures [3].

Although the importance of research regarding recovery from multiple failures in the MPLS/GMPLS networks has been addressed in IETF standards [4, 5], there have been relatively few attempts to investigate these issues [6, 7]. Lee and Griffith [6] presented a hierarchical scheme to resolve multiple failures in the GMPLS networks. Their scheme assigns a high-prioritized failed primary path to the pre-reserved path and the other failed path to the shared backup paths. They, however, did not consider segment restoration. Clouquer and Grover [7] analyzed the availability of span-restorable mesh networks under dual-failure scenarios. Their approach, however, is limited to dual failures.

In contrast to these findings, we have recently presented a mechanism for recovery management which can handle multiple simultaneous link failures, while

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 359–368, 2006.

satisfying resilience constraints [8]. In this article, we have extended the approach in [8] to the general mesh-type MPLS/GMPLS network by removing the topological constraints of the primary paths. In order to do that, first we derived the upper bound on the number of link failures in order to guarantee the existence of backup segments, by extending the work in [8].Next, we developed a decomposition theorem which enables nodes in the primary path, with an arbitrary configuration, to be decomposed into a set of segments, such that each segment has monotonic properties. Based on these theorems, we developed a fast backup segment construction algorithm for the dynamic restoration of resilience-guaranteed segments, under multiple link failures, in a MPLS/GMPLS network with arbitrary configurations. The results can be useful for cases of fast segment recovery in GMPLS network, which are currently being studied in IETF [9].

2 Basic Testing Conditions for Fast Recovery from Multiple Failures

In this section, we derive the conditions to test the existence of backup paths, which satisfy the resilience constraints in a special mesh-type MPLS/GMPLS network. Before proceeding further, we need to introduce the resilience model for segment recovery in the MPLS/GMPLS networks. In [2], the resilience attributes are used to determine the behavior of the LSPs when failures occur. For example, they determine how the failed LSP should be rerouted when segments of its path fail. In [8], we have defined the concept of k-protection to formally express the resilience constraints. A primary path is said to have k-protection if any subsets of the path, consisting of (k-1) adjacent nodes and k-links connecting these nodes is protected by the backup paths. The protection region of a primary path is defined as a segment of the primary path, which is protected by the backup paths. The path domain denotes the set of nodes in the primary path, and the non-path domain implies the set of nodes not in the primary path which can be used to construct backup paths. Path resilience is defined as the normalized ratio of the protected region of a path, and it is found to effectively represent various protection modes used in both protection mechanism and restoration mechanism in the GMPLS network [8].

In order to generate backup path candidates, it is assumed that neither links nor nodes in the protection region of the primary path are used to construct feasible backup paths, except for the beginning and end nodes of the protection region as in [8]. By applying the rule in [8] to segment recovery, we have two types of backup segment candidates: Type-1 and Type-2. Type-1 consists of two nodes; the beginning node and end node of the protection region, and the direct link that connects them. The Type-2 candidate starts at the beginning node of the protection region, and the intermediate nodes from the non-path domain, and finishes at the ending node of the protection region.

Let $\mathbf{P}$ and $\mathbf{R}$ denote the path domain and non-path domain, respectively. The primary path of the MPLS/GMPLS network is said to have monotonic properties if it has either a monotonic decreasing or increasing sequence of degrees. Let f

be defined as a function $f: \mathbf{P} \rightarrow \mathbf{R}$ such that for a given path, f is the number of direct links from a node P in $\mathbf{P}$ to nodes in $\mathbf{R}$.

Theorem 1. For a general mesh-type MPLS/GMPLS network, let us assume that there exists a sub-path $< P_1, P_{k+1}, P_{2k+1}, \ldots, P_{k(2m-1)+1}, P_{2km+1} >$ of the primary path, such that the degree of the nodes of the sub-path for the links from the path domain $\mathbf{P}$ to the non-path domain $\mathbf{R}$ is monotonic decreasing, i.e. $f(P_{ik+1}) \geq f(P_{(i+1)k+1})$ if i = 0, 1, 2, ..., 2m-1, and the subgraph, which consists of only nodes in $\mathbf{R}$ is connected. Suppose that there are no direct links between any non-adjacent nodes of the primary path, and n = 2km + 1 for some positive integers k and m. Then, the sub-path $< P_1, P_{k+1}, P_{2k+1}, \ldots, P_{k(2m-1)+1}, P_{2km+1} >$ of the primary path has *k-protection* even though $(\zeta - 1)$ number of links from $\{P_1, P_{k+1}, P_{2k+1}, \ldots, P_{k(2m-1)+1}, P_{2km+1}\}$ to $\mathbf{R}$ fails, where $\zeta = \sum_{j=1}^{m} f(P_{(2j-1)k+1})$.

Proof. It can be proved by induction. Let $\mathbf{P'}$ be $\{P_1, P_{k+1}, P_{2k+1}, \ldots, P_{k(2m-1)+1}, P_{2km+1}\}$. Let ζ be the minimum number of link failures from the subset $\mathbf{P'}$ to the non-path domain $\mathbf{R}$, which would not permit the construction of any primary paths with k-protection. Let us consider the case m = 1. In this case, $\mathbf{P'}$ is equal to $\{P_1, P_{k+1}, P_{2k+1}\}$. Thus, if all links, i.e. $f(P_{k+1})$ from the node P_{k+1} to the nodes in $\mathbf{R}$ fail, there is no way to construct any k-protection backup paths. Since $f(P_{ik+1}) \geq f(P_{(i+1)k+1})$ if $i = 0, 1, 2, \ldots, 2m-1$, the number $f(P_{ik+1})$ is a minimum number of link failures from $\mathbf{P'}$ to the non-path domain $\mathbf{R}$, which would not allow for the construction of a k-protection backup path. Therefore, ζ is equal to $\zeta = \sum_{j=1}^{m} f(P_{(2j-1)k+1})$ for m=1.

Now, let us assume that the minimum number of link failures from $\mathbf{P'}$ to the non-path domain $\mathbf{R}$ is equal to $\sum_{j=1}^{m} f(P_{(2j-1)k+1})$ for m = n. For m = n+1, if $f(P_{(2n-1)k+1+2k})$ links from the node $P_{(2n-1)k+1+2k}$ to the nodes in $\mathbf{R}$ fail, there is no way to construct a k-protection backup path for the sub-path $< P_{(2n-1)k+1+k}, P_{(2n-1)k+1+3k} >$, *i.e.*, $< P_{2nk+1}, P_{2(n+1)k+1} >$. Since $f(P_{(2n-1)k+1}) \geq f(P_{(2n-1)k+1+2k})$, we know that the minimum number of link failures becomes $\sum_{j=1}^{n} f(P_{(2j-1)k+1}) + f(P_{(2n-1)k+1+2k})$. Since $f(P_{(2n-1)k+1+2k})$ is equal to $f(P_{(2(n+1)-1)k+1})$, this leads to the fact that ζ is equal to $\sum_{j=1}^{n+1} f(P_{(2j-1)k+1})$ for m = n+1. Since any one link from the ζ number of link failures, if it does not fail, could be used to construct a k-protection backup path, $(\zeta - 1)$ is the maximum number of allowable link failures which guarantees the existence of a k-protection backup path. This completes the proof. ■

Example 1. Let us consider a GMPLS network as shown in Fig. 1, where there are 6 nodes $\{P_1, P_2, \ldots, P_6\}$ in the path domain $\mathbf{P}$ and 5 nodes in the non-path domain $\mathbf{R}$. There are no direct links between any non-adjacent nodes of

the primary path, and every node in the non-path domain is connected. Let **S** denote the sub-path $< P_1, P_3, P_5 >$. Let us construct a 2-protection primary path using **S**. Since $f(P_1) = 3, f(P_3) = 2, f(P_5) = 1$, the sub-path **S** has the monotonic property. Since m = 1 and k = 2, $\sum_{j=1}^{n} f(P_{(2j-1)k+1}) = f(P_3)$. Thus, $\zeta = f(P_3) = 2$.

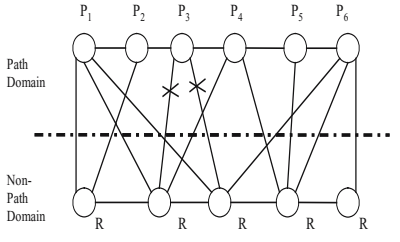


Fig. 1. A simple GMPLS network with 2 failures for the sub-path S_1

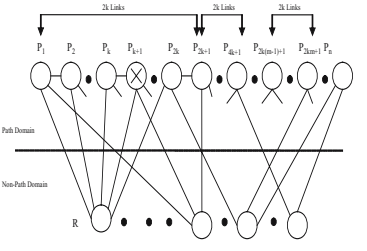


Fig. 2. A general k-protection primary path with monotonic properties

Therefore, according to Theorem 1, when the number of link failures is greater than $(\zeta - 1)$, i.e. 1, it may be not possible to construct a 2-protection backup path for the primary path using the sub-path **S**, as shown in Fig. 1. Now, let us consider the more general case in which all nodes in primary path have a monotonic decreasing sequence of degrees.

Theorem 2 (Optimality Theorem). For a general mesh-type MPLS/GMPLS network, let us assume that the degree of the nodes in the primary path $< P_1, P_2, \ldots, P_n >$ for the links from the path domain **P** to the non-path domain **R** is monotonic decreasing, i.e. either $f(P_i) \geq f(P_j)$ if $i \leq j$ for i, j = 1, 2, ..., n, and that the sub-graph, which consists of the only nodes in **R**, is connected. Suppose that there are no direct links between any non-adjacent nodes of the primary path. Then, the primary path with n nodes has k-protection even though $(\zeta - 1)$ number of links from **P** to **R** fails, where $\zeta = \sum_{i=1}^{k} \sum_{j=1}^{m} f\left(P_{(2j-1)k+i}\right)$ and n < N, n = 2km+1+ℓ for ℓ = 0, 1, ..., k.

Proof. Without loss of generality, let us assume that the primary path has the configurations, as shown in Fig. 2, in which the primary path is partitioned into groups of 2k links and the remaining ℓ - 1 links. Let us partition the nodes of the primary path into a collection of subsets of nodes such that $\mathbf{P} = \mathbf{P}_1 \cup \mathbf{P}_2, \ldots, \cup \mathbf{P}_k$, where $\mathbf{P}_1 = \{P_1, P_{k+1}, P_{2k+1}, \ldots, P_{k(2m-1)+1}, P_{k(2m-1)+1}\}$, $\mathbf{P}_2 = \{P_2, P_{k+2}, P_{2k+2}, \ldots, P_{k(2m-1)+2}, P_{k(2m-1)+2}\}$, ..., $\mathbf{P}_k = \{P_k, P_{2k}, P_{3k}, \ldots, P_{k(2m-1)+k}\}$.

Then, according to Theorem 1, we know that the minimum number of link failures which would not allow to build a k-protection backup path for the subset $\mathbf{P}_i$ for i = 1, 2, ..., k, is $\sum_{j=1}^{m} f(P_{(2j-1)k+1})$. Since there are k subsets which are

built from the partition, the total number of link failures from **P** to **R**, which would not permit the construction of the k-protection backup path, becomes the sum of link failures associated with subsets $\mathbf{P}_i$ for i = 1, 2, ..., k, i.e., $\sum_{i=1}^{k} \sum_{j=1}^{m} f\left(P_{(2j-1)k+i}\right)$. ■

$(\zeta - 1)$ is the maximum number of allowable link failures which guarantees the construction of a k-protection backup path under any *multiple* failure occurrences in a MPLS/GMPLS network. It should be noted that we can prove the monotonic increasing case if we count the indices of the nodes in the primary path from the ending node to the beginning node.

3 Decomposition Algorithm for Fast Restoration of Resilience-Guaranteed Backup Segments

We present a fast segment restoration algorithm for general mesh-type MPLS/GMPLS networks, whereby nodes in the primary path have an arbitrary configuration. The basic idea is that we decompose the set of primary nodes into a disjointed collection of segments such that each segment has a *monotonic property*, i.e. either a monotonic increasing or decreasing sequence of degrees. When a beginning node receives the failure notifications from a failed segment, it first tests the availability of a backup segment for the failed segment, and then it rapidly restores the failed segment if the testing condition is satisfied.

For segment **S** of the primary path **P**, let $\zeta(\mathbf{S})$ be defined as the minimum number of link failures which would not allow the construction of the k-protection backup path in **S**. Let $U(X_i)$ be a unit step function, such that it becomes one if there exists a direct link from node X_i to node X_{i+k}, otherwise zero, where $X_i, X_{i+k} \in \mathbf{S}$ for i = 1, 2, ..., (n-k), and $\mathbf{S} = \{X_1, X_2, \ldots, X_n\}$. First, we derive the upper bound of the number of failed links which guarantees the existence of the Type-1 backup segments, which satisfy the resilience constraint, i.e. the resilience-guaranteed backup segments.

Lemma 1. *For a general mesh-type MPLS/GMPLS network, let us assume that the degree of the nodes in the segment* $< X_1, X_2, \ldots, X_n >$ *for the links from the segment* $\boldsymbol{S}$ *to the non-path domain* $\boldsymbol{R}$ *is zero, i.e.* $f(X_i) = 0$ *for i = 1, 2, ..., n. Then, the segment with n nodes has k-protection even though* $(\zeta(\boldsymbol{S}) - 1)$ *number of links of the nodes in* $\boldsymbol{S}$ *fails, where* $\zeta(S) = \sum_{i=1}^{n-k} U\left(X_i\right)$ *and* $n \geq k$.

Proof. The Type-1 candidate can only be constructed by using links which directly connect two nodes in the segment **S**. Since there are n nodes with $n \geq k$ in the segment **S**, we can construct a backup segment with length k from X_i to $X_{(i+k)}$, if a direct link exists between X_i to $X_{(i+k)}$ for i = 1, 2, ..., (n-k). Thus, $\zeta(\mathbf{S})$ is equal to $\sum_{i=1}^{n-k} U\left(X_i\right)$. If any one link is non-faulty from these direct links, we can construct a k-protection backup segment.

Now, let us consider the segment recovery in a MPLS/GMPLS network, in which the nodes $< X_1, X_2, \ldots, X_n >$ of the segment $\mathbf{S}$ have the monotonic property. Let the total number of link failures, which are not in $\mathbf{P}$ and are incident to the nodes in $\mathbf{S}$, be denoted as $\varepsilon(\mathbf{S})$.

Theorem 3. For a general mesh-type MPLS/GMPLS network, suppose that there exists a segment $\mathbf{S}$ of the primary path $\mathbf{P}$. Suppose that the nodes in $\mathbf{S}$ have a monotonic property. Then, $\mathbf{S}$ has *k-protection* as long as there exists n nodes in $\mathbf{S}$ with $n \geq k$ and $\varepsilon(\mathbf{S}) \leq (\zeta(\mathbf{S}) - 1)$ where

$$\zeta(S)=\begin{cases} \sum_{i=1}^{k}\sum_{j=1}^{m} f\left(X_{(2j-1)k+i}\right) + \sum_{i=1}^{n-k} U(X_i) \;\; for\ \ell \leq k \\ \sum_{i=1}^{k}\sum_{j=1}^{m} f\left(X_{(2j-1)k+i}\right) + \sum_{j=1}^{\ell-k} f\left(X_{(2m+1)k+j}\right) + \sum_{i=1}^{n-k} U(X_i) \;\; for\ \ell > k \end{cases} \tag{1}$$

and n = 2km+ℓ for ℓ = 0, 1, …, (2k-1), and m is a non-negative integer.

Proof. We should consider two types of backup candidates: Type-1 and Type-2. First, for Type-1, according to Lemma 1, we know that $\zeta(\mathbf{S})$ is equal to $\sum_{i=1}^{n-k} U(X_i)$. Next, let us consider the Type-2 candidate. Since n ≥ k, the k-protection backup path can be constructed within $\mathbf{S}$. Furthermore, since the nodes in $\mathbf{S}$ have a monotonic decreasing sequence of degrees, we know, according to Theorem 2, that if $\varepsilon(\mathbf{S}) \leq (\zeta(\mathbf{S}) - 1)$ with $\zeta(S) = \sum_{i=1}^{k}\sum_{j=1}^{m} f\left(X_{(2j-1)k+i}\right)$, it is possible to construct a k-protection backup segment in $\mathbf{S}$ for n = 2km + 1 + ℓ, ℓ = 0, 1, …, k. This is because a segment itself can be treated as a primary path with a length of n.

In the case of ℓ > k, it should be noted that the links from the nodes $X_{(2m-1)k+1+2k}$ can be used for the construction of the k-protection backup segments. Therefore, since $X_{(2m-1)k+1+2k}$ is equal to $X_{(2m-1)k+1}$, all links from the node $X_{(2m-1)k+1}$ to the nodes in $\mathbf{R}$ should be accounted for when calculating the minimum number of link failures. Similarly, the links from the set of nodes $\{X_{(2m+1)k+2}, \ldots, X_{(2m+1)k+(1-k)}\}$ to the nodes in $\mathbf{R}$ should be accounted for when calculating the minimum number of link failures. Thus, $\zeta(S) = \sum_{i=1}^{k}\sum_{j=1}^{m} f\left(X_{(2j-1)k+i}\right) + \sum_{j=1}^{\ell-k} f\left(X_{(2m+1)k+j}\right)$ for ℓ > k. Finally, by summing up the results from Type-1 and Type-2, the proof is completed. ■

For a general mesh-type MPLS/GMPLS network, it is assumed that the backup path can only be constructed within a segment, and not across different segments. Let the primary path $\mathbf{P}$ decompose into a collection of disjointed segments $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m\}$ where $\mathbf{P} = \bigcup_{i=1}^{m} \mathbf{S}_i$. Then, the primary path is defined to have k-protection with m segments $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ if each segments $\mathbf{S}_i$ for i = 1, 2, …, m has k_i-protection and $k = \sum_{i=1}^{m} k_i$. Let $\zeta(\mathbf{S}_i)$ be defined in Equation(1).

Theorem 4 (Decomposition Theorem). Let the primary path $\mathbf{P}$ with n nodes decompose into a set of m disjointed segments $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m\}$ such that $\mathbf{P} = \bigcup_{i=1}^{m} \mathbf{S}_i$ and that the segment $\mathbf{S}_i$ for i = 1, 2, ..., m has a monotonic property. Then, the primary path has k-protection with m segments as long as $\varepsilon(\mathbf{S}_i) \leq (\zeta(\mathbf{S}_i) - 1)$ and $k \leq \sum_{i=1}^{m} k_i$. Here, $\zeta = \sum_{i=1}^{m} \zeta(S_i)$ is the minimum number of link failures which would not allow the primary path to have k-protection with m segments $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$.

Proof. According to Theorem 3, if $\varepsilon(\mathbf{S}_i) \leq (\zeta(\mathbf{S}_i) - 1)$ for i = 1, 2, ..., m, the segment $\mathbf{S}_i$ has k_i-protection backup segments since it has a monotonic property. Furthermore, since $\mathbf{P} = \bigcup_{i=1}^{m} \mathbf{S}_i$ and $k \leq \sum_{i=1}^{m} k_i$, the primary path has k-protection with m segments $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$. In order to show minimality, we prove it by contradiction. Suppose that the minimal number of link failures, which would not allow the construction of k-protection with m segments $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$, is less than ζ. Then, there should exist at least segment $\mathbf{S}$ where the minimal number of link failures in $\mathbf{S}$ is less than $\zeta(\mathbf{S})$. This leads to the contradiction. ■

Now, we present an algorithm to construct the resilience-guaranteed backup segments under multiple simultaneous link failure occurrences. It is assumed that the capacities of the backup links are abundant in order to satisfy the other QoS constraints such as bandwidth.

Algorithm Resilience-Guaranteed_Segment_Restoration
(MPLS/GMPLS Configuration, k-protection)
Begin

Step (1) Decompose the set of nodes in the primary path, associated with $\mathbf{R}$, into a disjointed collection of segments $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m\}$ such that $\mathbf{P} = \bigcup_{i=1}^{m} \mathbf{S}_i$ and each segment has the monotonic property.

Step (2) Pre-reserve a backup segment of each separate segment and determine $k_1, k_2, \ldots, k_m$, such that the path resilience k is equal to $\sum_{i=1}^{m} k_i$.

Step (3) Upon receiving multiple failure notifications, calculate the maximum number of allowable link failures $\zeta(\mathbf{S})$ for each failed segment $\mathbf{S}$, by applying Theorem 3. If for every failed segment $\mathbf{S}$, $\varepsilon(\mathbf{S}) \leq (\zeta(\mathbf{S}) - 1)$ and $k = \sum_{i=1}^{m} k_i$, it is possible to construct a backup path having k-protection with m segments, according to Theorem 4, and go to Step (4). Otherwise, exit ("The construction of the k-protection backup path is not guaranteed.");

Step (4) For each failed segment, re-construct the backup segment by using algorithms 2 and 3 in [8].

End

Both the decomposition process of Step (1) and the evaluation of k and $\zeta(\mathbf{S})$ in Step (3) can be done with O(n). Since the computational complexity of constructing a backup segment can be done with $O(n^2)$ [8], Step (4) can be computed with $O(n^3)$.

4 Simulation Results

Fig. 3(a) shows the test configuration of the GMPLS backbone for simulation purposes, consisting of 9 nodes and the links between them representing logical connections. In Fig. 3(a), the primary path is <N1, N3, N5, N7, N9> and there are two backup paths: 1 and 2. Backup Path 1 is <N1, N2, N6, N9> and Backup Path 2 is <N1, N4, N8, N9>. Here, N1 and N9 are target GMPLS end nodes, i.e. the source and destination nodes, respectively. In this case, the resilience value of the primary path could be 1, if only one backup path is reserved, and 2, if both backup paths are reserved.

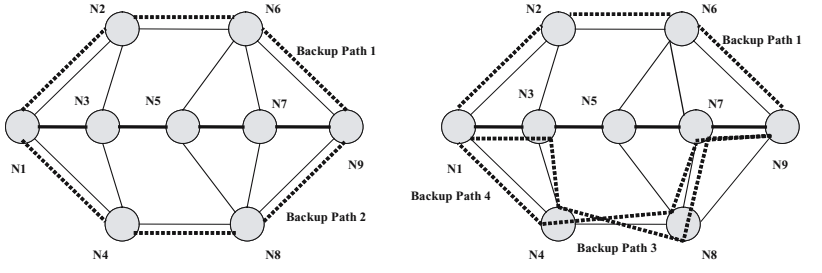


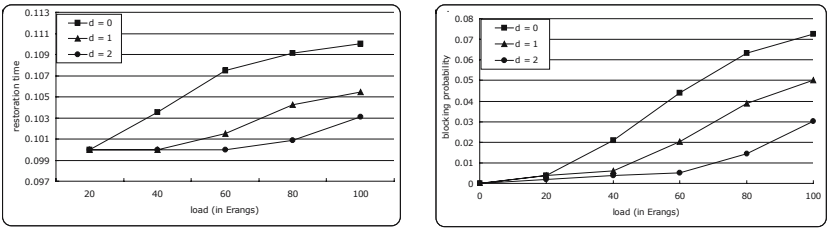
Fig. 3. The test configuration of the GMPLS backbone for simulation purposes

By applying the basic testing conditions in Subsection 3.1, the (ζ - 1) from the path domain {N1, N3, N5, N7, N9} to the non-path domain {N4, N8} for 4-protection is 1, and it is also 1 to the non-path domain {N2, N6}. Thus, the minimum number of link failures which would not allow the construction of a 4-protection backup path is 2, since two independent non-path domains {N2, N6} and {N4, N8} should be taken into account. Fig. 3(b) shows the backup paths, Backup Path 3 and Backup Path 4, for building the primary paths with 2-protection and 3-protection, respectively. We have incorporated a dynamic load sharing policy which is described as follows: *the input load to the primary path, with a resilience value of greater than or equal to one is equally shared among available backup paths when the primary path either fails or is overflowed.*

In the simulation environment shown in Fig. 3, the input data traffic enters the node P1, and is transmitted to node P9. Here, P1 and P9 serve as the source node and destination node, respectively. We simulate about 10,000 calls

for each simulation. When simulating the recovery procedure, it is assumed that the node adjacent to the failure location can detect and localize the failures. The source node usually waits for a very short time interval to check whether there are other failure notification messages, and then it tests the conditions for backup availability, if a backup path is not reserved or damaged. If the testing conditions are satisfied, it can rapidly reconstruct the backup path by using the backup path design algorithm. The source node can then immediately switch the input data traffic to the backup path, thus resulting in high service availability with minimal service disruption.

In Fig. 4, we show the simulation results for measuring performances under different values of resilience and network load. Fig. 4(a) shows the restoration time and Fig. 4(b) shows the blocking probability when the network load varies. The resilience value is denoted as δ . The network load is assumed to have a Poisson probability distribution. Here, the restoration time is defined as the average repair time from all successfully restored paths [3]. For a resilience value of 2, the curve goes up with an increase in the network load from 0.1ms at a load of 40 Erlangs to 0.103ms at a load of 100 Erlangs. For a resilience value of zero, the curve goes up from 0.1035ms at a load of 40 Erlangs to 0.11ms at a load of 100 Erlangs. Fig. 4(a) indicates that the restoration time generally increases as the network load increases. This is due to the fact that the larger resilience value is encompassed with greater load sharing.



(a) Restoration time vs. network load

(b) Blocking probability vs. network load

Fig. 4. The performance of the GMPLS network under multiple failures

In Fig. 4(b), we show the characteristics of the blocking probability for the resilience values 0, 1, and 2. The blocking probability is defined as the ratio of the number of unsuccessful connection requests to the total number of connection requests in a network [3]. For a resilience value of 2, the curve goes up with an increase of the network load from 0.002 at a load of 20 Erlangs to 0.03 at a load of 100 Erlangs. For a resilience value of zero, the curve has value of 0.004 at a load of 20 Erlangs to 0.075 at a load of 100 Erlangs. Fig. 3 indicates that the blocking probability also increases as the network load increases. The blocking probability, however, decreases as the resilience values become larger. This is because with the increase in the resilience value, the signaling data can be delivered more reliably. We compare the performance of the hybrid approach

with that of the IETF rerouting mechanism. Here, it should be noted that the rerouting mechanism corresponds to the case of the hybrid approach with a resilience value of 0. This is because for the rerouting mechanism, a separate alternative path is constructed when link failures occur in the primary path. As shown in Fig. 4, the proposed approach outperforms the rerouting mechanism in terms of restoration time and blocking probability.

5 Conclusion

In this article, we presented a methodology for the fast restoration of resilience-guaranteed segments under multiple link failures, in a general mesh-type MPLS/GMPLS network. We have derived the conditions to test the availability of backup paths which satisfy the resilience constraints for a general mesh-type GMPLS network, with arbitrary configurations. With these existing conditions, an efficient hybrid backup path management strategy has been developed to rapidly find the optimal backup path which satisfies the resilience constraints under multiple link failures in the GMPLS network. Simulation results show that the proposed hybrid mechanism provides a faster service recovery time and better blocking probability than the conventional rerouting mechanism of IETF standards.

References

1. J. P. Lang and J. Drake, "Mesh network resiliency using GMPLS," Proceedings of the IEEE, Vol. 90, No. 9, (2002) 1559 - 1564.
2. A. Banerjee, J. Drake, et al, "Generalized Multiprotocol Label Switching: An overview of Signaling Enhancements and Recovery Techniques," IEEE Comm. Mag., Vol. 39, No. 7, (2001) 144-151.
3. J. Wang, L. Sahasrabuddlhe, and B. Mukherjee, "Path vs. Subpath vs. Link Restoration for Fault Management in IP-over-WDM Network: Performance Comparisons Using GMPLS Control Signaling," IEEE Communications Magazine, (2002).
4. Papadimitriou et al., Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration), draft-ietf-ccamp-gmpls-recovery-analysis-03.txt, (2004).
5. J. P Lang and B. Rajagopalan, "Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification, draft-ietf-ccamp-gmpls-recovery-functional -03.txt, (2004).
6. S. K. Lee and D. Griffith, "Hierarchical Restoration Scheme for Multiple Failures in GMPLS Networks," Proceedings of the 2002 ICPP Workshops 18-21 (2002) 177-182.
7. M. Clouqueur and W. D. Grover, "Availability Analysis of Span-Restorable Mesh Net-works," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 4, (2002) 810-821.
8. Jong T. Park, "Resilience in GMPLS Path Management: model and mechanism," IEEE Comm. Mag., vol. 42, no.7, (2004) 128-135.
9. L. Berger, I. Bryskin et al., "GMPLS based segment recovery," Internet Draft, draft-ietf-ccamp-gmpls-segment- recovery-00.txt, (2004).

Impact of Burst Control Packet Congestion on Burst Loss Rate in Optical Burst Switched Networks

In-Yong Hwang, Seoungyoung Lee, and Hong-Shik Park

School of Engineering, Information and Communications University,
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
{iyhwang, seoungyoung, hspark}@icu.ac.kr

Abstract. In the Optical Burst Switching (OBS) research area, the burst control packet (BCP) queuing delay problem has not been addressed because it is believed to be quite small. However, with a realistic OBS simulator, we have investigated this issue and clarified its impact on performance degradation from the point-of-view of data burst loss rate. We know that the BCP load to the control channel is not negligible. Thus, the burst loss rate due to the queueing delay of BCP on the control channel is very serious compared to the existing well-known burst contention. We propose a Dynamic Offset-Time Update scheme to avoid serious data burst loss due to BCP queueing delay, and a Priority BCP queue to guarantee the minimum offset-time for high class bursts. Our simulation results show that the Dynamic Offset-Time Update scheme can completely avoid data burst loss due to BCP congestion while guaranteeing a certain level of minimum offset-time.

1 Introduction

Optical burst switching (OBS) is attractive for increasing network utilization in wavelength paths because of the potential of fine-granularity optical switching. In the OBS network, packets are assembled into bursts at an ingress router, routed via the OBS network, and then disassembled into packets at the egress router. An original feature of OBS is the physical separation of the optical data transport and the electronic control of the switch, which can facilitate the electronic processing of Burst Control Packet (BCP) at OBS core nodes and can provide end-to-end transparent optical paths for transporting data bursts [1]. There are many research issues [2] in the OBS area, including burst assembly, contention resolution, data channel scheduling, and QoS support, etc. The most basic issue is contention resolution because burst loss is still unavoidable in OBS networks due to limitations of the optical buffer. This is the main reason why the OBS network has not been rapidly deployed. Therefore, reducing loss rate is the most urgent problem that needs to be addressed in the OBS area. Even, so many OBS related researches have focused on contention resolution of the

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 369–378, 2006.

data channel. However, control channels as well as data channels may incur data burst loss in the OBS network under a heavily loaded condition. Usually, packets are delayed in highly loaded situation and some of them are maybe dropped with probability to avoid congestion. However, all delayed BCPs incur drop of all corresponding data bursts.

Compared to the burst contention problem of the data channel, BCP delay has not received much attention because the offered load to the control channel is believed to be quite small. In this paper, we focus on the impact of BCP delay on performance in OBS networks. The rest of the paper is organized as follows. In Section 2, we introduce the queueing delay problem of BCPs in OBS networks. In Section 3, we propose a dynamic offset-time updating scheme and Priority BCP queueing to solve the data burst loss problem due to BCP queueing delay while providing suitable end-to-end delay for packets. In Section 4, we show a performance evaluation of the impact of BCP queueing and the proposed scheme using the OBS-ns simulator. And finally, in Section 5, we draw our conclusions.

2 Queueing Delay Problem of BCP in OBS Networks

In Figure 1(a), we present the normal arrival of the Just-Enough-Time (JET) [1] OBS network. In the JET OBS, an ingress node sends out a burst control packet (BCP), which is followed by a data burst after offset-time $T(i) \geq \sum_{h=1}^{H} \delta_0(h)$ where $\delta_0(h)$ is the processing time at hop $1 \leq h \leq H$. In the core nodes, the offset-time is updated as follows:

$$T(i) = T - \sum_{h=1}^{i-1} \delta_0(h) \tag{1}$$

where the fixed valued processing time without the queueing time is considered for the offset-time update. This delay performance in aggregate First Come First Service (FCFS) based OBS control channel has, surprisingly, not been considered in the OBS research area because most believe the amount of control traffic to be small. However, it is required to learn a lesson from the Internet QoS community in that small utilization cannot guarantee low bound of worst case delay, especially in aggregate FCFS links [7, 8, and 9]. In other words, even if BCP traffic is ideally shaped and the input load is substantially smaller than the control channel bandwidth, instantaneous queues can be quite large. Such instantaneous queue build-up introduces a certain amount of jitter, which increases the burstiness of ideally shaped traffic. This burstiness can in turn cause further delay as traffic, which is no longer ideally shaped and merges downstream, leads to yet more delay, jitter and further accumulation of burstiness. This effect may be especially severe if the BCP traffic sharing aggregate FCFS link has a substantial burst-to-rate ratio, which is considered a real OBS control channel environment [11].

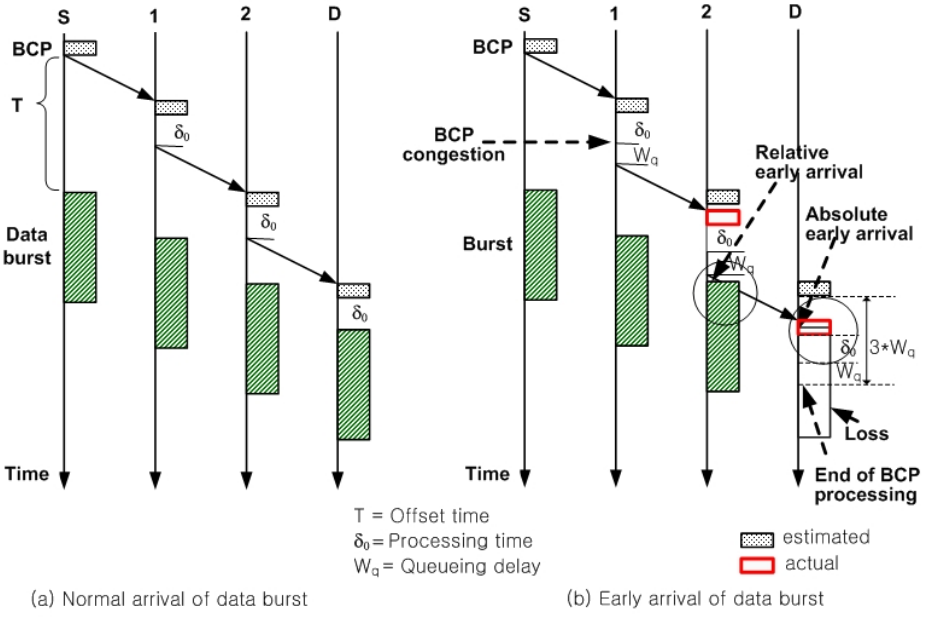


Fig. 1. Normal arrival and early arrival of data burst in OBS network

2.1 Early Arrival Problem in OBS Network

The defined (estimated) offset-time changes easily on the way to the destination due to excessive BCP queueing delay. When congestion occurs in the control channel, the elapsed time in the switching module of each core node increases. If a data burst enters the optical switching matrix before its BCP has been processed, the data burst is simply dropped because data bursts cannot be buffered at the optical level. This is referred to as the so-called "early arrival" [3] phenomenon. Remarkably, the difference between the defined (estimated) offset-time and the experienced (actual) offset-time results in inevitable data burst loss. We split the "early arrival problem" into "relative early arrival problem" and "absolute early arrival problem" to account for details.

1) Relative early arrival problem. The BCP arrives at the node later than the estimated arrival time due to excessive queueing delay; however, it arrives earlier than the data burst. Because the existing offset-time update scheme cannot avoid data burst loss, a new offset-time management scheme is necessary to avoid the relative early arrival problem. Node 2 in Figure 1 (b) shows an example of the relative early arrival case. To avoid the relative early arrival problem, the offset-time update process should consider the difference between defined and actual BCP delay.

2) Absolute early arrival problem. The BCP arrives at the node later than the data burst and is eventually dropped. The BCP close to a destination

experiencing many hops has potential high absolute early arrival probability, which may degrade performance from the overall resource utilization viewpoint. Node D in Figure 1 (b) shows an example of the absolute early arrival case. Once the offset-time is assigned to the BCP at the ingress node and sent to the destination, it is impossible to avoid the absolute early arrival problem under a heavily BCP loaded condition. Thus, the adaptive offset-time calculating scheme is necessary.

2.2 Related Works

To reduce the data burst loss rate in consideration of excessive BCP delay, there have been several researches. In the guard band scheme [1], the wavelength is reserved with margin to accommodate the early or late arrival of the data burst at the cost of link utilization reduction. If the guard band requires twice the burst duration, the overall utilization of data channels falls to 1/2. Thus, it is not appropriate for the case in which the time difference is huge compared to the data burst duration. The merit-based scheduling algorithm [4], which ranks an arriving burst against those which have already been scheduled for transmission, preempts the one which will cause the least impact in terms of lost resources in favor of the new arrival. Even though it uses offset-time information, it only concentrates on data channel scheduling with no consideration of control channel scheduling. Thus, it cannot solve the early arrival problem.

In the BSCOT algorithm [5], [6], a BCP with short residual offset-time is served prior to BCPs with long residual offset-time so that it can reduce the data burst loss rate at each node and the total loss rate over the entire network. Both algorithms assume that the offset-time can increase with the queueing delay in the control channel of the OBS network, which then enhances overall resource utilization by intentionally dropping the burst with high potential early arrival rate. Such data channel scheduling algorithms cannot completely avoid early arrival rate. Thus, we propose a new algorithm to provide an early arrival free OBS model.

3 Dynamic Offset-Time Update Scheme

3.1 Core OBS Node Architecture

In this paper, we adopt a queue for BCPs and propose a new offset-time updating algorithm under BCP queue architecture. Once a BCP arrives at an intermediate node, it enters the BCP queue. It then searches available data channels using the LAUC algorithm [3]. If the BCP is successful in finding an available data channel, it is scheduled to the suitable control channel and sent to the next node. If the BCP cannot find any available data channel, it should wait until finding an available data channel. After finishing data channel scheduling, the BCP is sent to the next node using the control channel scheduling algorithm for multiple control channels. To avoid huge delays in the case of the heavily BCP loaded

condition, the BCP scheduler with multiple queues operating in non-preemptive Priority Queueing mode is adopted. With the OBS core node architecture shown in Figure 2, a highly prioritized BCP's queueing delay can be kept at a certain level.

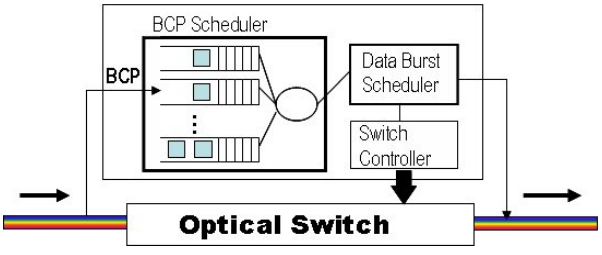


Fig. 2. Core OBS node architecture

3.2 Dynamic Offset-Time Update Scheme with BCP Scheduler

We can consider two methods, the static and the proposed dynamic scheme, for the offset time updating scheme.

1) Static offset time update. The offset time value is fixed which is introduced in equation (1).

2) Dynamic offset time update. The offset time is updated in relation to the offered load.

$$T(i) = T - \sum_{h=1}^{(i-1)} [\Delta + t_f(h) + W_q(h)] \tag{2}$$

where t_f is the BCP forwarding time and $W_q(h)$ is the BCP queueing time at j_{th} hop, respectively. In equation (1), the t_f is neglected, however, it is not a negligible value when compared to the processing time. When the BCP arrives at the intermediate node, the offset-time time is dynamically updated to compensate for the excessive BCP queueing delay. We can avoid the relative early arrival problem using the offset-time update scheme.

To resolve the absolute early arrival problem by assigning the suitable offset-time to the BCP is an important issue. Because the BCP queueing time $W_q(h)$ varies according to the offered load of the control channel, the ingress node continuously updates the offset-time value by the backward BCP from the egress node. When the remaining offset-time is insufficient, $T(j) < \Delta * (H - j)$, at j_{th} hop, the node sends backward BCP to the ingress node to inform the insufficient offset-time. Then, the ingress node updates the offset-time T as $T' = (T/j) * H$, and use it for sending BCP and data burst. With the dynamic offset-time updating scheme and suitable offset-time management we can achieve an early arrival free OBS network.

4 Performance Evaluation

For the simulation, we utilized an OBS-ns simulator [11] implemented under the well-known NS-2 environment. In many OBS simulator researches, only the control channel is implemented without data channels where BCPs are directly generated without the data burst assembly process. This is undesirable when one wishes to accurately investigate volume of offered load to the control channel and its impact on delay and loss rate. In this simulation, we provide a more realistic environment where generated packets are once assembled into a burst with two parameters, the buffer threshold and the timer. As well, data channel scheduling and control channel scheduling are performed in the intermediate nodes. With this realistic OBS simulation environment, we can investigate the data burst blocking probability due to the congestion of the control channel as well as the contention of data channels. Our simulation topology is shown in Figure 3.

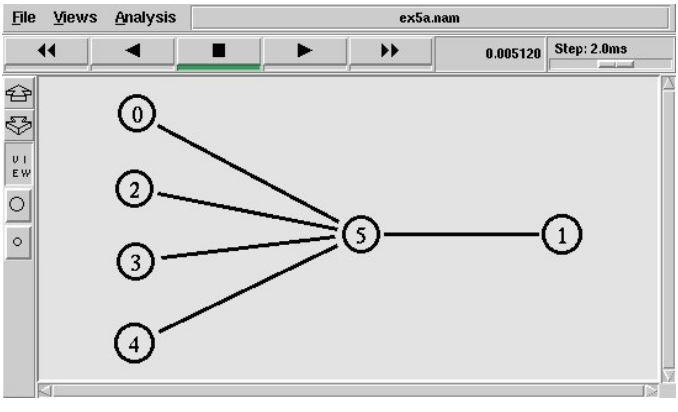


Fig. 3. Simulation Topology

The topology consists of four ingress nodes, one core node, and one egress node. In node 5, the intermediate node, the BCP of the control channel should be congested with heavy offered BCPs. As well, the burst should be dropped due to contention from multiple traffic sources. We can observe burst loss due to BCP congestion as well as burst contention.

We generate packets as Poisson traffic. In the ingress nodes, Poisson traffic is generated with a mean size 0.5 $Kbyte$ and then burstified with a fixed size threshold of 4 $Kbyte$ and a burst time out value of 0.001 sec. The BCP processing time per each node is 2 us and the propagation delay on each link is 1 ms. Each link consists of 16 or 32 data channels sharing 1 control channel, where all links have 1 $Gbps$ bandwidth. The BCP's size is 64 $byte$. Thus, the BCP forwarding time $t_f(h)$ is calculated as 0.5 us. The BCP queue consists of 4 multiple queues. In the simulations, "offered load to the data channels" is calculated as the ratio of entire input load to output link size at the intermediate node.

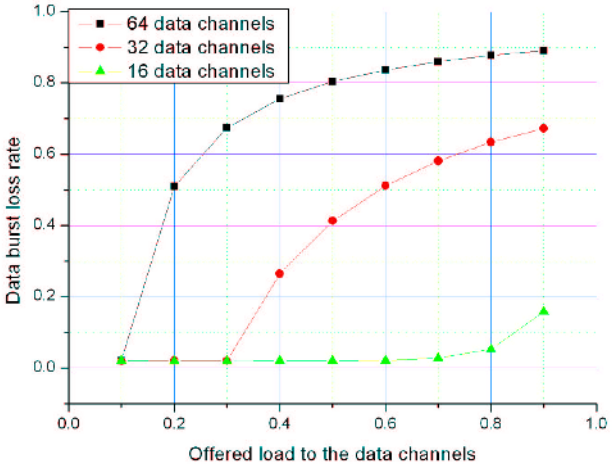


Fig. 4. Data burst loss rate of the static (existing) offset-time under BCP queueing

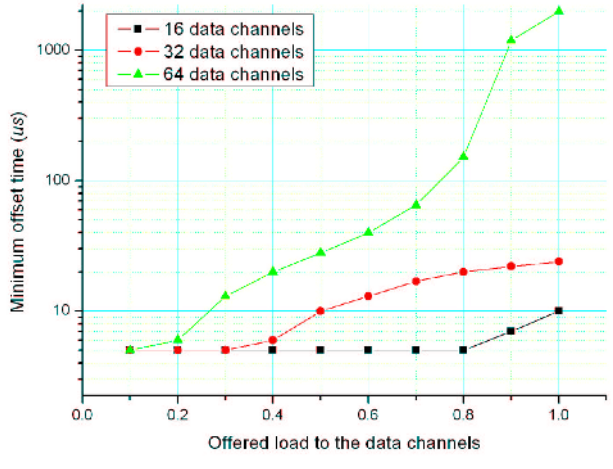


Fig. 5. Minimum offset-time to avoid early arrival of the dynamic offset-time scheme (single class)

In Figure 4, the data burst loss rate is presented for 16 and 32 data channels as increasing the offered load, where the static (existing) offset-time scheme in JET is used.

As the offered load to the data channels increases, the data burst loss rate is drastically increased accordingly. At most, 90 % of data bursts are lost in the case of 64 data channels. The data burst loss rate of 32 data channels is higher than that of 16 data channels because twice as much traffic is offered to the control channel. In the case of 32 data channels, more than 0.3 of the offered

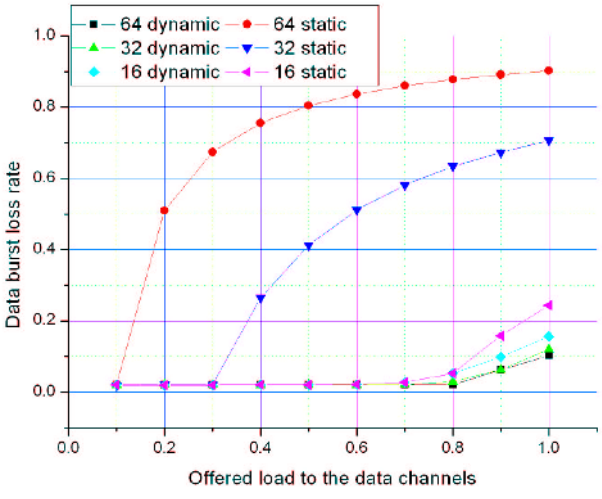


Fig. 6. Data burst loss rate with static offset-time and dynamic offset-time

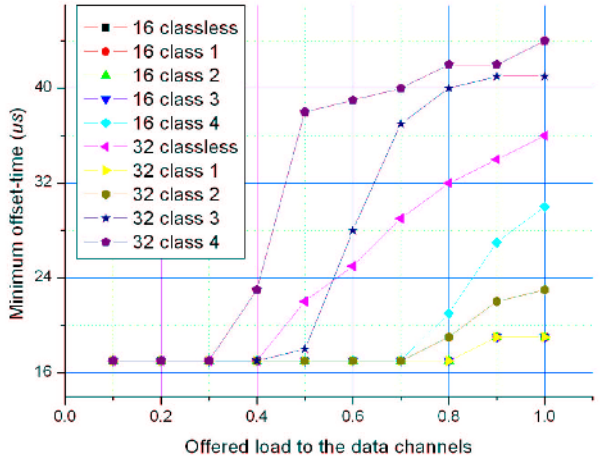


Fig. 7. Minimum offset-time to avoid early arrival of the dynamic offset-time scheme (multiple class)

load is simply dropped due to early arrival. As well, more than a 30 % link is not utilized. It is clear that the early arrival problem makes unexpected serious burst loss in the case of the static offset time scheme.

Figure 5 explains the reason for the above unexpected serious data burst loss, where the minimum offset-time to avoid early arrival of the dynamic offset-time scheme is presented. The offset-time, the end-to-end delay of the BCPs excluding propagation delay, also consists of the processing time and the queueing time of BCPs.

In the case of the static offset-time updating scheme, the offset-time has a fixed value because the BCP queueing time is not reflected on the offset-time. However, in the case of the dynamic offset-time updating scheme, the offset-time increases as the offered load increases. Because it dynamically chases the real offset-time, no BCPs are dropped due to early arrival. If we use 16 data channels, more than a 10 % load is dropped due to early arrival. As well, more than a 10 % link is not utilized. The minimum offset-time becomes quite large in the heavy load condition, thus the simple fixed size offset-time cannot avoid the early arrival problem. Thus, we know that the offset-time is a key factor for determining the data burst loss rate. The burst loss rate of the dynamic and static offset-time scheme with 16 and 32 channels given Poisson traffic is presented in Figure 6. The burst loss rate of the dynamic offset-time scheme is very low compared to the static offset-time scheme. We know that resolving the early arrival problem is prior to all other issues for successfully deploying the conventional OBS network.

From Figure 5, we know that the extra offset-time is provided to guarantee an early arrival free situation even in the heavily loaded condition. To reduce this extra offset-time, priority queueing with multiple queues is applied. Its result is shown in Figure 7 where four classes are used for 16 and 32 data channels.

We know that the high priority shows excellent delay performance compared to the low priority. The delay of class 1 for 16 and 32 data channels and class 2 for the 16 data channel is kept at a fixed value until the offered load is 0.8, at the cost of a delay increase in the low classes. Providing low delay with controllability for the high class is very useful for time sensitive real time applications which require low delay and jitter.

5 Conclusion

We investigated the BCP queueing delay on the data burst loss rate in OBS networks which had not been previously researched. Through the simulations, we found that the volume of BCP traffic is not that small and its queueing delay in JET OBS degrades performance drastically due to the serious early arrival rate. The loss rate from early arrival is much more severe than that from the well-known burst contention problem. Knowing that early arrival is a key factor determining the data loss rate of the OBS network, we proposed a dynamic offset-time updating scheme to get an early arrival free OBS network and priority BCP queueing to keep the offset-time delay at a certain level. From the simulation results, we can configure the OBS network as early arrival free while providing very low delay for high class bursts. The result of our proposed scheme can be very useful deployed in an OBS network in a real networking field.

Acknowledgement

This work was supported in part by the Institute of Information Technology Assessment (IITA) through the Ministry of Information and Communication

(MIC) and the Korea Science and Engineering Foundation (KOSEF) through the Ministry of Science and Technology (MOST), Korea.

References

1. C. Qiao, M. Yoo: Optical Burst switching (OBS) - a new paradigm for an optical Internet, Journal of High Speed Networks, Vol. 8, no 1, (1999) 68-84
2. Y. Chen, et al.: Optical burst switching: a new area in optical networking research, IEEE Network, May/June (2004)16-23
3. Y. Xing, M. Vanderhoute, C.C. Cankaya: Control architecture in optical burst-switched WDM networks, IEEE Journal on Selected Areas in Communications, Vol.18, No.10, October (2000) 1838-1851
4. J. White, R. Tucker, K. Long: Merit-base Scheduling Algorithm for Optical Burst Switching, COIN 2002, July, (2002)
5. J. Kim, H. Yun, J. Choi, M. Kang: A Novel Buffer Scheduling Algorithm for Burst Control Packet in Optical Burst Switching WDM Networks, APOC, Oct., (2002)
6. J.Kim, J.Choi, M.Kang: Offset-Time Based Scheduling Algorhthm for Burst Control Packet in Optical Burst Switched Networks, Lecture Notes in Computer Science, Vol. 3098. (2003)
7. Y. Jiang et al.: Delay Bounds for a Network of Guaranteed Rate Servers with FIFO Aggregation, Computer Networks, Elsevier Science, Vol. 40, No. 6, (2002) 683-694
8. Z. Zhang, Z. Duan, Y. T. Hou: Fundamental trade-offs in aggregate packet scheduling, Proceedings of ICNP 2001, (2001)
9. A. Charny and J. Y. Le Boudec: Delay bounds in a network with aggregate scheduling, Proceedings of QOFIS, October 2000, (2000)
10. D. Morato, J. Aracil, L.A. Diez, M. Izal, and E. Magana: On linear prediction of internet traffic for packet and burst switching networks, Proceedings of ICCCN, (2001) 138-143
11. In-Yong Hwang et al.: OIRC OBS-ns simulator supported by OIRC and Samsung Advanced Institute of Technology (SAIT), (2006)

EIMD: A New Congestion Control for Fast Long-Distance Networks

Eunho Yang[1], Seong-il Ham[1], Seongho Cho[1], Chong-kwon Kim[1], and Pillwoo Lee[2]

[1] School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Republic of Korea
{ehyang, siham, shcho, ckim}@popeye.snu.ac.kr
[2] Korea Institute of Science and Technology Information, Supercomputing Center, Taejon, Republic of Korea
pwlee@kisti.re.kr

Abstract. In fast long-distance networks, TCP fails to fully utilize the bandwidth due to its congestion control mechanism. A plethora of congestion control schemes that may enhance the performance of the transport protocol in fast long-distance networks have been proposed. The proposed schemes aim to satisfy three requirements of congestion control schemes: bandwidth scalability, TCP friendliness, and RTT fairness. However, due to the trade-off among these requirements, it is difficult to satisfy all the requirements simultaneously. In this paper, we propose a new window-based congestion control scheme called EIMD (Exponential Increase/Multiplicative Decrease) that increases congestion window size exponentially to quickly grap the unutilized bandwidth. We evaluate the performance of EIMD via computer simulations. The simulation results show that EIMD satisfies the three requirements. In addition, EIMD converges fast to the fair-share points.

1 Introduction

Ever since the introduction of the ARPANet in 1969, the internet has experienced evolutionary advancements in network speed as well as in the number of users. Several scientific applications such as bioinformatics, telemedicine, and real-time environment monitoring require to access large scale databases, from hundreds of tera-bytes to petabytes, widely distributed across scientific communities. Several high-speed and long-distance networks, ESNet, Abilene, and Grid Networks, to name a few, have been introduced to meet the needs of these applications.

The internet uses TCP for reliable data exchanges. TCP uses a congestion control scheme that enables effective sharing of network resources by hundreds and thousands of users at the same time. Let us briefly examine the basic mechanisms of the TCP congestion control mechanism. Each sender of a TCP connection has congestion window, *cwnd*, which limits the number of outstanding packets that can be sent without receiving acknowledgement (ACK). When a sender receives an ACK successively, then it increase *cwnd* additively as follows

$$cwnd \leftarrow cwnd + 1/cwnd \tag{1}$$

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 379–388, 2006.

On the other hand, if an ACK does not arrive on time, the TCP sender decreases *cwnd* multiplicatively as follows

$$cwnd \leftarrow cwnd/2 \tag{2}$$

Several previous works pointed out that TCP cannot effectively utilize the bandwidth of fast long-distance networks due to its congestion control mechanism [1]. Suppose a link of 10Gbps bandwidth and of 100ms round-trip time (RTT). If the packet size is 1500bytes, *cwnd* should be 83,333 packets to fully utilize the bandwidth. Because a single packet loss would drop *cwnd* to one half of the previous size, it requires at least 42,666RTTs (that is 71 minutes!) to reclaim the maximum *cwnd*.

Many researchers have attacked the underutilization problem of TCP in fast long-distance networks and proposed a plethora of solutions. These solutions can be classified into several categories; tuning of existing TCP [9], window-based protocols [1,2,3], delay-based protocols [4], and other protocols that go beyond modification only to TCP [5]. Most protocols addressed one or more of three properties, bandwidth scalability, TCP friendliness, and RTT fairness, required for congestion control protocol in fast long-distance networks. Bandwidth scalability means that a congestion control protocol should effectively utilize bandwidth of high-speed networks. On the other hand, TCP friendliness means that a TCP flow competing with a high-speed protocol should get as much bandwidth as a flow of high-speed protocol when packet loss event rates are high. RTT fairness means that competing flows with different RTTs achieve fair allocation of bottleneck bandwidth. In addition to above three requirements, a congestion control algorithm for fast long-distance networks should be easy to deploy; it should not require modification of intermediate routers and receivers [6]. Several researchers favor window-based algorithms because window-based algorithms are self-clocking and fail-safe.

Designing a window-based congestion control protocol, however, is still challenging because it is very difficult to satisfy bandwidth scalability, TCP friendliness, and RTT fairness requirements at the same time. We may achieve bandwidth scalability and TCP friendliness by adaptively applying different *cwnd* update rules depending on packet drop probabilities. However, mechanisms to improve scalability and TCP friendliness tend to worsen RTT fairness. The main cause for RTT unfairness is synchronous packet losses. When the bandwidth of the network is large, loss event rates for flows with different RTTs experience unfairness while all the flows apt to have similar loss event rates when the bandwidth is not so huge [3].

This paper presents a new window-based congestion control protocol for high-speed networks called *Exponential Increase/Multiplicative Decrease (EIMD)*. The most distinctive feature of EIMD is its *cwnd* increment mechanism. EIMD increases *cwnd* exponentially proportional to the time elapsed since the last packet loss occurs. Exponential *cwnd* increment guarantees the bandwidth scalability when packet loss event rate is low. In addition to the elapsed time between two consecutive packet loss events, EIMD considers two other factors in determining the *cwnd* increment. The two factors are RTT and the *cwnd* size just

before a packet loss occurs. EIMD enhances RTT fairness and TCP friendliness by using different response functions depending on RTT. In addition to the three requirements explained before, fair-share convergence time is another requirement of high-speed congestion control. EIMD aims to expedite the convergence to the fair-share by increasing *cwnd* inversely proportional to the *cwnd* size just before the last packet loss occurs.

The remainder of this paper is organized as follows. As background, we describe the properties of high-speed TCP in Section 2. In Section 3, we first propose a new response function adjusted by RTT. We then derive the congestion control rules from the proposed response function. Our simulation results are described in Section 4. Conclusion appears in Section 5.

2 Properties of Congestion Control Algorithm

The characteristics of congestion control algorithm can be represented by a response function. A response function relates the average *cwnd* size(or sending rate) as a function of packet loss event rate p. In general, the response functions of most existing congestion control can be approximated as

$$w = A/p^s, \tag{3}$$

where w is the average *cwnd* size and A and s are protocol-dependent constants.

Let us briefly review the characteristics of response functions in terms of three requirements: bandwidth scalability, TCP friendliness, and RTT fairness. At low packet loss event rates, a phenomenon typically observable in high-speed networks, *cwnd* should be large to consume abundant bandwidth effectively. On the other hand, when the network is congested and the packet loss event rate is high, *cwnd* should be small to allocate adequate resources to co-existing TCP flows. Therefore, the response function must decrease steeply as the packet loss event rate increases to satisfy both bandwidth scalability and the TCP friendliness. Note the slope of a response function is determined by a parameter s in Equation (3).

The response function determines RTT fairness also. Xu, Harfoush and Rhee [3] showed that the throughput ratio of two flows with RTT_1 and RTT_2 is $(RTT_1/RTT_2)^{1/(1-s)}$ where -s is the slope of response function in a log-log scale. The s of TCP, HSTCP, and Scalable TCP are 0.5, 0.82 and 1, respectively. Therefore, throughput ratios of TCP, HSTCP, and Scalable TCP are 4, 43, and ∞, respectively, if $RTT_1 = 2 \cdot RTT_2$. The challenge is how to support bandwidth scalability, TCP friendliness, and RTT fairness at the same time. BIC [3] tried to satisfy the three requirements by varying the parameter s from 0.5 to 1 as the packet loss event rate increases.

The trajectory model developed by Jin and et al. [7] can be used to determine the fair-share convergence time. Using this model, we can prove that the TCP's AIMD mechanism guarantees the convergence to the fair-share point. However, many high-speed congestion control algorithms either fail to reach to the fair-share point or have a long convergence time. This is because they increase *cwnd*

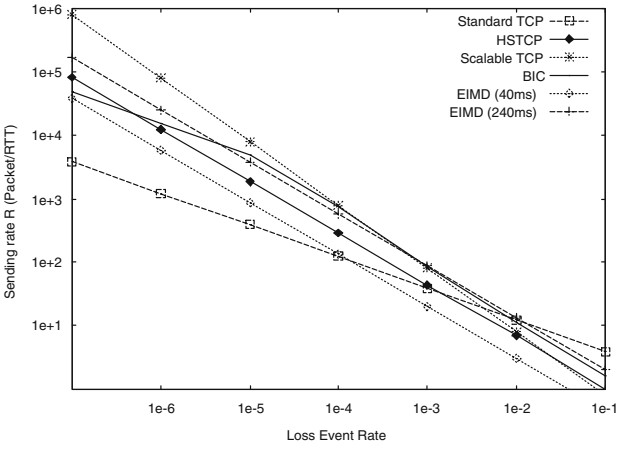


Fig. 1. Response function of EIMD

in proportion to the current *cwnd* size. For faster convergence to fair-share, Jin and et al. [7] suggested to increase *cwnd* in proportion to $1/w_{max}$, where w_{max} is the *cwnd* size just before the last packet loss.

3 EIMD Congestion Control

In this section, we first present the response function of EIMD, and then describe its congestion control rules. We assume that TCP flows have smaller RTTs (below 50ms) than high-speed flows. We also assume packet loss occur synchronously because this phenomenon is typically observed in fast long-distance networks [3].

3.1 Response Function

As discussed in Section 2, response function of Equation (3) cannot support bandwidth scalability, TCP friendliness, and RTT fairness simultaneously. We augment the RTT factor to Equation (3) and derive a new response function as,

$$w = \frac{A}{p^s} RTT^r, \tag{4}$$

where r is a parameter that determines the degree of RTT fairness. Flows with different RTTs now have different response functions. This effect compensates for irrationally high loss event rates by a synchronous loss. Figure 1 describes response functions of EIMD flows with RTTs of 40ms and 240ms. The parameters s, r and A can be adjusted according to network environment or application's requirements. When r and s are set to 0.82 and A= 1, EIMD flow with 100ms

RTT has the same response function as HSTCP. However, we cannot directly compare the properties of EIMD and other protocols (including HSTCP) solely based on the response function because the response function of EIMD considers RTT. Moreover, to evaluate TCP friendliness, we should explicitly consider the RTT factor since TCP flows have the relatively shorter RTTs than high-speed flows. We will examine the throughputs of EIMD and other protocols in terms of loss event rate.

First, let us analyze the relationship between r and RTT fairness. Let w_i and RTT_i denote the average *cwnd* size and the RTT of flow i, respectively. Let τ be the time (in seconds) between two consecutive loss events. Note that τ is the same for all flows under the synchronous packet loss assumption. Suppose that loss events of flow i are uniformly distributed with rate p_i. The average number of packets sent between two consecutive loss events is $1/p_i$. Since the total number of RTTs is τ/RTT_i, we obtain the following Equation

$$w_i = \frac{1/p_i}{\tau/RTT_i} = RTT_i/(\tau \cdot p_i). \tag{5}$$

From Equation (4) and (5), we derive $w_i = \frac{RTT_i^{1-r/s}}{\tau \cdot (A/w_i)^{1/s}} \rightarrow w_i = \left(\frac{\tau^s A}{RTT_i^{s-r}}\right)^{1/(1-s)}$, and $\frac{w_1}{w_2} = \left(\frac{RTT_2}{RTT_1}\right)^{\frac{s-r}{1-s}}$. The RTT unfairness of the two flows, which is defined as the ratio of average throughputs, is determined as follows

$$\frac{(w_1/RTT_1)/(1-p_1)}{(w_2/RTT_2)/(1-p_2)} \approx \frac{w_1/RTT_1}{w_2/RTT_2} = \left(\frac{RTT_2}{RTT_1}\right)^{\frac{1-r}{1-s}}, \tag{6}$$

where $(1-p_i)$ is approximated to be 1. We can now compute loss event rate of flow from Equation (4) and (5) as

$$p_i = \left(\frac{RTT^{1-r}}{A \cdot \tau}\right)^{1/(1-s)}, \; p_1/p_2 = (RTT_1/RTT_2)^{(1-r)/(1-s)}. \tag{7}$$

If r is set to be the same as s, the loss event rate is proportional to RTT and the *cwnd* size is independent of RTT. As a result, a throughput is inversely proportional to RTT. If we ignore the RTT factor in the response function as in Equation (3) (that is to set r to 0), the *cwnd* ratio grows exponentially as s approaches to 1. Figure 2 shows the *cwnd* sizes when two flows with 40 and 120ms RTT coexist. *cwnd* sizes of EIMD and HSTCP flow with 120ms RTT are separately described when the flow with 40ms RTT experiences a loss event rate of 10^{-6}. The loss event rate of each high-speed flow with 120ms RTT is determined by the Equation (7). While EIMD flows have the same *cwnd* size regardless of RTT, HSTCP flows experience severe unfairness in loss event rate, as a result, they have unfair *cwnd* sizes.

Now, let us examine the TCP friendliness. Figure 3 shows the throughputs (not sending rates!) of two HSTCP flows with 40 and 240ms, respectively. TCP friendliness is determined by the points where the throughput lines of HSTCP

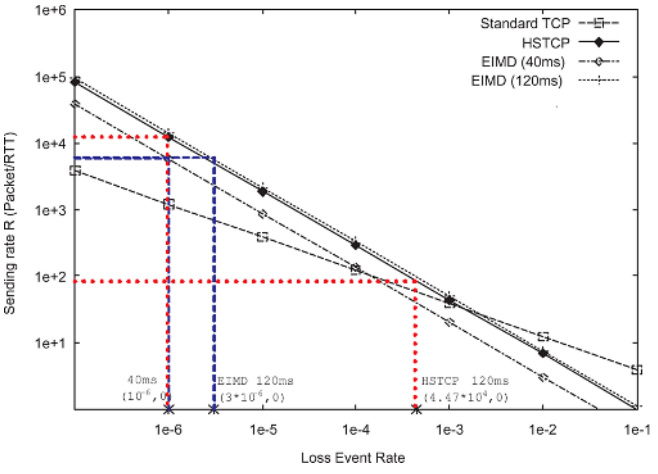


Fig. 2. Loss event rates of flows with different RTTs

intersect with that of TCP. Note that the high-speed flows with relatively longer RTTs hardly destroy the TCP friendliness of a protocol. Therefore, making flows with smaller RTTs to be less aggressive improves TCP friendliness as well as RTT fairness. That is, we need to reduce the gap between throughput lines of flows with different RTTs to improve the RTT fairness and TCP friendliness at the same time. Figure 4 shows the throughput of EIMD and HSTCP. EIMD has a smaller gap between flows with different RTTs than HSTCP. This indicates that EIMD achieves higher RTT fairness and TCP friendliness than HSTCP.

Lastly, let us consider the bandwidth scalability. We designed EIMD such that EIMD flows whose RTT is larger than 100ms get more throughput than HSTCP flows, and EIMD flows whose RTT is smaller than 100ms get less throughput than HSTCP flows. As shown in Figure 4, flows whose RTT is smaller than 100ms achieve more than 10Gbps of throughput if the loss event rate is less than or equal to 10^{-7} according to Equation (6).

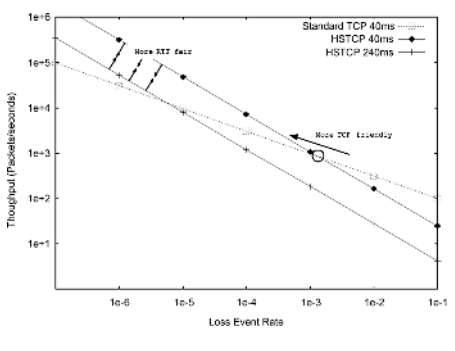


Fig. 3. Throughputs of HSTCP flows

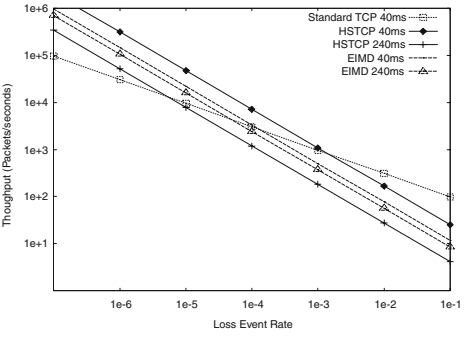


Fig. 4. Throughputs of EIMD flows

3.2 Increase/Decrease Rules of EIMD

We derive the increase/decrease rules of EIMD from the EIMD's response function described in Section 3.1. As stated in Section 1, EIMD increases *cwnd* exponentially proportional to the time after the last packet loss occurs. So *cwnd* can be represented as

$$w(T) = w(0) + cT^u \tag{8}$$

where c and u are constants and $w(T)$ is the continuous approximation of the *cwnd* size at time T (in RTTs) elapsed since the last packet loss.[1] We define the increase rule of EIMD as $w_{T+1} \leftarrow w_T + \alpha(w_T - w_0)^{1-1/u}$ where w_T is the *cwnd* size at time T, $\alpha = u \cdot c^{1/u}$ and $w_0 = w(0)$. From the increase rule, we have $dw(T)/dT = \alpha(w(T) - w_0)^{1-1/u}$. Transposing and integrating both sides, we have the same approximation as Equation (8). After a packet loss event, EIMD decreases *cwnd* to $w_{max}(1-\beta)$ like conventional TCP. In summary, EIMD updates *cwnd* as

$$w_{T+1} \leftarrow w_T + \alpha(w_T - w_0)^{1-1/u}, \text{if ACK arrives on time,}$$
$$w_{T+\delta} \leftarrow w_T - \beta w_T, \text{if timeout.}$$

In the above rule, δ denotes the delay to detect a packet loss and β is fixed to 0.125.

Now, we calculate the c (in fact, α in increase rule) and u as a function of w_{max} and RTT such that EIMD follows the response function presented in the previous section. Under the steady-state assumption, the amounts of increment and decrement should be the same, so we have $c \cdot T^u = \beta \cdot w_{max}$. So, the total number of packets transmitted in duration T, $L(T)$, can be calculated by integrating Equation (8) as $L(T) = \int_o^T ((w_{max} - \beta w_{max}) + cT^u)dT = (w_{max} - \beta w_{max})T + \frac{c}{u+1}T^{u+1}$. Since this has to be same as $1/p$, we can have

$$L(T) = \frac{1}{p} = (w_{max} - \beta w_{max})T + \frac{c}{u+1}T^{u+1} \tag{9}$$

Applying an equation $T = \tau/RTT$ to Equation (5), we derive the average sending rate, w, as $w = 1/(p \cdot T)$. This w has to be consistent with the proposed response function in Euqation (4). So, we obtain the following equation

$$\frac{1}{p \cdot T} = \frac{A}{p^s} RTT^r. \tag{10}$$

Using Equation (9), (10), we can have the following results

$$c = \left(\frac{(A \cdot RTT^r)^{1/(1-s)}}{1 - (u/(u+1))\beta} \right)^{u(1-s)/s} \cdot \frac{\beta}{\Gamma(u+1)} w_{max}^{1-u(1-s)/s} \tag{11}$$

$$\alpha = u \left(\frac{(A \cdot RTT^r)^{1/(1-s)}}{1 - (u/(u+1))\beta} \right)^{(1-s)/s} \cdot \frac{\beta}{\Gamma(u+1)} w_{max}^{\frac{1}{u} - \frac{(1-s)}{s}} \tag{12}$$

[1] SIMD selects a special case of the equation (7) to be TCP friendly; $w(T) = w_0 + cT^2$. And increase rule of SIMD is $w_T \leftarrow w_T + \alpha\sqrt{(w_T - w_0)}$ Therefore, EIMD can be regarded as a high-speed version of SIMD.

In Equation (11), we can see that the *cwnd* size increases in proportion to $w_{max}^{1-u(1-s)/s}$. We derive $u = 2s/(1-s)$ from an equation $1 - u(1-s)/s = -1$. It makes $c \propto 1/w_{max}$, which means that *cwnd* increases in proportion to $1/w_{max}$. This is the fastest way that guarantees the trajectory line to converge to the optimal point, as mentioned in Section 2. If r, s and A are set as shown in Figure 1, we can increase *cwnd* by the following increase rule for each ACK

$$w_{new} \leftarrow w_{old} + \alpha(w_{old} - w_0)^{1-1/u}/w_{old},\ \alpha = 1.8RTT/w_{max}^{0.11},\ u = 9.11. \quad (13)$$

4 Simulation Results

We analyze the performance of EIMD via computer simulations, both in conventional network environments and fast long-distance network environments. We also compare the performance of EIMD with those of several window-based congestion control schemes. We assume a dumbbell network topology that is roughly comparable to that of real high-speed networks, because bottleneck is likely to occur at a point where high-speed flows meet. To reduce the phase effect, we assume that each flow starts and finishes independently. For background traffic, we generate short TCP flows whose *cwnd* size is restricted to be less than 64 in each direction. We compare EIMD, HSTCP, Scalable TCP, and BIC in terms of the following properties: RTT fairness, TCP friendliness, bandwidth utilization, and the fair-share convergence time. In all simulations, we use fixed values for EIMD parameters as the $r = s = 0.82$, $A = 1$, and $\beta = 0.125$. For other protocols, default values are used.

RTT Fairness. We set a simulation where two high-speed flows with different RTTs share the bottleneck link. RTT of one flow is fixed to 40ms and RTT of other flow varies from 40ms, to 120ms, and to 240ms. Table 1 and Table 2 show inverse throughput ratios in terms of the RTT ratio when the capacity of the bottleneck link is 200Mbps and 2.5Gbps, respectively. In both cases, EIMD shows much better RTT fairness than HSTCP and BIC.

Table 1. Inverse throughput ratios under 200Mbps link

RTT ratio	1	3	6
EIMD	1.1	2.27	2.93
BIC	1.09	18.54	43.49
HSTCP	1.1	10.45	26.51
Scalable TCP	1.08	37.78	76.15

Table 2. Inverse throughput ratios under 2.5Gbps link

RTT ratio	1	3	6
EIMD	1.07	3.2	4.91
BIC	1.03	10.63	35.08
HSTCP	1.08	28.03	108.08
Scalable TCP	1.03	137.87	330.01

TCP Friendliness. In this simulation, we evaluate the throughput of coexisting long-lived TCP flows. Two high-speed flows, f_1 and f_2 have RTT of 40ms and 120ms, respectively. Figure 5 shows the percentage of bandwidth that each flow

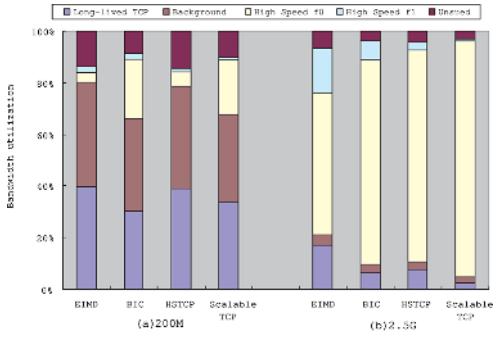


Fig. 5. Percentage of bandwidth share under (a)200Mbps and (b)2.5Gbps

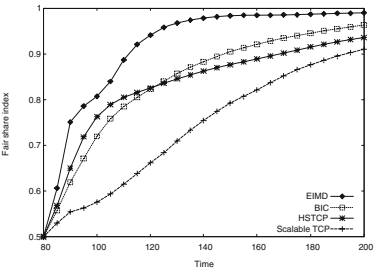


Fig. 6. Fair-share convergence of high-speed protocols

shares. The sum of throughputs achieved by long-lived TCP flows and background traffic indicates the degree of TCP friendliness. In both cases, EIMD is the most TCP friendly. While BIC utilizes bandwidth more efficiently than EIMD, its efficiency is achieved by taking bandwidth away from regular TCP flows (one BIC flow consumes almost same as 10 TCP flows under 200Mbps) regardless of *cwnd* size.

Bandwidth Scalability. We measure the bandwidth utilizations to test whether the proposed protocol is suitable for high-speed environments. The bandwidth of the bottleneck link is varied from 200Mbps to 5Gbps. Table 3 shows that the ratio of unused bandwidth decreases as the total bandwidth increases. EIMD fails to achieve 100% utilization because EIMD flows act like TCP flows when the packet loss rate is high. However, considering unused bandwidth includes the ACK for the backward traffic, these results suggest that EIMD utilizes the available bandwidth very effectively in high-speed networks. Both HSTCP and BIC utilize more than 95% of bandwidth as the capacity of the bottleneck link increases.

Table 3. Utilizations of EIMD

Bottleneck (bps)	200M	2.5G	5G
Utilization (%)	86.22	93.34	96.95

Fair-Share Convergence. In this simulation, one high-speed flow with 100ms RTT and background flows start randomly from 0 to 20 seconds. Another high-speed flow with same RTT joins and starts to compete for the bottleneck bandwidth at 80 seconds. We measure the cumulative utilization after 80 seconds. The samples are collected every five seconds. Figure 6 shows fair-share index [8] of each protocol. As shown in Figure 6, EIMD converges to fair-share faster than other protocols. EIMD also maintains the fair-share state without oscillation.

5 Conclusions

We have presented transport protocol called EIMD that can thoroughly alleviate the RTT unfairness problem while supporting bandwidth scalability, and TCP friendliness in fast long-distance networks. EIMD is a window-based congestion control protocol that takes into account RTT and the time elapsed since *cwnd* recovery phase starts. Our simulation results show that EIMD adequately satisfies the three requirements of high-speed congestion control algorithm. In addition, EIMD achieves fast fair-share convergence by increasing *cwnd* in proportional to $1/w_{max}$. We also compare the performance of EIMD with those of HSTCP, Scalable TCP, and BIC in dynamic network environments as well as static network environments. The results of our simulations confirmed that EIMD outperforms these protocols in terms of above four requirements.

References

1. S. Floyd.: HighSpeed TCP for Large Congestion Windows. RFC 3649, December 2003
2. T. Kelly.: Scalable TCP: Improving performance in highspeed wide area networks. ACM SIGCOMM Computer Communication Review, Vol. 33, Issue 2, pp. 83-91, April 2003
3. L. Xu, K. Harfoush and I. Rhee.: Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks. IEEE INFOCOM '04, March 2004
4. C. Jin, D. Wei and S. H. Low.: FAST TCP: Motivation, Architecture, Algorithms, Performance. IEEE INFOCOM '04, March 2004
5. D. Katabi, M. Handley and C. Rohrs.: Congestion Control for High Bandwidth-Delay Product Networks. ACM SIGCOMM '02, August 2002
6. D. Bansal, H. Balakrishnan, S. Floyd and S. Shenker.: Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms. ACM SIGCOMM '01, August 2001
7. S. Jin, L. Guo, I. Matta and A. Bestavros.: A Spectrum of TCP-friendly Window-based Congestion Control Algorithms. IEEE/ACM Transactions on Networking, vol. 11 no 3, pp. 341-355, June 2003
8. D.-M. Chiu and R. Jain.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Computer Networks and ISDN systems, 17:1-14, 1989
9. H. Sivakumar, S. Bailey and R. L. Grossman.: PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks. High-Performance Network and Computing Conference, November 2000

Dynamic Routing Tables Using Simple Balanced Search Trees

Yeim-Kuan Chang and Yung-Chieh Lin

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan R.O.C.
ykchang@mail.ncku.edu.tw

Abstract. Various schemes for high-performance IP address lookups have been proposed recently. Pre-computations are usually used by the special designed IP address lookup algorithms to improve the lookup speed and reduce the memory requirement. However, the disadvantage of the pre-computation based schemes is when a single prefix is added or deleted, the entire data structure may need to be rebuilt. Rebuilding the entire data structure seriously affects the lookup performance of a backbone router, and thus is not suitable for dynamic routing tables. In this paper, we develop a new dynamic routing table algorithm. The proposed data structure consists of a collection of balanced binary search trees. The search, insertion, and deletion operations can be finished in $O(\log N)$ time, where N is the number of prefixes in a routing table. Comparing with the best existing dynamic routing table algorithm, PBOB (Prefix Binary tree On Binary tree), our experiment results using the real routing tables show that the proposed scheme performs better than PBOB in terms of the lookup speed, insertion time, deletion time, and memory requirement.

1 Introduction

To handle gigabit-per-second traffic rates, the current backbone routers must be able to forward millions of packets per second at each port. Thus the IP address lookup is the most critical task in the router. When a router receives a packet, the destination address in the packet's header is used to lookup the routing table. Since maybe more than one route entries (prefixes) in the routing table would match the destination address, it may require some comparisons among these matched prefixes to determine which has the longest prefix length. The prefix with the longest prefix length from all the matched prefixes is called the longest-matching prefix (LMP). The IP address lookup problem becomes a longest prefix matching problem.

Besides, current backbone routers typically run the Boder Gateway Protocol (BGP). To avoid routing instabilities, there are a peak of a few hundred BGP updates per second. These updates can seriously affect the lookup performance of backbone routers. Thus the address lookup algorithm that can support fast updates are desirable.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 389–398, 2006.

Various algorithms for high-performance IP address lookup have been proposed. In the survey paper [10], a large variety of routing lookup algorithms are classified and their complexities of the worst case lookup speed, update time, and memory references are compared. We brief summarize the previous schemes as follows. Schemes like [2], [3], [5], [8], [11] perform a lot of pre-computation and thus improve the performance of the lookup speed and memory requirement. However, the disadvantage of the pre-computation is when a single prefix is added or deleted, the entire data structure may need to be rebuilt. Rebuilding seriously affects the lookup performance of backbone routers. Thus schemes based on pre-computation are not suitable for dynamic routing tables. On the other hand, schemes based on the trie data structure like binary trie, multi-bit trie and Patricia trie [9] do not use pre-computation; however, their performances grow linearly with the address length, and thus the scalability of these schemes is not good when switching to IPv6 or large routing tables.

Sahni and Kim [4] developed a data structure, called a collection of red-black tree (CRBT), that supports three operations for dynamic routing table of N prefixes (longest prefix match, prefix insert, prefix delete) in $O(\log N)$ time each. In [6], Lu and Sahni developed a data structure called BOB (Binary tree On Binary tree) for dynamic routing tables. Based on the BOB, data structures PBOB (Prefix BOB) and LMPBOB (Longest Matching Prefix BOB) are also proposed for highest-priority prefix matching and longest-matching prefix. On practical routing tables, LMPBOB and PBOB permit longest prefix matching in $O(W)$ and $O(\log N)$, where W is 32 for IPv4 or 128 for IPv6. For the insertion and delete operations, they both take $O(\log N)$ time. Suri et al. [12] have proposed a B-tree data structure called multiway range tree. This scheme achieves the optimal lookup time of binary search, but also can be updated in logarithmic time when a prefix is inserted or deleted. Although schemes like [4], [6], [12] all develop a search tree data structure that is suitable for the representation of dynamic routing tables, the complex data structure leads to the memory requirement expanded and reduce the performance of lookup.

Despite the intense research that has been conducted in recent years, schemes that can get the balance between the lookup speed, memory requirement, update time, and scalability are scarce. Actually, for any schemes, it is hard to fulfil all these four issues. In this paper, we develop a data structure based on a collection of independent balanced search trees. Unlike the augmented data structures proposed in the literature, the proposed scheme can be implemented with any balanced tree algorithm without any modification. As a result, the proposed data structure is simple and can get the balance among the four issues described above.

The rest of the paper is organized as follows. Section 2 presents a simple analysis for the routing tables. Section 3 illustrates proposed scheme based on the analysis in section 2 and the detailed algorithms. The results of performance comparisons using real routing tables are presented in section 4. Finally, a concluding remark is given in the last section.

Table 1. Prefix enclosure analysis for three realistic routing tables

Database (year-month)	AS6447 (2000-4)	AS6447 (2002-4)	AS6447 (2005-4)
number of prefixes	79530	124798	163535
Level-1 prefixes	73891(92.9%)	114745 (91.9%)	150245 (91.9%)
Level-2 prefixes	4874 (6.1%)	8496 (6.8%)	11135 (6.8%)
Level-3 prefixes	642 (0.8%)	1290 (1%)	1775 (1.1%)
Level-4 prefixes	104 (0.1%)	235 (0.2%)	329 (0.2%)
Level-5 prefixes	17	29	45
Level-6 prefixes	2	3	6

2 Analysis of Covering and Covered Prefixes

The Border Gateway Protocol (BGP) is the de facto standard inter-domain routing protocol in the Internet. BGP provides loop-free inter-domain routing between autonomous systems, each consisting of a set of routers that operate under the same administration. The address space represented by an advertised BGP prefix may be a sub-block of another existing prefix. The former is called a *covered* prefix and the latter a *covering* prefix. For example, the address block 140.116.82.0/24 is covered by another address block 140.116.0.0/16.

We analyzed three BGP routing tables obtained from [1], and obtained the detailed statistics for the enclosure relationship between the covered and covering prefixes. Theoretically, one prefix may be covered by at most 31 prefixes for IPv4. The prefix and the ones that cover it form a prefix enclosure chain. Therefore, the theoretical worse-case enclosure chain size is 32 for IPv4. Contrary to the definition in [7], we number the prefixes in a bottom-up manner. For example, if a prefix enclosure chain consists of five prefixes P_i for $i = 5$ to 1, where P_5 is the shortest prefix that covers the other four prefixes and P_1 is the longest one that is covered by the other four prefixes. The prefixes like P_1 that do not cover any other prefix in the routing table are called the level-1 prefixes. The prefixes that only contain level-1 prefixes are called level-2 prefixes, and so on. Fig. 1 shows the enclosure relationship between covering and covered prefixes marked with their levels for an example routing table that has the enclosure chain size of 5. Our analysis shows that the chain size is 6 for all the tables we examined. We further show the number of prefixes in each level for all the three routing tables in Table 1. The level-1 prefixes account for about $92\% \sim 93\%$ of the prefixes in a routing table. The level-2 prefixes account for about $6\% \sim 7\%$ of the prefixes. The prefixes in other levels only account for less than 1% of the total prefixes. Since the prefixes in each level are disjoint, it is straightforward to design dynamic routing lookup algorithms with search and update complexity of $O(\log N)$ for a routing table consisting of N prefixes.

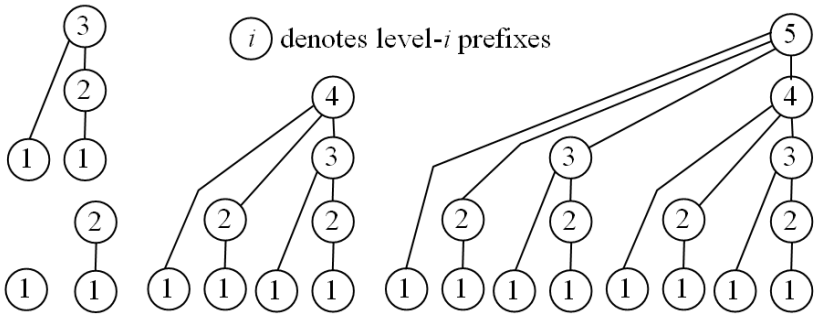


Fig. 1. Enclosure relationship between covering and covered prefixes, assuming the maximum size of the prefix enclosure chain is five

3 Proposed Scheme

From Fig. 1, two important properties of the prefix enclosure relationship can be obtained. The first property is that all the prefixes in one level are disjoint. The second property is that the prefix containing any one of the level-i prefixes must be stored in level-(i+1) or in higher level. Therefore, if we can find a prefix match in level-i prefixes, no search is needed in i+1 or higher level. Assume there are at most s levels in the routing table. Based on the above two properties, we can build s independent data structures for the lookup problem by obeying the *tree* level *constraint* as follows.

Tree Level Constraint: Based on the enclosure relationship between prefixes, the level-i prefixes are stored in the level-i data structure.

Therefore, for a destination address d, if a prefix in the level-1 data structure is found to match d, it must be the only matching prefix in the level-1 data structure. Moreover, this matching prefix must be the longest prefix match. Other prefixes that also match d must be in the higher level data structures. As a result, the higher level data structures do not need to be searched. Furthermore, if the level-1 data structure does not contain a prefix that matches d, we perform the same search process in the level-2 data structure. If a matching prefix is found, it must be the longest prefix match. No other higher level data structure needs to be searched. This search process continues until the level-s data structure is searched, where s is the maximum number of levels.

If the enclosure relationship between prefixes is changed because of insertion or deletion, the locations of some of the prefixes must also be adjusted in order to follow the tree level constraint. In this paper, we decide to use a balanced binary search tree to implement each level of prefixes. Other data structures will be considered in the future. Since the number of levels is a constant and each level is implemented as a balanced binary search tree, the search time complexity must be $O(\log N)$ for a routing table of N prefixes. Fig. 2 shows the search

```
Algorithm Search(d, root[], s)
{ // d is the destination address, s is the number of trees
01  for ( i = 1 ; i <= s ; i++ ) {
02    x = root[i];
03    while ( x ≠ NULL ) {
04      if ( x.prefix ⊇ d ) return x.prefix; // A ⊇ B denotes A covers B
05      else
06        if ( d <x ) x = x.LeftChild;
07        else x = x.RightChild;
08    } // end while
09  } // end for
10  return default_prefix;
}
```

Fig. 2. Algorithm to find the longest prefix match

```
Algorithm Insert(P, root[], s)
{ // P is the newly added prefix, s is the number of trees
01  for ( i = 1 ; i <= s ; i++ ) {
02    x = root[i];
03    while ( x ≠ NULL ) {
04      if ( P = x.prefix ) return;
05      if ( P ⊆ x.prefix ) { /* x.prefix encloses prefix P */
06        Q = x.prefix; x.prefix = P; P = Q; break; }
07      if ( x.prefix ⊆ P ) break;
08      if ( P >x.prefix )
09        if ( x.RightChild = NULL ) {
10          x.RightChild = Create_A_Node(P);
11          BST_Balancing(root[i], x.RightChild); return;
12        } else x = x.RightChild;
13      else
14        if ( x.LeftChild = NULL ) {
15          x.LeftChild = Create_A_Node(P);
16          BST_Balancing(root[i], x.LeftChild); return;
17        } else x = x.LeftChild;
18    } // end while
19  } // end for
20  root[++s] = Create_A_Node(P); // The level is increased by one
}
```

Fig. 3. Algorithm to insert a prefix

algorithm $Search(d, root[], s)$, where parameter d is the destination address and there are s balanced binary search trees.

The insertion of a prefix P is done by performing tree traversals from the level-1 tree to the level-s tree. The main task when traversing the trees is to check if there exists a prefix that covers P or is covered by P. If no such prefix is found, then P is disjoint from all the prefixes in the level-1 tree. And thus, P is

inserted as a leaf node in the level-1 tree. A possible rotation of balanced binary search trees is needed after P is inserted. However, if a prefix Q in the level-1 tree is found to cover P, then Q is replaced by P and the process of inserting Q in the level-2 tree is performed. If a prefix Q is found to be covered by P, the process of inserting P in the level-2 tree is performed. In other words, the same insertion process repeats for trees of level-2 to level-s, where s is the number of trees before inserting a prefix. If P covers all the prefixes in the routing table, a new tree at level $s+1$ will be generated.

Fig. 3 shows the insertion algorithm $Insert(P, root[], s)$ that inserts a prefix P in the balanced binary search trees rooted at $root[1..s]$. After P is inserted in one of the balanced binary search trees, a possible balancing operation (function *BST_Balancing*()) must be performed.

The deletion process of deleting a prefix D first finds out which tree contains D among s balanced binary search trees, assume D is in the level-i tree. There may be a prefix Q that covers D in the level-(i+1) tree. If prefix D is the only prefix that is covered by Q in the level-i tree, then the tree level constraint will be violated after deleting D from the level-i tree. Therefore, in this case, prefix Q must be moved from the level-(i+1) tree and inserted into the level-i tree (i.e., use Q to replace D). The violation of tree level constraint may cause a chain effect to higher level trees. On the other hand, if prefix D is not the only prefix covered by Q in the level-i tree, then anything other than deleting prefix D is not required.

The process of checking whether or not prefix D is the only prefix covered by prefix Q may have a strong impact on the overall process time for deletion. Therefore, we propose an efficient scheme to minimize the time taken for this process. This scheme only checks if prefix Q covers the prefixes Y and Z that are the smallest prefix in D's right subtree and the largest prefix in Ds left subtree, respectively. To explain why we dont need to examine other prefix, it is sufficient to consider Y only as follows. If there is another prefix U that is also covered by prefix Q, then Q must also cover Y because Y locates between D and U. One may argue that the faster way to know if there exists another prefix that is also covered by Q in the level-i tree is to examine the prefixes on the path from the node associated D to the node associated Y one-by-one while traversing the tree and stop as soon as we find another prefix is covered by Q. But we should know that even we find a prefix T covered by Q earlier than reaching the node associated Y, the node associated with prefix Y still needs to be visited because Y can be used to replace D. Therefore, the best way is directly go to the node associated with Y from the node associated with D and checking if Y is covered by Q. Also notice that it is possible that Q is at level-k for $k \geq i+2$ (i.e., no prefixes in the level-m tree cover D, $m = i+1$ to $k-1$). In this case, there must be a prefix enclosure chain for D consisting of a prefix in each level-j for $j = k$ to i. Thus, prefix Q must remain in the level-k tree because of the prefix enclosure chain for D.

Fig. 4 shows the details of the deletion algorithm. The while loop search for the prefix D in each tree rooted at $root[i]$ for $i = 1$ to s. When a node

```
Algorithm Delete(D, root[], s)
{ // y and z are the successor and predecessor of node x containing prefix D
01  for ( i = 1 ; i ≤ s ; i++ ) {
02    x = root[i]; y = z = NULL;
03    while ( x ≠ NULL ) {
04      if ( x.prefix = D ) {
05        q = Search_a_Tree_for_Enclosure(root[i+1], D);
06        if ( q = NULL ) { BST_Delete(root[i], x); return; }
07        else {
08          if ( x.RightChild ≠ NULL ) y = Smallest_Prefix(x.RightChild);
09          if ( y ≠ NULL and q.prefix ⊇ y.prefix ) {
10            x.prefix = y.prefix;
11            BST_Delete(root[i], y); return; }
12          if ( x.LeftChild ≠ NULL ) z = Largest_Prefix(x.LeftChild);
13          if ( z ≠ NULL and q.prefix ⊇ z.prefix ) {
14            x.prefix = z.prefix;
15            BST_Delete(root[i], z); return; }
16          x.prefix = q.prefix; D = q.prefix; break;
17        }
18      }
19      if ( D ⊃ x.prefix ) break; // D contains x.prefix and break inner loop
20      if ( D ⊂ x.prefix ) return; // D does not exist
21      if ( D <x ) { y = x; x = x.LeftChild; }
22      else { z = x; x = x.RightChild; }
23    } // end while
24  } // end for
}
```

Fig. 4. Algorithm to delete a prefix

x associated with D is found in level-i tree, the function $Search\_a\_Tree\_for\_Enclosure(root[i+1], D)$ as shown in line 5 is performed to find a prefix Q that contains D in the level-(i+1) tree. If such Q does not exist, we delete node x from the level-i tree directly by using the standard balanced binary search tree deletion algorithm as shown in line 6. As explained above, we dont worry about if a prefix containing prefix D exists in the higher level tree than i. Lines 7-17 take care when a prefix Q that contains D exists in the level-(i+1) tree. If the right subtree of node x is not empty, it must exist a node y which is the xs successor. Otherwise, xs successor is the node already visited and recorded in line 21. If y exists and y.prefix is contained in Q, we replace x with y and delete node y directly, as shown in lines 10-11. Similar operations are done for the largest prefix in the left subtree of node x.

4 Performance Evaluations

In this section, we present the performance results for IPv4 routing tables. Three BGP tables of different sizes obtained from [1] are used in our experiments. These

Table 2. Performance statistics

(a) Memory requirements (KB)

schemes	AS6447 (79,560)	AS6447 (124,824)	AS6447 (163,574)
PBOB	1,525	2,374	3,101
Proposed scheme	1,330	2,087	2,734

(b) Average search time (microseconds)

schemes	AS6447 (79,560)	AS6447 (124,824)	AS6447 (163,574)
PBOB	1.02	1.37	1.57
Proposed scheme	0.65	0.79	0.88

(c) Average insertion time (microseconds)

schemes	AS6447 (79,560)	AS6447 (124,824)	AS6447 (163,574)
PBOB	0.90	0.89	1.01
Proposed scheme	0.71	0.75	0.76

(d) Average deletion time (microseconds)

schemes	AS6447 (79,560)	AS6447 (124,824)	AS6447 (163,574)
PBOB	0.57	0.57	0.64
Proposed scheme	0.47	0.48	0.49

BGP routing tables reflect the realistic sizes of the routing tables in the backbone routers currently deployed on the Internet. We compare the proposed algorithm with the prefix binary tree on binary tree structure (PBOB) [6]. We only choose PBOB for comparisons because other dynamic schemes (e.g., [4], [12]) do not perform better than PBOB. The performance experiments are implemented in C language on a Redhat Linux platform with a 2.4G Pentium IV processor containing 8KB L1, 256KB L2 caches and 768MB main memory. Moreover, GNU gcc-3.2.2 compiler with optimization level VO4 is used.

Table 2 (a) shows the amount of memory used by each scheme. We can see that the proposed scheme uses about 15% less memory than the PBOB structure. This result can be attributed to that the node structure of our scheme is much simpler than that of PBOB. Besides, each node of the PBOB structure is associated a prefix set, and less than 1% of these prefix sets are empty. For every PBOB nodes that associate the non-empty prefix sets, it needs additional memory to store these non-empty prefix sets (each non-empty prefix set is constructed by an array structure with six entries). To measure the lookup times, we first use an array A to store the address parts of all prefixes in a routing table and then randomize them to obtain the input query address sequence. The time required to determine all the LPMs is measured and averaged over the number of addresses in A. The experiment is repeated 100 times, and the mean of these average times is computed. These mean times are reported in Table 2 (b). Although the worst case search time may be worse than that in PBOB because all the balanced

binary trees must be searched, the average time is better than PBOB. This is because most of the search result can be determined in the level-1 tree. For the average update (insertion/deletion) time, we start by randomly selecting 5% of prefixes from the routing tables. The remaining prefixes are used to build the desired data structures (PBOB and the proposed balanced binary search trees). After the desired data structure is constructed, the 5% selected prefixes are inserted into the structure one by one. Once the selected prefixes are all already inserted, we proceed to remove them from the constructed structure one by one. The total elapsed insertion and deletion times are averaged to get the average insertion and deletion times. This experiment is also repeated 100 times and the mean of the average times is reported in Table 2 (c) and (d). The deletion times for PBOB are obtained by the implementation with the optimized version of the deletion algorithm proposed in [6]. In other words, the empty nodes in PBOB are not removed if they have two children nodes. However, in the proposed scheme, we implement the complete deletion procedure such that as long as a prefix is deleted, the corresponding node in one of the balanced trees is removed and the required rotations are also performed. Even with this implementation difference, the deletion time of the proposed scheme still performs better than PBOB.

5 Conclusions

We have developed a dynamic routing table algorithm based on the prefix enclosure relationship. By applying the proposed algorithm, the routing tables of current backbone routers can be stored as six balanced binary search tree at most. Each of trees consists of a set of disjoint prefixes, and about $97\% \sim 99\%$ of prefixes are stored in the first two trees, the level-1 and the level-2 trees. Since the number of the balanced trees is a constant, the search, insertion, and deletion operations can be finished in $O(\log N)$ time, where N is the number of prefixes in a routing table. Our experiment results show that the proposed scheme performs better than the best existing dynamic routing table algorithm, PBOB [6], in terms of the lookup speed, insertion time, deletion time, and memory requirement.

References

1. BGP Routing Table Analysis Reports, `http://bgp.potaroo.net/`.
2. A. Brodnik, S. Carlsson, M. Degermark, and S. Pink, "Small Forwarding Tables for Fast Routing Lookups," in Proc. of ACM SIGCOMM, pp. 3-14, Sept. 1997.
3. N. F. Huang, S. M. Zhao, J. Y. Pan, and C. A. Su, "A Fast IP Routing Lookup Scheme for Gigabit Switching Routers," in Proc. of INFOCOM, pp. 1429-1436, Mar. 1999.
4. K. Kim and S Sahni, "An $O(\log n)$ Dynamic Router-Table Design," IEEE Transactions on Computers, vol. 53, no. 3, pp. 351-363, Mar. 2004.
5. B. Lampson, V. Srinivasan and G. Varghese, "IP Lookups Using Multiway and Multicolumn Search," IEEE/ACM Transactions on Networking, Vol. 3, No. 3, pp. 324-334, Jun.1999.

6. H. Lu, and S. Sahni, "Enhanced Interval Tree for Dynamic IP Router-Tables," IEEE Transactions on Computers, vol. 53, no. 12, pp. 1615-1628, Dec. 2004.
7. X. Meng, Z. Xu, B. Zhang, G.. Huston, S. Lu, and L. Zhang, "IPv4 Address Allocation and the BGP Routing Table Evolution," in Proc. of ACM SIGCOMM, pp. 71-80, Jan. 2005.
8. S. Nilsson and G. Karlsson "IP-Address Lookup Using LC-trie," IEEE Journal on selected Areas in Communications, vol. 17, no. 6, pp.1083-1092, Jun. 1999.
9. K. Sklower, "A Tree-based Packet Routing Table for Berkeley Unix," in Proc. of Winter Usenix Conference, pp. 93-99, 1991.
10. M. A. Ruiz-Sanchez, Ernst W. Biersack, and Walid Dabbous, "Survey and taxonomy of IP address lookup algorithms," IEEE Network Magazine, vol. 15, no. 2, pp. 8-23, Mar./Apr. 2001.
11. M. Waldvogel, G. Varghese, J. Turner and B. Plattner, "Scalable High-Speed IP Routing Lookups," in Proc. of ACM SIGCOMM, pp. 25-36, Sept. 1997.
12. P. Warkhede, S. Suri, and G.. Varghese, "Multiway Range Trees: Scalable IP Lookup with Fast Updates," The International Journal of Computer and Telecommunications Networking, vol. 44, no. 3, pp. 289-303, Feb. 2004.

On the Use of Balking for Estimation of the Blocking Probability for OBS Routers with FDL Lines*

D. Morató[1] and J. Aracil[2]

[1] Universidad Publica de Navarra, Spain
[2] Universidad Autónoma de Madrid, Spain

Abstract. This paper deals with estimation of blocking probabilities for OBS switches with Fiber Delay Lines (FDLs) and full wavelength conversion. An incoming burst that finds the wavelengths occupied is temporarily stored in a FDL. Hence, contention will be sorted out successfully if the residual life of the system is smaller than the maximum FDL delay. In order to derive the blocking probability, the most accurate methodology to date is the use of *balking* systems [1,2,3,4]. Even though the approach is accurate for very short lengths of the FDLs we identify the cases in which inaccuracy is detected. This happens precisely when the system works with low loss probabilities. Mainly for large number of wavelengths on the fibers and values of the FDL length at least in the vicinity of the burst service time.

1 Introduction and Problem Statement

Optical Burst Switching (OBS) has received considerable research attention as a promising solution for all-optical transmission of data bursts. Burst Control Packets (BCPs) are transmitted through a control channel that is separated from the data channel. Hence, the data payload is transmitted entirely in the optical domain, while the control packet is processed electronically and suffers O/E/O conversion. Such BCPs are sent before the burst is actually transmitted and serve to announce the incoming bursts so that switches can be configured beforehand. However, since resources are unconfirmed, blocking may occur at any switch along the path from source to destination. Thus, OBS is a transfer mode that is halfway between circuit switching and packet switching.

Needless to say, the burst blocking probability is the primary performance measure for OBS networks. This paper is concerned with the analysis of the blocking probability for OBS switches equipped with Fiber Delay Line (FDLs). Since optical buffering is not available at the moment, nor it is a foreseeable technology that will appear in the close future, optical switch designers resort to alternate solutions such as the FDLs. An FDL is a fiber with an specified length, that provides a delay which is equal to the propagation time of a burst in the

* This work was funded by Spanish MEC (project CAPITAL subproject code: TEC2004-05622-C04-04 and project PINTA).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 399–408, 2006.

FDL. Due to the limited delay availability, a buffered burst may be dropped if the output port/wavelength occupation persists when the burst is to exit the FDL.

An architecture named the *Tune and Select* switch is shown in [5]. The switch features N input and output ports. We assume c wavelengths per port, with full wavelength conversion capability, i. e. bursts can be switched from any input wavelength and port to any output wavelength and port.

Assumming traffic destinations are uniform, the analysis can be focused on a single output fiber only. Regardless of the possible internal switch architecture, an abstract model for performance evaluation can be derived with c parallel servers representing the wavelengths per port and a number of fiber delay lines. In the system each FDL has a length L and a maximum storage delay of D_{max} time units. For comparison with previous works we assume the architecture uses variable-delay FDLs, JET signaling and LAUC as the scheduling algorithm [1,6].

In this paper, we wish to study the accuracy of modeling this particular scenario of FDL-equipped OBS switches using a form of queue with impatience, named a *balking* model. The use of balking for the analysis of such FDL-equipped OBS routers has been proposed in other studies [1,2,3,4], with successful results. However, such papers considered a number of wavelengths per port relatively small and they did not check the range of the design parameters where the approximation was accurate. Alternatively, we focus on switch architectures with larger number of wavelengths. For those cases, even with a small number of FDLs the number of bursts that could be simultaneously buffered is large enough to make the length of the FDLs and not the number of fibers as the limiting factor. In this paper we show that a strong mismatch between analytical and simulation results is observed and we reveal that some assumptions about the applicability of the balking model are wrong.

Let us assume that the c wavelengths of an output port are occupied (namely the output port is blocked). An arrival to the system will not enter an FDL *if the delay provided by the FDLs is not large enough to hold the burst during the system blocking time*. In other words, an arrival will not enter the FDL if the output port residual life is larger than the delay provided by the fibers. A queueing system in which arrivals decide on whether to enter the system based on the system state (number of users, current delay, etc) is called a *balking* system or a system with *discouraged arrivals* [7, pp. 123]. For instance, an $M/M/c/K$ system falls within this category, since arrivals will not enter the system if K customers are already inside it.

The balking system lends itself as a good model for analyzing the OBS routers with FDLs. In [8] the $M/M/c/K$ is used as a lower bound. The latter model assumes that up to $K - c$ bursts may fit in the FDLs. However, the number of bursts that fit within an FDL cannot be determined beforehand. This is illustrated in Fig. 1 [2]. The number of bursts that can be accommodated in a single FDL depends on the FDL length and the burst length, which in turn depends on the burst length distribution. For example, [2] reports that the number of bursts per FDL can be arbitrarily large if the burst length is exponential. An improved $M/M/c/K$ approximation is presented in [9], in which K is derived as

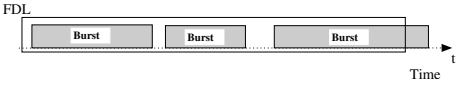


Fig. 1. The number of wavelengths in an FDL depends on the burst length distribution

a function of the FDL length. However, it fails to provide a lower bound as the FDL length becomes larger.

The fundamental limitation of the abovementioned models is that an FDL does not behave as a queue with limited number of customers. Actually, an FDL will reject bursts if the residual life of the servers is larger than the FDL holding time, irrespective of the number of bursts already present in the FDLs. Precisely, the balking models that have been proposed use the FDL holding time as the reject criteria [1,2,3,4].

This paper contributes to the modeling of OBS routers with FDLs by pushing the limits of the balking models. The aim is to identify the limiting cases for which the balking approach is not accurate. While the models proposed in [1,2,3,4] are most valuable and serve to analyse the most common cases of FDL-equipped OBS routers, further insight into the accuracy of balking systems is provided in this paper. In fact, we show that the model is less accurate as the number of wavelengths per fiber increases. This is a serious drawback of balking models, since the foreseeable technological evolution is towards hundreds of wavelengths.

2 Analysis

First, the balking model is presented, as a continuous-time discrete Markov chain. Then, the accuracy of the model is discussed and the causes for deviation with the empirical results are identified.

2.1 A Balking Model for FDL-Equipped OBS Routers

The balking model incorporates the probability that a burst is dropped, i.e. the probability that a burst does not enter the system *because the FDL is too short to hold the burst for the system residual life*. Let $\{X_t, t > 0\}$ be a continuous-time discrete Markov chain that represents the number of bursts in the output port (c servers and FDLs). Let λ be the Poisson arrival rate and μ the service rate, $\rho = \lambda/c\mu$. Let us denote the probability that an incoming burst does not enter the system by β_k for $k = n - c, n \geq c$ where n is the system state (there is balking only for states larger than the number of wavelengths). The arrival rate in state $n > c$ is thus $\lambda_n = \lambda(1 - \beta_{k-1})$, where λ is the output port arrival rate (Poisson). Being consistent with the analysis in [1,2,3,4], the service time distribution will be exponential. Fig. 2 shows a state diagram of $\{X_t, t > 0\}$ with the corresponding arrival and departure rates.

In order to obtain the steady state probabilities π_n, calculation of the probabilities β_k is needed. Let us assume that all wavelengths (servers) are fully occupied and consider the time that a new arrival must wait before it can be

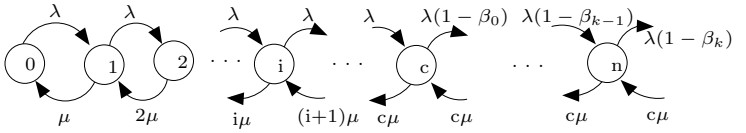


Fig. 2. $\{X_t, t > 0\}$, number of bursts in the output port

served, named the *residual life of the system (wavelengths + FDLs) in state n*, which will be denoted by T_n. Then

$$T_n = \hat{R} + \sum_{i=1}^{n-c} \hat{U}, \quad n \geq c \tag{1}$$

where $\hat{R}$ is the residual life of the wavelengths (the servers) and $\hat{U}$ is the random variable that represents the interdeparture time. Note that an arriving burst will wait for the residual life of the wavelengths plus $n - c$ departures corresponding to the bursts already in the FDLs, due to the PASTA property (Poisson Arrivals See Time Averages).

The residual life of the wavelengths is derived using the residual life of a single wavelength R_1, which is exponential due to the memoryless property.

$$P(\hat{R} > x) = \prod_{i=1}^{c} P(R_i > x) = P(R_1 > x)^c \tag{2}$$

On the other hand, the probability density function is [10, pp. 172]

$$\hat{f}_{R_i}(x) = \frac{P(X > x)}{E[X]} \tag{3}$$

which, for exponential random variables, yields $P(\hat{R} > x) = e^{-c\mu x}$.

Due to the memoryless property the interdeparture times are also exponential and the sum of $n - c$ independent exponential random variables is an Erlang random variable. Thus,

$$\beta_{n-c} = P(T_n > L) = e^{-c\mu L} \sum_{h=0}^{n-c} \frac{(c\mu L)^h}{h!} \quad n = c, c+1 \ldots \tag{4}$$

The Markov chain is solved using the equilibrium equations and the steady state probabilities are

$$\pi_0 = \left(\sum_{i=0}^{c} \frac{(c\rho)^i}{i!} + \rho^c \sum_{i=c+1}^{\infty} \prod_{j=c}^{i-1} \rho(1 - \beta_{j-c}) \right)^{-1}$$

$$\pi_n = \begin{cases} \pi_0 \frac{(c\rho)^k}{k!} & , \quad n < c \\ \pi_0 \frac{c^c}{c!} \rho^k \prod_{i=0}^{k-c-1} (1 - \beta_i) & , \quad n \geq c \end{cases} . \tag{5}$$

Finally, the blocking probability is the ensemble average of blocking probabilities over the wavelength occupation states: $P(blocking) = \sum_{k=0}^{\infty} \pi_{k+c}\beta_k$.

This is the model that has been proposed in [1,2,3,4]. In [4] the Erlang random variable is used to model a single wavelength model ($c = 1$), with special emphasis in the study of FDLs with discrete step allowable delays. In [2,3] the authors use the balking model presented in this section and an improvement with respect to previous models is achieved [8,9]. However, the number of wavelengths is $c = \{2, 3\}$. In [1] no greater number than $c = 10$ is simulated. In this paper, we focus on a scenario with larger number of wavelengths since this is consistent with the foreseeable evolution of optical networks. For example, commercial CWDM routers are available with 8 wavelengths [11] and DWDM prototypes have been reported with 32 [12] and 128 wavelengths [13]. For such number of wavelengths we find discrepancies between the analytical and simulation results.

3 Results and Discussion

A simulation model has been built using the *dsim* [14] building blocks. Such simulation library has also been used in other papers [15,16,17,18]. The wavelength speed is set to 10 Gbps and the number of wavelengths c from 8 to 128.

The burst average size will be set to $15KBytes$, which is the average file size in the Internet as reported by [19], yielding a transmission time $E[X] = 12.288\mu s$. This transmission time is similar in other studies [20]. Switching times will be assumed to be negligible, since SOA-based switches achieve switching times in the vicinity of nanoseconds [21,22,23,24]. Finally, each simulation run consists of 10^8 burst arrivals.

Fig. 3(a) shows the blocking probability versus the normalized FDL length (D_{max} divided by the burst transmission time), for a system with $\rho = 0.94$ and different number of wavelengths c. For $D_{max} = 0$ the blocking time can be approximated accurately by the Erlang-B formula. However, as D_{max} increases, a discrepancy with the model is detected. This is shown in Fig. 3(b) where the percentage of error in the estimation using the model with balking increases as c or D_{max} increase.

In the following subsection the detected discrepancy between the analytical and simulation models is explained in detail. It turns out that neither the probabilities β_k in equation 4 (discouraged arrivals) nor the state probabilities π_n (equation 5) accurately model our case study of OBS router with FDLs. Consequently, the balking model (Fig. 2) hypothesis are revised.

3.1 Discouraged Arrival Probability

From equation 4, the system residual life T_n can be approximated by an Erlang random variable, since interdeparture times are assumed to be exponential. Fig. 4(a) shows the residual life survival function conditioned to the number of bursts (n) in the system (FDLs+wavelengths) $P(T_n > x)$, for $D_{max} = E[X]/2$.

For $n = c$ (or $k = 0$, no bursts in FDLs), the residual life turns out to be exponential as expected. However, as the system occupancy grows larger there

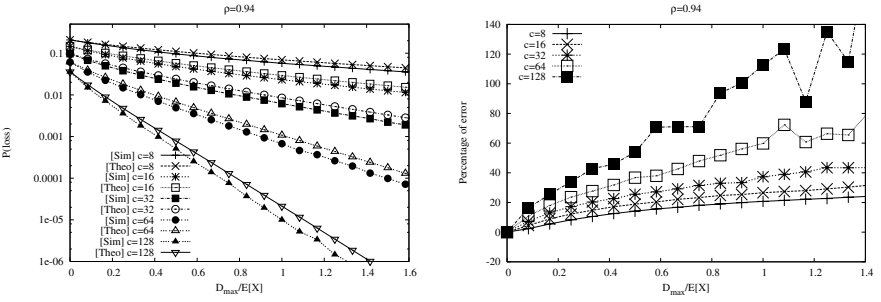


Fig. 3. Burst dropping probability versus normalized FDL length for different utilization factors (Theoretical and Simulation results)

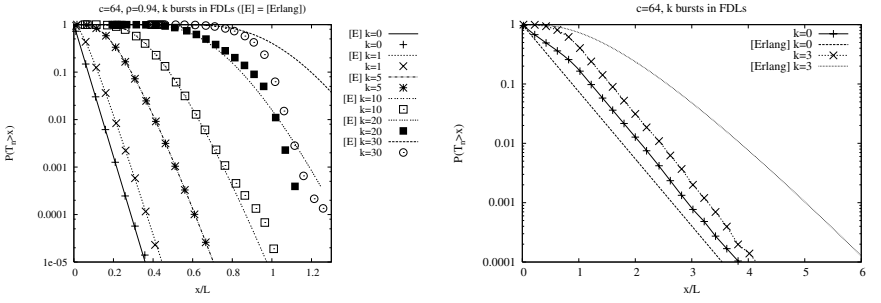


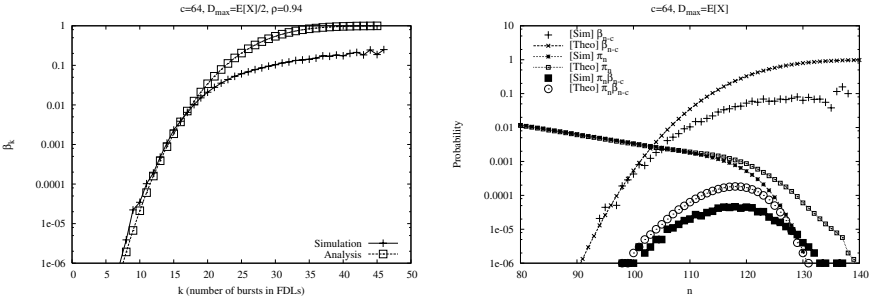
Fig. 4. Residual life survival function, experimental and theoretical

is a significant deviation between the Erlang approximation and the measured residual life. It is also verified by simulation that the discrepancy between the Erlang residual life and the empirical counterpart is larger for residual life values close to the fiber delay. For example, Fig. 4(b) presents a comparison between analytical and empirical residual lives for $D_{max} = E[X]/24$. It shows that even for low n values, the discrepancy between the curves is significant when the residual life is close to D_{max}.

This deviation carries over to the discouraged arrival probability β_k, as shown in Fig. 5(a). As we move to states with larger number of bursts in FDLs the values of β_k obtained from the balking analytical model differ greatly from the simulation results. However, those are the low probability states of the Markov chain. We show in the next section that those states are fundamental in the final value of the loss probability.

3.2 State Probabilities (π_n)

Note that the discouraged arrival probabilities β_k play a crucial role in the calculation of the state probabilities (equation 5). Hence, the discrepancy between analitycal and empirical results in β_k carries over to the state probabilities π_n.



(a) Discouraged arrival probability (β_k) with $D_{max} = E[X]/2$

(b) Comparison between the state probabilities (π_n) and the discouraged arrival probabilities (β_k)

Fig. 5. Influence of the computation of β_k on the state probabilities

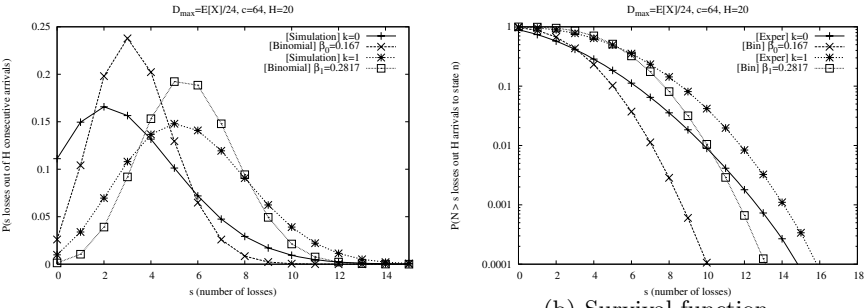
Fig. 5(b) shows empirical versus analytical results for both state probabilities and discouraged arrival probabilities, for a number of wavelengths equal to 64. Both values (β_k and π_n) take part in product form on the calculation of the loss probability. Fig. 5(b) also shows this product $\beta_k \pi_n$. The discrepancy in the discouraged arrival probability and state probabilities happen precisely for high occupancy states with small probabilities of occurrence. However, those are the states where losses take place. Therefore, the deviation from the analytical to the real values in that region of the state-space produces the misbehavior of the loss probability shown in Fig. 3.

3.3 Explaining the Discrepancy Between Analytical and Empirical Results

The above figures show that the discrepancy between analytical and empirical results become more significant as the loss probability is decreased. Hence, the model becomes less accurate for realistic systems of WDM technology, with a higher output degree (number of wavelengths) and lower losses. This is due to the effect of the FDLs on the behavior of the system.

On a congested balking system (system state $n \geq c$), the acceptance of a new arrival depends only on the system state. Even the conditional probability β_{n-c} depends only on the state number n. However, the FDLs act as queues with a maximum delay D_{max}. On a system with FDLs, β_k *depends also on the system residual life*. Independent of the system state, the arrivals that find the system with a residual life longer than D_{max} will get lost.

Consider the arrival process when the system is in state n where $n \geq c$. In a balking system this arrival process is sampled randomly with probability β_{n-c}. We could model the arrivals to the state as the events of a random variable with a value of 1 when the arrival is lost (probability β_{n-c}) and with value 0 when it



(a) Mass probability function (b) Survival function

Fig. 6. Empirical probability of s discouraged arrivals out of H consecutive arrivals to system state $n = c$, compared to a binomial random variable (i.e. random sampling). $D_{max} = E[X]/24$, $c = 64$

gets into the system (probability $1 - \beta_{n-c}$). Then, the distribution of number of losses out of H consecutive arrivals would be Binomial, i. e,

$$P(s) = \binom{H}{s} \beta_{n-c}^{s} (1 - \beta_{n-c})^{H-s} \tag{6}$$

Fig. 6 shows the empirical distribution compared to the values of equation 6 for $H = 20$, taking β_{n-c} from the simulation results. Both distributions differ significantly, and this implies that sampling is not performed randomly in the original process. Actually, when the system occupancy is large, an arrival is discouraged depending on the previous arrivals. Consider a system with c bursts in service and arrivals i and $i + 1$ that happen in state c and $c + 1$ respectively (arrival i is accepted). The residual life of the system will depend on the size (transmission time) of arrival i, and so the event of the arrival $i + 1$ being discouraged or not. Consequently, *the discouraged arrival probability does not depend on the number of bursts in the system solely, but also on the system residual life, which is a continuous random variable.* As a result, the balking Markov model (Fig. 2) cannot be applied.

4 Conclusions

In this paper we study the applicability of the balking model for FDL-equipped OBS routers. We conclude that the balking model accuracy depends on the ratio between fiber delay and service time. If the ratio is large then the balking model is not accurate to derive the blocking probability. Stronger discrepancies between analytical and simulation results are observed as the number of wavelengths per port increases. But precisely, the foreseeable technological evolution is towards hundreds of wavelengths.

References

1. Lu, X., Mark, B.L.: Performance modeling of Optical-Burst Switching with Fiber Delay Lines. IEEE Transactions on Communications **52**(12) (2004) 2175–2182
2. Lu, X., Mark, B.L.: Analytical modeling of Optical Burst Switching with Fiber Delay Lines. In: Proceeding of the 10th IEEE International Symposium on Modeling, Analysis & Simulation of Computer & Telecommunications Systems (MASCOTS'02). (2002)
3. Lu, X., Mark, B.L.: A new performance model of Optical Burst Switching with Fiber Delay Lines. In: Proceedings of the IEEE International Conference on Communications, 2003 (ICC'03). (2003) 1365–1369
4. Callegati, F.: Optical buffers for variable length packets. IEEE Communications Letters **4**(9) (2000) 292–294
5. Gauger, C.M., Buchta, H., Patzak, E., Saniter, J.: Performance meets technology - an integrated evaluation of OBS nodes with FDL buffers. In: Proceedings of the First International Workshop on Optical Burst Switching (WOBS 2003), Dallas, Texas (2003)
6. Xiong, Y., Vandenhoute, M., Cankaya, H.C.: Control architecture in Optical Burst-Switched WDM networks. IEEE Journal on Selected Areas in Communications **18**(10) (2000) 1838–1851
7. Gross, D., Harris, C.M.: Fundamentals of Queueing Theory. 2 edn. John Wiley and Sons (1985)
8. Yoo, M., Qiao, C., Dixit, S.: QoS Performance of Optical Burst Switching in IP-Over-WDM Networks. IEEE Journal on Selected Areas in Communications **18**(10) (2000) 2062–2071
9. Fan, P., Feng, C., Wang, Y., Ge, N.: Investigation of the time-offset-based QoS support with Optical Burst Switching in WDM networks. In: Proceedings of ICC 2002. Volume 5., IEEE (2002) 2682–2686
10. Kleinrock, L.: Queueing Systems. Volume 1. John Wiley and Sons (1975)
11. Networks, N.: OPTera Metro 5100 Multiservice Platform. (2005)
12. Okamoto, S., Eiji Oki, K.S., Sahara, A., Yamanaka, N.: Demonstration of the highly reliable HIKARI router network based on a newly developed disjoint path selection scheme. IEEE Communications Magazine **40**(11) (2002) 52–59
13. Telecom, H.: AMN6100 DWDM, http://www.hitel.com/optical/amn6100.htm. (2004)
14. Morato, D.: DSIM - A C Library for the Development of Discrete Event Simulators, http://www.tlm.unavarra.es/. (2004)
15. Aracil, J., Izal, M., Morato, D., Magana, E.: Multiresolution analysis of Optical Burst Switching traffic. In: Proceedings of ICON 2003, IEEE (2003) 409–412
16. Morato, D., Aracil, J., Diez, L., Izal, M., Magana, E.: On linear prediction of internet traffic for packet and burst switching networks. In: Proceedings of the International Conference on Computer Communications and Networks (ICCCN 2001), Scottsdale, Arizona (2001)
17. Aracil, J., Morato, D.: Characterizing internet load as a non-regular multiplex of TCP streams. In: Proceedings of the International Conference on Computer Communications and Networks (ICCCN 2000), Las Vegas, Nevada, USA (2000) 94–99
18. Morato, D., Izal, M., Aracil, J., Magana, E., Miqueleiz, J.: Blocking time analysis of OBS routers with arbitrary burst size distribution. In: Proceedings of GLOBECOM 2003, IEEE (2003)

19. Downey, A.: Evidence for long-tailed distributiond in the internet. In: Proceedings of ACM SIGCOMM Internet Measurement Workshop 2001. (2001)
20. Buchta, H., Patzak, E., Saniter, J., Gauger, C.: Maximal and effective throughput of optical switching nodes for Optical Burst Switching. In: Proceedings of 4 ITG-Workshop on Photonic Networks. (2003)
21. Ma, X., Kuo, G.S.: Optical switching technology comparison: Optical MEMS vs. other technologies. IEEE Communications Magazine **41**(11) (2003) 16–23
22. Sahri, N., Prieto, D., Silvestre, S., Keller, D., Pommerau, F., Renaud, M., Rofidal, O., Dupas, A., Dorgeuille, F., Chiaroni, D.: A highly integrates 32-SOA gates optoelectronic module suitable for IP multi-terabit optical packet routers. In: Proceedings of Optical Fiber Communication Conference and Exhibit. (2001) 32.1–32.3
23. Chiaroni, D., et al.: First demonstration of an asynchronous optical packet switching matrix prototype for multi-terabit-class routers/switches. In: Proceedings of 27th European Conference on Optical Communication. Volume 6. (2001) 60–61
24. Masetti, F., et al.: Design and implementation of a multi-terabit optical burst/packet router prototype. In: Proceedings of Optical Fiber Communication, Anaheim, CA (2002)

Dropping Policy for Improving the Throughput of TCP over Optical Burst-Switched Networks*

LaeYoung Kim, SuKyoung Lee, and JooSeok Song

Dept. of Computer Science, Yonsei University, Seoul, Korea
{leon, sklee, jssong}@cs.yonsei.ac.kr

Abstract. Burst loss due to contention in Optical Burst-Switched (OBS) networks significantly degrades the throughput of TCP sources in the local access networks because TCP congestion control mechanism makes a TCP source enter a slow start phase regardless of the congestion degree of OBS networks. In this paper, to improve TCP throughput over OBS networks, we first introduce a dropping policy (DP) with retransmission of the bursts dropped due to contention, by the ingress node. Then, we extend the DP with retransmission to drop a burst that has experienced fewer retransmissions in the event of contention in order to reduce the number of events that a TCP source enters the slow start phase due to contention. Additionally, we propose to limit the number of retransmissions of each burst to prevent severe congestion. For the performance evaluation of the proposed schemes, we provide an analytic model of TCP throughput over OBS networks. It is shown via numerical and simulation results that the proposed schemes can achieve better TCP throughput performance than an existing DP without retransmission.

1 Introduction

Optical Burst Switching (OBS) [1] is regarded as a promising technology for the next-generation Internet because it could efficiently support the ever-growing broadband traffic. A major concern in OBS networks is contention which occurs when two bursts contend for the same data channel at the same time. Due to the bufferless nature of OBS networks, contentions randomly occur regardless of the congestion degree of the network. Therefore, many contention resolution schemes such as optical buffering using Fiber Delay Lines (FDLs), deflection routing, and wavelength conversion have been proposed [2]. However, when contention occurs at any core node in an OBS network without any contention resolution scheme or when the degree of contention exceeds the capability limitations of these contention resolution schemes, the core node will certainly drop one burst after selecting a burst to drop based on its dropping policy (DP).

TCP over OBS network has recently received considerable attention since TCP occupies the largest portion of Internet traffic. Burst loss due to contention

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 409–418, 2006.

in OBS networks significantly degrades the throughput of TCP sources in the local access networks because TCP congestion control mechanism makes a TCP source enter a slow start phase regardless of the congestion degree of OBS networks [3]. However, existing works about TCP over OBS networks have mainly focused on the burst assembly [4,5] and there has not been much study about the impact of burst loss on TCP throughput over OBS networks. To avoid TCP sources in the local access networks to judge the network congested from burst loss that occurs when the OBS network is not congested, the authors of [3] have proposed schemes to detect false network congestion. However, these schemes are not practical because they require modifications at the TCP source in the local access networks as well as at the OBS node.

In this paper, to improve TCP throughput over OBS networks, we first introduce a DP with retransmission of the bursts dropped due to contention, by the ingress node. In this proposed scheme, when a contention occurs at any core node, existing Time-based DP (TDP) is used to select a burst to drop. In TDP, the *original burst* that arrives to the core node first, wins the contention while the *contending burst* that arrives to the core node later and raises contention, is dropped [6]. Then, we extend the DP with retransmission to drop a burst which has experienced fewer retransmissions in the event of contention at a core node in order to reduce the number of events that a TCP source enters the slow start phase due to contention. Additionally, we propose to limit the number of retransmissions of each burst to prevent severe congestion. For the performance evaluation of the proposed schemes, we provide an analytic throughput model of TCP over OBS networks. It is shown via numerical and simulation results that the proposed schemes can achieve better TCP throughput performance than an existing DP without retransmission.

The rest of this paper is organized as follows. Section 2 describes the detailed operation of the proposed schemes. Section 3 provides an analytic model for the evaluation of TCP throughput for the proposed schemes. In Section 4, the performance of the proposed schemes is evaluated by means of simulation as well as an analytic model. Section 5 concludes this paper.

2 Dropping Policies for Improving TCP Throughput over OBS Networks

Because TDP uses the arrival time of bursts as the criterion to determine a burst to drop as mentioned in Section 1, it well matches TCP which uses a retransmission timer to activate its congestion control mechanism. Therefore, we first present a TDP with retransmission (TDPwR) to improve TCP throughput over OBS networks. In addition, we propose to limit the number of retransmissions of each burst to prevent severe congestion. For the proposed scheme, a new "*Retransmission-Count*" (*RC*) field is added to the burst header packet (BHP). The detailed steps of TDPwR are illustrated in Fig. 1 and proceed as follows:

- **Step 0:** The ingress node sets *SN* to 1 at its startup. The ingress node uses this *SN* variable to give a unique sequence number to each burst.

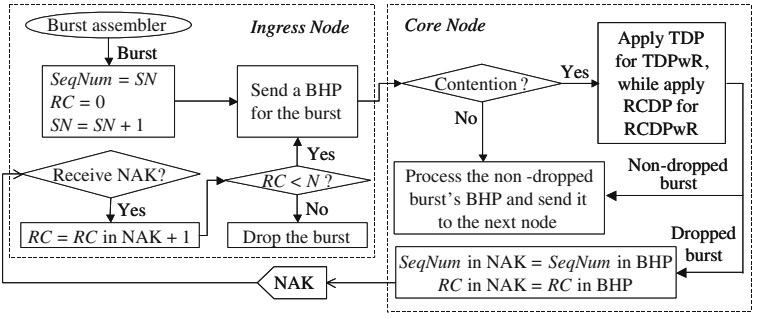


Fig. 1. Operations of the TDPwR and RCDPwR

- **Step 1:** When a burst assembly is completed, the ingress node sets the *SeqNum* and the *RC* fields in the burst's BHP as *SN* and as zero, respectively. Thereafter, it increases *SN* by one for next burst.
- **Step 2:** The ingress node sends the BHP into the OBS network for reserving resources for the burst and keeps the copy of the burst sent along with the sequence number of the burst in its buffer.
- **Step 3:** When a contention occurs at any core node, TDP is applied.
- **Step 4:** For the dropped burst, the steps below will be processed. For the non-dropped burst, go to step 5.
  - **Step 4-1:** A NAK is sent back from the congested node (i.e., core node where the contention occurs) to the ingress node to inform the ingress node of the burst's drop. At the moment, the *SeqNum* and the *RC* fields in the dropped burst's BHP are copied to the *SeqNum* and the *RC* fields in the NAK, respectively.
  - **Step 4-2:** On receiving the NAK, the ingress node sets the value of the *RC* field for the dropped burst as the *RC* in the NAK plus one.
  - **Step 4-3:** If the value of the *RC* field is less than N that is a predefined limit of retransmissions, the ingress node schedules the BHP to retransmit the dropped burst. Otherwise, the ingress node stops retransmitting the dropped burst and discards this burst from its buffer to prevent the network to be severely congested, resulting from retransmitting the same burst successively.
- **Step 5:** The BHP of the non-dropped burst is processed for resource reservation and then sent to the next node.

The above procedure is operated based on the offset-based reservation protocol [1] where a BHP is first sent from an ingress node into an OBS network and then the burst follows the BHP after a base offset time. Each copy of a transmitted burst will be removed from the buffer when the number of retransmissions reaches a predefined limit such as described above in Step 4-3 or when a predefined timer for each burst expires. Note that we name conventional TDP without retransmission as TDPw/oR.

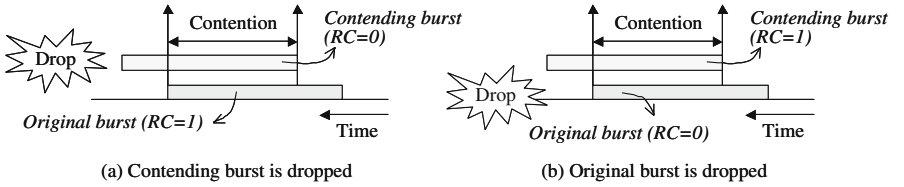


Fig. 2. Retransmission-Count-based Dropping Policy (RCDP)

Now, to reduce the number of events that a TCP source enters the slow start phase due to contention, we extend TDPwR to drop a burst with a lower RC in the event of contention because the higher the RC of a burst is, the less time remains until the Retransmission Time Out (RTO) for the TCP segments contained in the burst expires. We name the DP proposed here as Retransmission-Count-based DP (RCDP) and the extended scheme which uses this RCDP as RCDPwR. The detailed steps of RCDPwR are illustrated in Fig. 1. Because all the steps except DP that is applied by core nodes are same as in TDPwR, we explain here the detailed operation only related to DP.

– **Step 3:** When a contention occurs at any core node, the values of the RC field of the *original* and the *contending bursts* are compared and then the burst with the lower RC is dropped. Fig. 2 (a) shows a case in which the *contending burst* is dropped while Fig. 2 (b) shows a case in which the *original burst* is dropped. If the values of RC are same, TDP is simply applied.

3 TCP Throughput Analysis

In this section, we provide an analytic model for TCP throughput in the cases of TDPw/oR, TDPwR, and RCDPwR. In this study, we focus on fast TCP flows which experience significant degradation of throughput due to the Time Out (TO) event caused by burst loss because all segments in their sending window are assembled into one burst due to high access bandwidth [5]. Our analytic model is developed based on the network architecture shown in Fig. 3. Here t_p denotes the processing and forwarding time of control message such as BHP and NAK at each node. In this figure, T_{edge} is defined as the burst assembly/disassembly time and constant based on a timer-based burstification scheme.

Let n be the number of transmissions of a burst ($1 \leq n \leq N$). That is, n is $RC + 1$. Given the link capacity, B(bps) and the burst length, L(bits), we have the time elapsed in an OBS network, T_{IE} to deliver a burst from an ingress node I to an egress node E successfully, as follows

$$T_{IE} = \sum_{k=1}^{n-1} \left\{ \sum_{i=I}^{c_k-1} t_p + \sum_{i=c_k}^{I-1} t_p + t_w(k) \right\} + \sum_{j=I}^{E-1} t_p + \frac{L}{B} \tag{1}$$

where $t_w(k)$ denotes the waiting time of a dropped burst at the ingress node until it is retransmitted. Generally, a dropped burst is retransmitted after a random

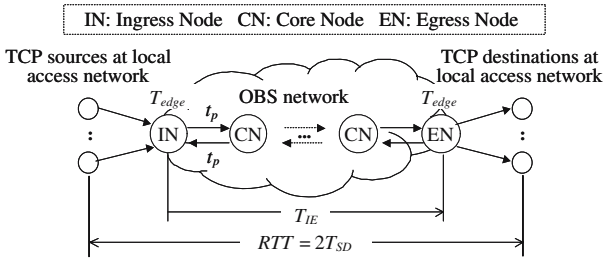


Fig. 3. Overall network architecture with local access networks

backoff time, whose value is evenly distributed between 0 and $2L/B$ with an average of L/B [1]. The first term in Eq. (1) is the time elapsed due to the burst drop at the congested node c_k including the processing and forwarding time of BHP and NAK. The second and the third terms are the amount of time it takes for a BHP and the corresponding burst to be sent from the ingress node to the egress node, respectively. For TDPw/oR, it is obvious that N becomes one.

Now we assume that a contention occurs at a core node with q. Let p_k be the burst drop probability when $n = k$. Then, we have $p_k = q$ regardless of k for TDP since if TDP is applied during contention, the *original burst* always wins and the *contending burst* always drops. And we have $p_k = \frac{(2N-2k+1)q}{N^2}$ for RCDP since the drop probability of the *original burst* and the *contending burst* is $\frac{(N-k)q}{N^2}$ and $\frac{(N-k+1)q}{N^2}$ for $n = k$, respectively.

Let H be the total number of hops on the path from the ingress node to the egress node. Assuming that the TCP segment loss does not occur at local access networks, the state transition diagram for TCP over OBS network is illustrated in Fig. 4. A state is defined as $\overline{s} = \{2^y, k\}$, where:

- $2^y \in [1, M]$ (where $y \in [0, \log_2 M]$) is the number of segments from one TCP source contained in one burst and is equal to the congestion window size of the TCP source since we focus on the fast TCP flow as mentioned earlier. Here, M is the maximum window advertised by the TCP destination at connection establishment time. Thus, a TCP source can send at the maximum M segments at once.
- $k \in [1, N]$ means that a burst containing 2^y segments coming from one TCP source, has been sent k times by the ingress node into an OBS network since the burst has been dropped $k - 1$ times in succession.

We assume that M is an exponent of 2 for simplification of analysis. When a burst containing 2^y segments sent by any TCP source is dropped due to contention at state $\{2^y, k\}$ (for $\forall y$, $1 \leq k < N$), the state will change to $\{2^y, k + 1\}$ since the ingress node retransmits the dropped burst. Note that the probability that there is no available wavelengths on at least one of the H hops can be expressed as $1 - (1 - p_k)^H$. When a burst is dropped due to contention at state $\{2^y, N\}$ (for $\forall y$), the state will change to $\{1, 1\}$ since the ingress node does not retransmit the burst any more. As a result, the TCP source goes into the slow start phase with its

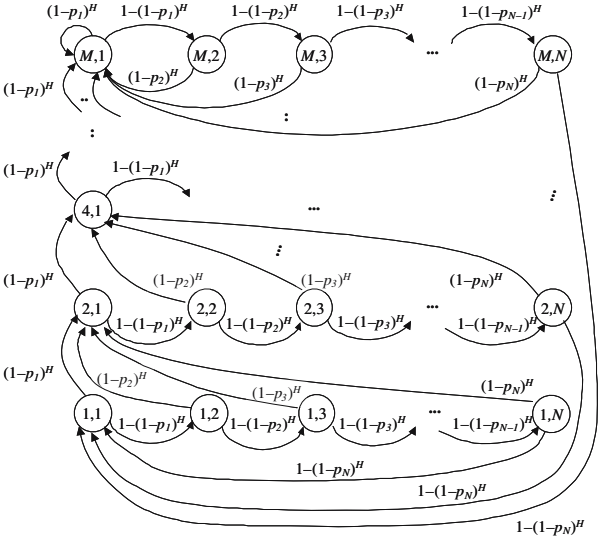


Fig. 4. State transition for TCP over OBS network

congestion window of size one due to the TO event. During the slow start phase, the congestion window grows exponentially. Therefore, when a burst is successfully delivered to its destination at state $\{2^y, k\}$ (for $\forall k$, $0 \leq y < \log_2 M$), the state will change to $\{2^{y+1}, 1\}$. When a burst is successfully delivered to its destination at state $\{M, k\}$ $(1 \leq k \leq N)$, the state will change to $\{M, 1\}$ because the TCP source is allowed to send a maximum of M segments. Let $Q(2^y, k)$ denote the steady probability of state $\{2^y, k\}$. All the steady state equations are as follows:

For $y = 0, k = 1$:

$$\left((1-p_1)^H + \left(1-(1-p_1)^H\right)\right)Q(1,1) = \sum_{i=0}^{\log_2 M}(1-(1-p_N)^H)Q(2^i, N) \tag{2}$$

For $1 \leq y < \log_2 M, k = 1$:

$$\left((1-p_1)^H + \left(1-(1-p_1)^H\right)\right)Q(2^y,1) = \sum_{j=1}^{N}(1-p_j)^H Q(2^{y-1}, j) \tag{3}$$

For $y = log_2 M, k = 1$:

$$\left(1-(1-p_1)^H\right)Q(M,1) = \sum_{j=2}^{N}(1-p_j)^H Q(M,j) + \sum_{j=1}^{N}(1-p_j)^H Q(2^{y-1}, j) \tag{4}$$

For $0 \leq y \leq \log_2 M, 2 \leq k \leq N$:

$$\left((1-p_k)^H + \left(1-(1-p_k)^H\right)\right)Q(2^y,k) = \left(1-(1-p_{k-1})^H\right)Q(2^y, k-1) \tag{5}$$

By solving Eqs. (2)–(5) coupled with the probability conservation relation $\sum_{y=0}^{\log_2 M}\sum_{k=1}^{N} Q(2^y,k) = 1$, we can obtain the steady state probabilities $Q(2^y, k)$.

Here we consider the mean delay involved in each unsuccessful transmission of a burst that is expressed as the first term in Eq. (1). When the transmission of a burst fails j $(1 \leq j < N)$ times in succession, the mean delay in both cases of TDP and RCDP is $E[T_{FAIL,j}] = \left(\sum_{i=1}^{H} 2it_p p_j (1-p_j)^{i-1} \right) + \frac{L}{B}$. Therefore, from Eq. (1), the total mean successful transmission time of each burst including retransmissions for $n = k$ is derived as $E[T_{IE,k}] = (k-1)E[T_{FAIL,k-1}] + Ht_p + \frac{L}{B}$.

Let T_l denote the mean propagation delay in local access networks, i.e., the sum of propagation delay from a TCP source node to an ingress node at the OBS network and the propagation delay from an egress node at the OBS network to the TCP destination node. Under the assumption that the packet loss does not occur at the local access networks, the mean transmission time of a TCP packet from a TCP source node S to a TCP destination node D for $n = k$, $E[T_{SD,k}]$ is given by $E[T_{SD,k}] = T_l + E[T_{IE,k}] + T_{edge}$. The edge delay, T_{edge} is counted from the time of the arrival of the first bit of the first packet to the queue, so that the average queueing delay for all packets aggregated into a single burst becomes $T_{edge}/2$. From this equation, the average round trip time (RTT) of a TCP packet for $n = k$ can be expressed as

$$\overline{RTT_k} = 2E[T_{SD,k}] \tag{6}$$

Finally, we have the TCP throughput using Eq. (6) and the obtained steady state probabilities. Since most recent studies related to TCP over OBS networks are based on TCP Reno [4,5], we also consider the throughput of TCP Reno. Using Eq. (53) in [4] for $Thruput(2^y, k)$, the TCP Reno throughput is expressed as

$$Thruput = \sum_{y=0}^{\log_2 M} \sum_{k=1}^{N} (Thruput(2^y, k) \times Q(2^y, k)) \tag{7}$$

where

$$Thruput(2^y, k) = \begin{cases} \dfrac{2^y/q}{\overline{RTO}\left(\sqrt{(b2^{y+1})/q} + log\sqrt{2^{y+1}/bq}\right) + \overline{RTO}} + o(\sqrt{\frac{1}{q}}) & \text{if } y = 0 \text{ and } k = 1 \\ \dfrac{2^y/q}{\overline{RTT_k}\left(\sqrt{(b2^{y+1})/q} + log\sqrt{2^{y+1}/bq}\right) + \overline{RTO}} + o(\sqrt{\frac{1}{q}}) & \text{otherwise} \end{cases}$$

where b is the number of ACKed rounds before the sending window size is increased (b is typically 2) and $\overline{RTO}$ can be simply expressed as twice the average RTT in the entire networks in case that the delivery of a burst is successful at the first try (i.e., $2 \times \overline{RTT_1}$) [7]. If the TCP source does not receive any acknowledgement for the sent packets until RTO expires, it enters the slow start phase that is the state $\{1, 1\}$ in Fig. 4. Therefore, $\overline{RTO}$ instead of $\overline{RTT_k}$ should be applied for calculating the throughput for state $\{1, 1\}$.

4 Numerical Results

In this section, we show numerical results by means of simulations as well as based on the models introduced in Section 3. All the performance results are evaluated by using the parameters shown in Table 1.

Table 1. Parameter values

Parameters	Values	Parameters	Values
TCP packet size	1 Kbyte	T_l	20 msec
Number of TCP sources per ingress node	5	t_p	5 msec
Access bandwidth for TCP sources	100 Mbps	T_{edge}	$2t_p \times H \times 0.1$ [5]
Link bandwidth in OBS network	10 Gbps		

Figs. 5 (a) and (b) plot the analytic TCP throughput from Eq. (7) versus burst contention probability (i.e. q) for TDPw/oR, TDPwR, and RCDPwR when M=16 and 32, respectively. Here, H is set to 10 and N is set to 3. We see that in all the three schemes, the TCP throughput degrades as the burst contention probability becomes higher. However, from Fig. 5, we observe that TDPwR and RCDPwR performing retransmission achieve much better TCP throughput performance than TDPw/oR at most ranges of the burst contention probability. Specifically, the throughput improvement by TDPwR over TDPw/oR is from 2.18% up to 90.99% and from 2.99% up to 135.42% for M=16 and 32, respectively while compared to TDPw/oR, RCDPwR increases the TCP throughput from 2.18% up to 221.05% and from 3.00% up to 349.62% for M=16 and 32, respectively. It is also observed from these graphs that at high contention probabilities, the performance difference between TDPwR and RCDPwR is noticeable. For example, when the burst contention probability is 10^{-1}, the throughput improvement by RCDPwR over TDPwR is 88.85% and 132.15% for M=16 and 32, respectively. And when the contention probability is 5×10^{-2}, the throughput improvement by RCDPwR over TDPwR is 20.71% and 27.35% for M=16 and 32, respectively. At low contention probabilities, however, the performance difference between two schemes is less dramatic. This means that although both TDPwR and RCDPwR perform retransmission, RCDP outperforms TDP in terms of TCP throughput in case that contention occurs frequently in an OBS network.

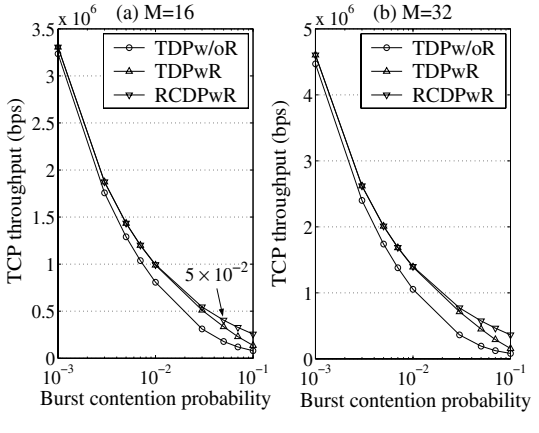


Fig. 5. Analytic TCP throughput versus burst contention probability (H=10, N=3)

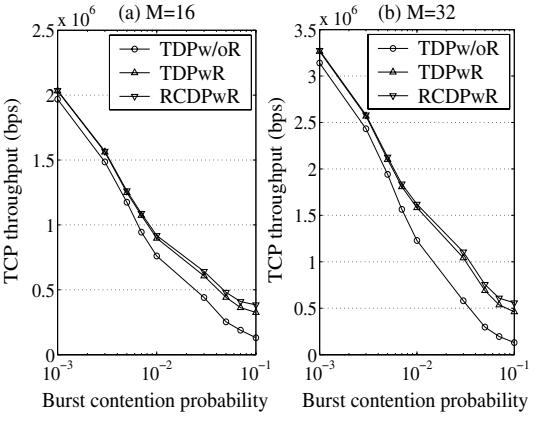


Fig. 6. Simulated TCP throughput versus burst contention probability (H=10, N=3)

Now, the analytic results are verified by simulation where we use the same parameters of Table 1 as done for the analytic model. For our simulation tests, we use the OBS-ns [8] which extends ns-2 with an implementation of OBS components. In our simulation, TCP Reno is operated as considered in the analytic model and FTP traffic is used as TCP flows. Figs. 6 (a) and (b) show the TCP throughput performance from simulation tests for different values of burst contention probability when M=16 and 32, respectively, for H=10. The throughput values shown in this figure are the ones averaged over all the TCP flows. Note in Fig. 6 that TDPwR and RCDPwR achieve much better TCP throughput performance compared to TDPw/oR at most ranges of the burst contention probability as in the analytic results from Fig. 5. Thus, we see that the behavior of the simulation results matches analytic results quite well, as we expected. Moreover, as in the analytic results, the throughput curves in Figs. 6 (a) and (b) show that the TCP throughput improvement by RCDPwR is greater especially for higher contention probabilities, in comparison to TDPwR. For example, when the burst contention probability is 10^{-1}, the throughput improvement by RCDPwR over TDPwR is 17.88% and 20.82% for M=16 and 32, respectively.

Table 2. TCP throughput (bps) by varying N (H=10, M=16)

	Analytic throughput		Simulated throughput	
N	TDPwR	RCDPwR	TDPwR	RCDPwR
2	109348	201916	306613	365826
3	135930	256705	325612	383829
4	158000	267539	336063	391343
5	174578	271478	339257	392613
6	186286	274120	340624	393002
7	194257	276095	341195	393067

In order to capture the impact of N upon the TCP throughput for our proposed schemes, the simulation is repeated with different values of N when the burst contention probability is 10^{-1}. Table 2 shows the analytic results as well as the simulation results. Both the simulated and analytic results show that when N increases, the throughput for both TDPwR and RCDPwR increases as well. This is because the bursts dropped due to contention have more chance to be retransmitted and reach their destination, as N becomes larger. However, we can observe from Table 2 that in the simulated results as well as the analytic results, compared to when N is small (up to 5), TCP throughput increases slowly by increasing N when N is large for both schemes. This is because the TCP source has already entered the slow start phase due to TO event while the ingress node is trying to retransmit the burst dropped several times.

5 Conclusions

To improve TCP throughput over OBS networks, this paper proposed two DPs with retransmission, i.e., TDPwR and RCDPwR. For the proposed schemes, we also systematically investigated the TCP throughput by presenting an analytic model, in comparison with an existing scheme, TDPw/oR. Our analytic results have shown that both TDPwR and RCDPwR significantly improve TCP throughput over OBS networks, and RCDPwR achieves better throughput than TDPwR at high burst contention probabilities. Supporting the analytic results, our simulation results indicated that the proposed schemes improve the performance in terms of TCP throughput compared to the existing scheme.

References

1. Qiao, C., Yoo, M.: Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet, Journal of High Speed Networks, Vol.8, No.1 (Jan. 1999) 69–84.
2. Yao, S., Mukherjee, B., Yoo, S.J.B., Dixit, S.: A Unified Study of Contention-Resolution Schemes in Optical Packet-Switched Networks, Journal of Lightwave Technology, Vol.21, Issue 3 (March 2003) 672–683.
3. Yu, X., Qiao, C., Liu, Y.: TCP Implementations and False Time Out Detection in OBS Networks, IEEE INFOCOM 2004, Vol.2 (March 2004) 774–784.
4. Yu, X., Qiao, C., Liu, Y., Towsley, D.: Performance Evaluation of TCP Implemetations in OBS Networks, Tech. Report 2003-13, CSE Department, SUNY Buffalo, (2003).
5. Detti, A., Listanti, M.: Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks, IEEE INFOCOM 2002, Vol.3 (June 2002) 1803–1812.
6. Vokkarane, V.M., Jue, J.P.: Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks, IEEE Journal on Selected Areas in Communications, Vol.21, Issue 7 (September 2003) 1198–1209.
7. Postel, J.: Transmission Control Protocol, RFC 793 (September 1981).
8. OBS-ns (Optical Burst Switching-network simulator), version 0.9, http://www.oirc.org/index.htm

LBSR: A Load-Balanced Semiminimal Routing Algorithm in Cellular Routers*

Zuhui Yue, Youjian Zhao, Jianping Wu, and Xiaoping Zhang

Department of Computer Science, Tsinghua University, Beijing, P.R. China, 100084
{yuezuhui, zhaoyj}@csnet1.cs.tsinghua.edu.cn,
jianping@cernet.edu.cn,
zhxp@tsinghua.edu.cn

Abstract. In the Internet, the exponential growth of user traffic is driving routers to run at increasing bit-rates and have a very large number of ports. Traditional routers consist of line cards and centralized switching fabrics. However, in such a router, the centralized switching fabric is becoming the bottleneck for its limited ports and complicated scheduling algorithms. Interconnection networks, such as 3-D Torus topology, have been applied to routers. They show excellent scalability and fault tolerance. Unfortunately its scalability is limited in practice. In this paper, we propose a novel architecture called Cellular Router (CR) and give a simple discussion of this architecture. We introduce a load-balanced semiminimal routing algorithm (LBSR) for CRs. This algorithm makes use of path diversity and shows high throughput on tornado and random traffic patterns. We also discuss some other aspects of this algorithm, such as dropping ratio, effects of queue length and speedup.

1 Introduction

Routers play very important roles in the Internet. The continual growth of user traffic requires a corresponding increase in the router capacity. Nowadays most routers consist of line cards and centralized switching fabrics. The switching fabrics receive fixed-size data units (often referred as *cells*) from input ports and forward them to output ports. To meet the growing user traffic, switching fabrics are needed to support both faster ports and more ports. However, there exist a lot of problems [1] which limit the increase of port-rate. So focus has been moved to support more ports.

Historically, switching fabrics based on backplane *buses* and *crossbar* switches are used in routers. It is well known that buses can not be scaled to support high bit rates for its limited bandwidth. For low numbers of ports, crossbar is often selected as the switch topology, owing to the simplicity and non-blocking properties. However, its cost grows as the square of the number of ports, and cannot be economically scaled to a large number of ports. Multistage switching fabric architectures, whose cost growth rate is less than quadratic, can handle

* This work was supported in part by grants 863-2005AA112132 and 973-2003CB314801.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 419–428, 2006.

modest or large numbers of ports. Such topologies have been studied since the days of electromechanical telephony [2]. The *Banyan* network [3] has a low cost of $N * logN$ (N is the number of ports) and a lot of paths. But it suffers from internal blocking. The *Benes* network [4] features a low cost of $N * 2logN$ and is free of internal blocking. However, it is rearrangeably non-blocking, and setting up new connections may destroy the existing ones. The *Clos* network [5] is useful in practice and stimulating in research [6]. There are still some unsolved theoretical problems with it. The *load-lalanced* switch architecture proposed in [7] is considered promising for this approach eliminates scheduler and is amenable to optics [8]. It's a pity that the router described in [8] cannot be built with technology available today.

There exist centralized switching fabrics in the routers implemented with the above architectures. This considerably limits the scalability and the centralized switch becomes the *single point of failure* (SPF). Most of them need complicated schedulers. *Maximum weight matching* (MWM) schedulers, such as LQF and OCF, can achieve 100% throughput asymptotically under any admissible workload, uniform or not, with no speedup needed [9][10]. However, they have a high computational complexity of $O(N^3)$, hence infeasible in practice.

Interconnection networks are originally used as switches, for processor-memory interconnect [11], and for I/O interconnect [12]. Afterwards interconnection networks based on the 3-D torus or *k-ary* 3-*cube* topologies [13] are used as router fabrics in the *Avici TSR* [14][15]. In the Avici TSR switching fabric, each line card carries one node of the torus. There are many optional paths between the source node and destination one. This design offers some good properties [14]: economical scalability, incremental extensibility, load balance, fault tolerance and non-blocking. Although 3-D torus topology shows good scalability, the implementation of TSR has limited the number of line cards which can be added to this system to 560 because of constant bisection bandwidth [14]. The *Cellular Router* (CR) is a new architecture, and can be used to scale routers. There are more optional paths between source-destination pair. With some modifications, this architecture provides better fault tolerance than 3-D torus topology.

The rest of the paper is organized as follows. Section 2 presents the basic CR architecture. Section 3 introduces two minimal routing algorithms. Section 4 presents the load-balanced semiminimal routing algorithm. In Section 5, the performance of the proposed algorithms is evaluated. Section 6 provides a summary of our work and talks about the future work.

2 Cellular Router

Cellular Router is motivated by the 3-D torus topology. It's well known that there are totally only three types of regular polygons which can cover the whole plane seamlessly: *equilateral triangles*, *squares* and *regular hexagons*. Square has been selected as the atomic structure of the *k-ary n-cube* topology. If we add a central point in each regular hexagon, we get a modified version of this topology as shown in Fig. 1.

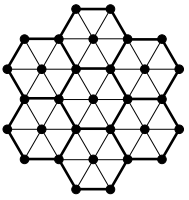


Fig. 1. Modified regular hexagons

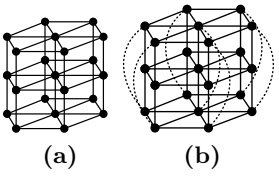


Fig. 2. Improved multilayer scheme

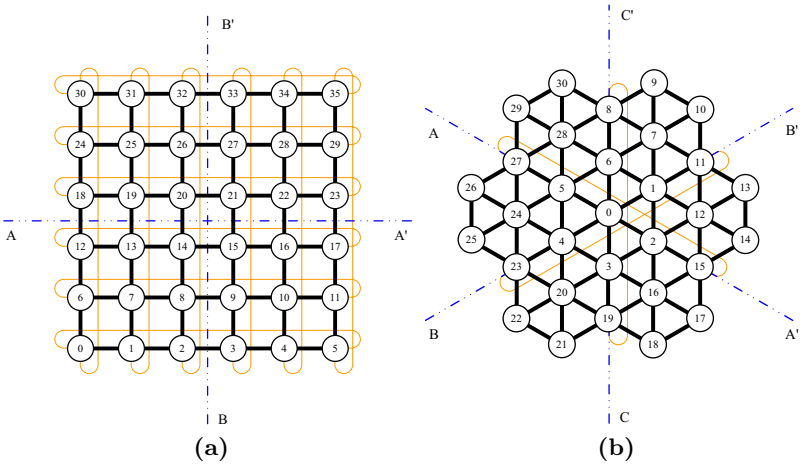


Fig. 3. Improved multi-rack scheme

If each line card carries one node of this topology and data channels replace the edges, we can get a new architecture, called Cellular Router (CR), for scalable routers. As line cards are added, they may appear in only one single layer in one single rack, several layers in one single rack or several different racks. A detailed discussion was given in [16]. This architecture shows excellent scalability and can be easily packaged with short wires. Basic CR architecture shows poor fault tolerance. We can improve this by connecting the edge nodes. Fig. 2b shows the improved multilayer scheme and Fig. 3b shows the improved multi-rack scheme.

In Fig. 3a, we can find that each edge node can find its connected edge node according to the axis AA' or BB'. Analogously we can find such axes AA', BB'

and CC' as shown in Fig. 3b. The degree of a node n is the number of data channels which are connected to n. In 2-D case, the maximum degree of each node in CR architecture is 6. To ease the problem, we only consider the 2-D case. We can easily find that there are only two types of edge nodes in the regular CR architecture, nodes with degree of 3 and nodes with degree of 5. If one node happens to appear on one of the three axes AA', BB' and CC', its connected node can be found in the same axis. To each node with degree of 3, we can connect it to the three symmetric nodes according to the three axes. To each node with degree of 5, if it doesn't appear on any of the three axes, it chooses the axis with the longest distance. As shown in Fig. 3b, node 27 is connected to node 15, and node 29 is connected to node 26, node 10 and node 18. In this way, all the edge nodes have the same degree of 6. This can greatly improve the fault tolerance and reduce the average hop counts each node takes to arrive at its destination.

3 Two Minimal Routing Algorithms

The algorithms described here are minimal ones. That is, they select the shortest paths among all the optional paths. We restrict our discussion to 2-D CR architecture and ignore the impact of edge nodes. Each link is unidirectional, so there are two separate links between any two adjacent nodes. We define the length of each link as 1.

We further assume that the architecture uses *store-and-forward* flow control with each node having buffers of finite length. Whenever contention between cells for the same outgoing link in a node occurs, the oldest cell is chosen. This can isolate the effect of the routing algorithm from flow control issues.

Suppose the source node is s, and the destination node is d. When cells are forwarded from s to d, they pass a series of intermediate nodes: $J = (j_1, j_2, ..., j_n)$. We define D_{sd}, the distance between s and d, as the length of the shortest path between them. For shortest paths, we get $D_{sd} > D_{j_1 d} > D_{j_2 d} > ... > D_{j_n d} = 1$.

We denote the six neighbors of s as n_0, n_1, n_2, n_3, n_4 and n_5. If we connect s and d, there exist two cases. In the first case, d appears on the extended line connecting s and one of its neighbors n_i. In the second case, d appears between two extended lines. When we choose the next-hop node on the shortest path, n_2

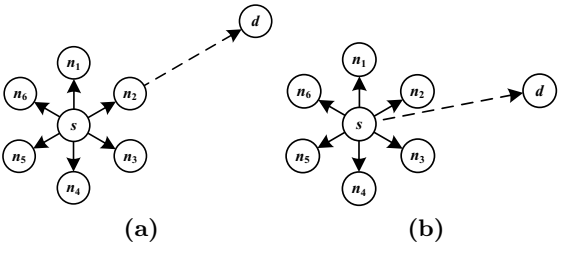


Fig. 4. Relationship between source node s and destination node d

is selected in Fig. 4a while in Fig. 4b we can choose either one between n_2 and n_3. In Fig. 4b, we call n_3 as the *Right Neighbor* (RN) of s and n_2 as the *Left Neighbor* (LN) of s for source destination pair (s, d).

On torus networks, an oblivious algorithm, *Dimension-order Routing* (DOR), was first reported by Sullivan and Bashkow [17]. In DOR, each cell first traverses a certain selected dimension, arriving at the correct coordinate in each dimension before proceeding to the next one. Analogously, we can always choose the Right Neighbor of the current node where the cell locates. We call this algorithm as *right neighbor first* (RNF) algorithm. In the same way, we can design *left neighbor first* (LNF) algorithm. Although these two algorithms only provide a single path between each source destination pair, they are very simple and inexpensive to implement in hardware.

For the selection of next node is deterministic, the RNF or LNF algorithm shows poor performance on adversarial traffic patterns. We can improve this by doing some work on adaptive routing. That is, when there are two next-hop nodes, we can choose the one with lower queue load. When there is only one next-hop node on the shortest path, we choose it without considering the queue state. We call this algorithm as *load-balanced minimal routing* (LBMR) algorithm.

4 Semiminimal Routing Algorithm

For a source destination pair (s, d), we can choose an arbitrary next-hop node j. When we compare the values of D_{sd} and D_{jd}, there exist three cases: $D_{sd} > D_{jd}$, $D_{sd} = D_{jd}$ or $D_{sd} < D_{jd}$. Then we get three types of routing decisions: *advancing routing*, *stagnant routing* and *backward routing*. Normally, we call advancing routing as minimal routing and the combination of stagnant routing and backward routing as non-minimal routing. *Semiminimal routing* is defined as the combination of advancing routing and stagnant routing. In the torus topology, no stagnant routing exists.

In Fig. 5a, when cells are forwarded from s to d, we can choose n_1 or n_2 as the next-hop according to the minimal routing algorithms. One good property

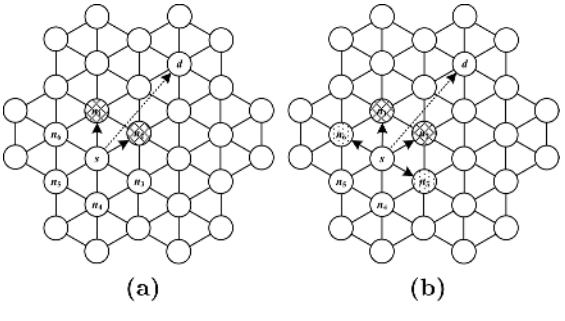


Fig. 5. Minimal routing and semiminimal routing

of interconnection networks is that they provide multiple choices with high path diversity. However, with minimal algorithms there are at most two choices. If neither of them is available, the cell must wait for its turn in the queue. In semiminimal routing, we have much more choices. For example, in Fig. 5b, n_3 and n_6 also can be considered. In this way, we can make better use of path diversity. When some nodes or links fail, minimal routing may not work at all. From this point of view, semiminimal routing also shows better fault tolerance. We can first consider the minimal paths and then the stagnant ones. We also can equally consider all these paths. We will compare them in the following section.

In practice, each node has buffers of finite length. So when we choose the next node, we must consider the queue status of each node. The *load-balanced semiminimal routing* (LBSR) algorithm always chooses the node with lower load.

5 Performance Evaluation

Unlike the architectures of most other routers, where dedicated switching fabrics are need, CR architecture distributes the switching task into each line card. This indicates that each line card should handle not only its own traffic but also the traffic from the neighbors. The input traffic could arrive from the neighboring ingress channels and the ingress link of its own. This input traffic is then distributed to the different output queues. The inner switching fabric can be normal crossbar. We can introduce separated queues, named *Virtual Output Queues* (VOQs), for each output.

It is necessary to develop a simulation system to evaluate the performance of the CR architecture. There are two common approaches for designing such a simulation system: *cycle-based* and *event-driven*. Here we choose the former. Then in each timeslot, one or zero cell arrives at the node and is put into one of the VOQs according to its source node and destination node. To build such a system, we define the following classes: *Router*, *NodeSet*, *LinkSet*, *CellSet*, *Node*, *Link*, *Cell*, *Queue*, *RouteTable* and *RouteItem*. The relationship between them is shown in Fig. 6.

To ease the problem, we choose the *Basic Element* (BE, the modified regular hexagon with 7 nodes as shown in Fig. 7) as our target to evaluate all the

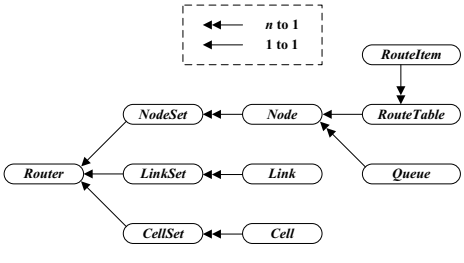


Fig. 6. Relationship between the classes

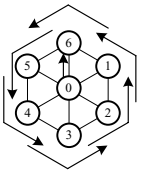


Fig. 7. Tornado traffic pattern in the BE

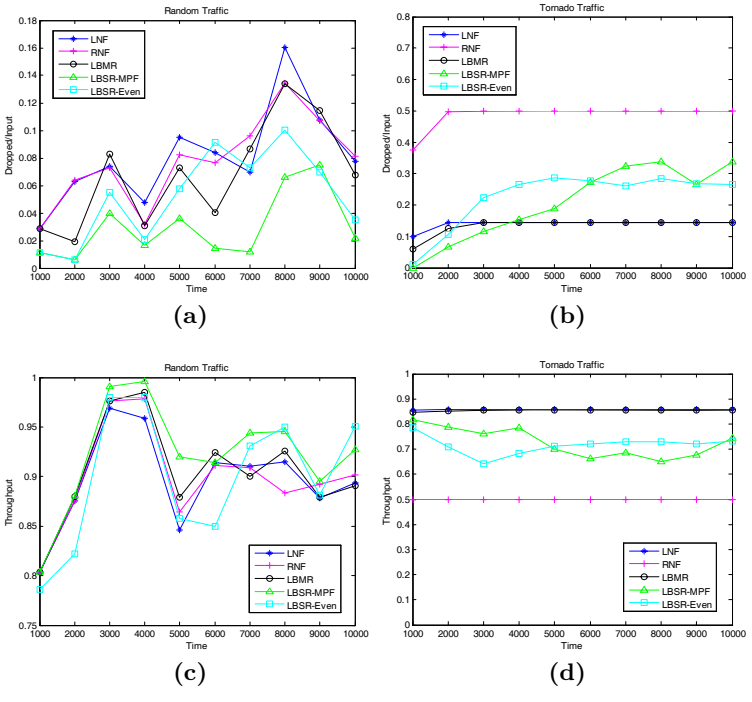


Fig. 8. Dropping ratio and throughput of two traffic patterns

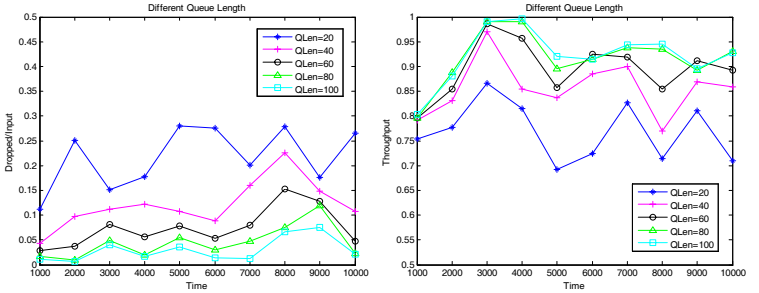
routing algorithms. We choose two traffic patterns: *random traffic* and *tornado traffic*. With random traffic pattern, each node in the BE randomly chooses one destination node from the remaining nodes for each incoming cell. With tornado traffic, node 0 sends cells to node 6, node 1 sends cells to node 5, node 6 sends cells to node 4, etc., as shown in Fig. 7. We assume that at each time slot a cell will arrive at each node.

Our simulator is initialized with empty queues before any cells are injected. This will introduce a systematic error into our measurement. Cells that are injected earlier will see a relatively empty network. These cells have less contention and therefore traverse the network more quickly. However, as buffers begin to fill up; later cells meet more contention, increasing their latencies. Over time the influence of the initialization becomes minimal, and at this point the simulation is

said to be warmed up. By ignoring all the events that happen before the warm-up point, the impact of systematic error on measurements can be minimized.

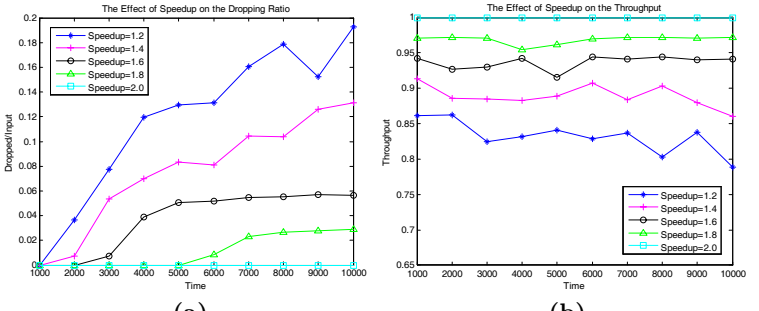
Here the length of each VOQ is set as 100. We run our simulator for 10000 time slots. We will consider the following algorithms: LNF, RNF, $LBMR$ and $LBSR$. With LBSR, if we first consider the minimal paths and then the stagnant ones, we call it $LBSR\text{-}MPF$; in the other end, if we equally consider all the paths, we call it $LBSR\text{-}Even$.

We compare the ratio of dropped cells to input cells and the throughput of tornado traffic and random traffic over the five algorithms. As shown in Fig. 8a and 8b, $LBSR\text{-}MPF$ and $LBSR\text{-}Even$ algorithms show low dropping ratio on both cases, while others are affected greatly by the traffic patterns. For example, RNF algorithm performs well with random traffic, while it performs poorly with tornado traffic. With tornado traffic pattern, RNF algorithm results in considerable load imbalance. The counterclockwise links are fully loaded while the clockwise links are idle. So the throughput remains about 0.5 when the system is steady. If we distribute some load to the clockwise links, we can get better results. The random traffic pattern is benign for RNF algorithm for the load is more balanceable. Analogously, both $LBSR$ algorithms show high throughput with these two traffic patterns.



(a) **(b)**

Fig. 9. Dropping ratio and throughput of random traffic with different queue length



(a) **(b)**

Fig. 10. Effects of speedup on the dropping ratio and throughput with tornado traffic

To develop a routing algorithm, we always assume that each node has buffers of infinite length. However this cannot be true. We choose random traffic pattern, and set the length of each queue as 20, 40, 60, 80 and 100 respectively. The $LBSR\text{-}MPF$ algorithm is chosen. Fig. 9a shows the dropping ratio with different queue length. The influence of queue length is obvious in early time, and becomes more and more inconspicuous. For cells injected early in the nodes see relatively empty queues. These cells can be stored in the queues waiting for their turn. However, as queues begin to fill up; later cells might be dropped for there is no room for them. Over time the influence of the queue length becomes minimal. So it is not a good idea to reduce the dropping ratio only by increasing the capacity of buffers. It is more important to design a good routing algorithm.

Speedup plays an important role in the switching process. This time we choose tornado traffic pattern, and the $LBSR\text{-}MPF$ algorithm is chosen again. We set the speedup of the switching fabrics as 1.2, 1.4, 1.6, 1.8 and 2.0 respectively. As the speedup increases, Fig. 10a and 10b show that the dropped cells are greatly reduced. A speedup of 2.0 is enough for the $LBSR\text{-}MPF$ algorithm with tornado traffic pattern.

6 Summary

As we mentioned at the very beginning of this paper, the driving forces for the evolution of router design is the stupendous growth of user traffic. Now it becomes more and more difficult to increase the speed of ports because we are encountering not only some intrinsic limitations of silicon technology but also a whole set of physical, electrical and mechanical issues. To support a very large number of ports, traditional switching fabrics are not suitable for their complicated schedulers, and centralized switching fabrics show poor fault tolerance. 3-D torus topology used in Avici TSR gives us a pleasant surprise for its simplicity and good scalability. However, TSR can only support up to 560 line cards for its physical limitation. The CR architecture is a new architecture. It is promising and suitable for scalable routers. With some improvements, this architecture works well even when some nodes or links are down.

In future, we plan to go deep into the properties of the regular and irregular CR architectures. We also plan to develop routing algorithms which will be applied to normal operations and some other algorithms which can be used in case of faults. With this architecture, we also can make some research on multicast or QoS switching.

References

1. Fabio M. Chiussi, Andrea Francini. Scalable Electronic Packet Switches. IEEE Journal on Selected Areas in Communications, Vol. 21, No. 4, May 2003, pp. 486-499.
2. M. Marcus. The Theory of Connecting Networks and Their Complexity: a Review. IEEE Proceedings, vol. 65, No. 9, Sept. 1977, pp. 1263-1271.

3. C.-L. Wu and T.-Y. Feng. On a Class of Multistage Interconnection Networks. IEEE Trans. On Computers, vol. 29, No. 8, Aug. 1980, pp. 694-702.
4. V. Benes. Optimal Rearrangeable Multistage Connecting Networks. Bell Systems Technical Journal, vol. 43, No. 7, July 1964, pp. 1641-1656.
5. Charles Clos. A Study of Non-blocking Switching Networks. Bell System Technical Journal, 1953, vol. 32, no. 2, pp. 406-424.
6. Andrzej Jajszczyk. Nonblocking, Repackable, and Rearrangeable Clos Networks: Fifty Years of the Theory Evolution. IEEE Communications Magazine, v 41, n 10, October, 2003, pp. 28-33.
7. C.-S. Chang, D.-S. Lee and Y.-S. Jou. Load balanced Birkhoff-von Neumann switches, Part I: one-stage buffering. Computer Comm., Vol. 25, pp.611-622, 2002.
8. Isaac Keslassy, Shang-Tse Chuang, Kyoungsik Yu, David Miller, Mark Horowitz, Olav Solgaard, Nick McKeown. Scaling Internet Routers Using Optics. ACM SIGCOMM Aug. 2003, Karlsruhe, Germany.
9. N.W. McKeown, V. Anantharam, and J. Walrand. Achieving 100% Throughput in an Input-Queued Switch. IEEE Trans. Comm., vol. 47, no. 8, August 1999, pp. 1260-1267.
10. A. Mekkittikul and N.W. McKeown. A Starvation-free Algorithm for Achieving 100% Throughput in an Input-Queued Switch. Proc. ICCCN '96, Oct. 1996, pp. 226-231.
11. S. Scott and G. Thorson. The cray t3e network: adaptive routing in a high performance 3d torus. In Proceedings of Hot Interconnects Symposium IV, Aug. 1996.
12. G. Pfister. An Introduction to the InfiniBand Arechitecture (http://www.infinibadta.org). IEEE Press, 2001.
13. W. J. Dally. Performance analysis of k-ary n-cube interconnection networks. IEEE Transactions on Computers, 39(6):775-785, 1990.
14. William Dally, Philip Carvey, and Larry Dennison. Architecture of the avici terabit switch/router. In Proceedings of Hot Interconnects Symposium VI, August 1998, pages 41-50, 1998.
15. Available at: http://www.avici.com/products/tsr.shtml
16. Zuhui Yue, Youjian Zhao, Jianping Wu, Xiaoping Zhang. Designing Scalable Routers with a New Switching Architecture. ICAS-ICNS 2005. Joint International Conference on 23-28 Oct. 2005 Page(s):1-1.
17. H. Sullivan and T. R. Bashkow. A large scale, homogeneous, fully distributed parallel machine, I. In Proc. of the International Symposium on Computer Architecture, pages 105-117, 1977.

A Study of Matching Output Queueing with a 3D-VOQ Switch

Ding-Jyh Tsaur[1,2], Hsuan-Kuei Cheng[1], Chia-Lung Liu[1], and Woei Lin[1]

[1] Institute of Computer Science, National Chung-Hsing University,
No.250, Guoguang Rd., 402 Taichung, Taiwan, R.O.C.
{djchou, s9256054, s9056005, wlin}@cs.nchu.edu.tw
[2] Department of Information Management, Chin Min Institute of Technology,
No.110, Hsueh-Fu Rd., Toufen, 351 Miaoli, Taiwan, R.O.C.

Abstract. In this paper, a novel architecture of three-Dimensional Virtual Output Queue (3D-VOQ) switch is proposed. The 3D-VOQ switch requires no speedup and provides an exact emulation of an output-queued switch with a broad class of service scheduling algorithms regardless of its incoming traffic pattern and switch size. First, an N×N 3D-VOQ switch was proposed. In this architecture, input queues were designed with a few virtual output queues (VOQ) to avoid Head-Of-Line problems and output sides were arranged using sufficient separate queues. The combination of this scheme makes switch an input/output contention-free architecture. Next, we propose a Small Time-to-leave Cell First (STCF) algorithm of which it can produce a stable many-to-many assignment. It is also demonstrated and illustrated that the proposed 3D-VOQ switch can be used to mimic an exact OQ switch. Finally, analysis and simulation are employed to verify the performance of 3D-VOQ.

Keywords: switching system, output queueing emulation, QoS, 3D-VOQ.

1 Introduction

The usual Internet backbone is composed of high-speed electrical switches and/or routers. Many commercial switches and routers today employ output-queueing, but, that is impractical for switches with high internal-line rates. Therefore, if the speedup of input queuing switch is increased by one, then the speedup of output queuing switch requires a change by as much as N times [1]. Whereas following that the required speedup of combined input/output queued switch (CIOQ) is between 1 and N. Once the CIOQ switch via the following algorithms: MUFCA [2], CCF [3] and JPM algorithm [4] can emulate its speedup by 2 or 4 times to be simulated as OQ switch.

Otherwise, the fabric and the memory need to run only as fast as the line rate for an input queued (IQ) switch. This makes input queueing very appealing for switches with fast line rates, or with a large number of ports. However, although many scheduling algorithms [5]-[9] proposed for the input-queued architecture can achieve 100% asymptotic throughput, none of these algorithms perform as well as an output-queued switch. The main problem of IQ switching is head-of-line (HOL) blocking, which can severely affect the throughput. If each input maintains a single FIFO, then HOL blocking can reduce the throughput to only 58.6% [10]. Restated, HOL blocking can

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 429 – 439, 2006.

be eliminated entirely using a method known as virtual output queueing in which each input maintains a separate queue for each output. The throughput of an IQ switch can be increased up to 100% for independent arrivals [9].

Input buffered switches with VOQ can achieve 100% throughput [9,11], thus specifying the relationship of proper scheduling with high speed. Existing scheduling algorithms, such as PIM[11], DRR[8] and iSLIP[6], are based on matching parallel and iterative request-grant-accept cycles. However, these algorithms are impracticl and extremely time-intensive. Even iterations may be possible in the worst case. Mei Yang [12] presented a CIOQ switch with Space-division multiplexing expansion by grouping input/output ports (SDMG CIOQ switch) that use extra hardware architecture to reduce the transfer pressure and competition for internal cells based on a space-division concept. This development trend is therefore considered feasible.

This study proposes a novel three-dimensional Virtual Output Queue (3D-VOQ) switch architecture is ever proposed in our previous paper [13]. The physical lines between the input port and output port are increased without using additional hardware. A new algorithm, called Small Time-to-leave Cell First (STCF) Algorithm in this work, which is similar to that proposed by MUCFA [2] is proposed. The next section presents the new switch architecture and composition with 3D drawing. Section 3 outlines the architecture's operational schemes and the operation of the scheduling/matching algorithms. Section 4 provides the system analysis and evaluation of simulation results. Section 5 summarizes the performance evaluation.

2 3D-VOQ Switch Architecture

A 3D-VOQ switch architecture of size N consists of N input buffers at each input, M output buffers at each output, and an N-layer N×M crossbars switch fabric along with a scheduler for each layer. Figure 1 shows the N×N 3D-VOQ switch architecture.

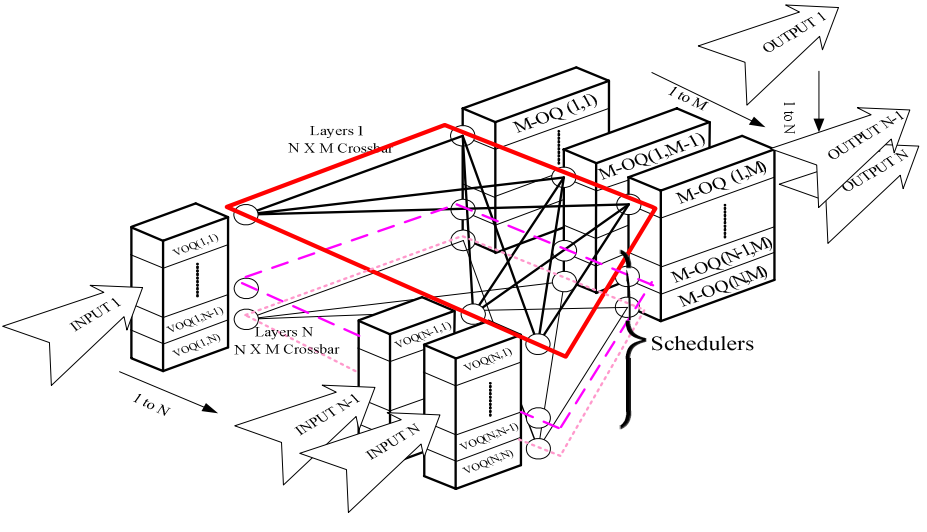


Fig. 1. N×N 3D-VOQ Architecture

This architecture, adopts N input buffers at each input port for virtual output queueing (VOQ) to eliminate a well-known *head-of-line blocking*. An N-Layer N×M crossbars switch fabric, abbreviated to a Layer-j-N×M crossbar, provides N ingress lines for the j^{th} queues of all inputs in all VOQs and M egress lines to output-*j*'s M queues. The output buffer on some output port *j* is divided into Multi Output Queues, which are called M-OQ(*j,k*) where j is in the range 1–N, and *k* is in the range 1–M. Each queue is accessed only once in each time slot without speedup. Each input/output queue is pushed into an arbitrary-out (PIAO) queue, in which cells are removed from the queue in an arbitrary order. Additionally, as illustrated Fig.1, input ports are arranged horizontally from left to right, and output ports are arranged vertically from top to down. The switch fabric based on N-layer N×M crossbars is shown in layers.

As previously noted, cells entering input ports can successfully reach output ports after completing two competitions: transferring VOQ to the switching fabric with internal lines (input contention), and competing for output ports with other input ports (output contention). Contention is described in terms of output *j*. Based on the switch fabric, among all input's VOQs, only those cells of the j^{th} queue are transmitted to (M-OQ(*j,k*), $1 \le k \le M$) via the Layer-*j*-N×M crossbar switch, without competing with those with different destinations. Only a single competition was required for the output ports. For example, all first-layer VOQs were transferred through the first-layer transfer line from VOQ(1,1) and VOQ(2,1) to VOQ(N,1), instead of competing with the other layer cells. An independent small switch was developed after the cells finished entering VOQ at every layer. A 3D-VOQ still cannot avoid output contention, so an efficient algorithm is required above the schedulers.

The schedulers employ a central control unit (CCU), which is called *i*_CCU, located between the input port and the output port. This CCU judges the scheduling of cells transferring from the input port to the output port. An additional CCU, called *o*_CCU is placed on the output side to judge the sequence of cells leaving the switch. The next section describes the operating mechanism of the 3D-VOQ switch in detail.

3 Operation Schemes for 3D-VOQ

This section proposes a novel scheduling algorithm, called Smallest Time-to-leave Cell First (STCF), which enables a 3D-VOQ switch to emulate an OQ switch precisely without speedup for any input traffic.

3.1 STCF Algorithm

The proposed algorithm adopts centralized control to assign the switching to cells. By doing so, the algorithm can perform maximum matching with only one iteration. The centralized control unit is divided into two parts, *i*_CCU and *o*_CCU. First, the main aim of *i*_CCU is to determine which cells must be transferred between the input and output ports. *i*_CCU maintains three tables, *i_table*, *o_table* and *g_table*. The *i_table* indicates input queues requesting status with size N×N, the *o_table* shows available status of output queues by N×(M+1) matrix, and the *g_table* denotes cells routing selection with size N×N. Each row in the matrix denotes an output port, while each column denotes an input port. The M+1 column of *o_table* is the sum of each row.

The steps in the internal operation of *i*_CCU and *o*_CCU are explained below. First, three steps are described below in response to the operating steps of *i*_CCU:

Initialize: Set all elements of *i_table* and *g_table* to ∞ before the 3D-VOQ switch working. The N × M matrix in the o_table is initialized to 1, indicating that the queue is currently available. Column M+1's is initialized to the sum of each row of *o_table*(value at M), i.e., the available size of each output. Figure 2 showes the initialize of a 3×3 3D-VOQ switch.

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{bmatrix} X & Y & Z \\ \infty & \infty & \infty \\ \infty & \infty & \infty \\ \infty & \infty & \infty \end{bmatrix} \qquad \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{bmatrix} X & Y & Z & Sum \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \end{bmatrix} \qquad \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{bmatrix} X & Y & Z \\ \infty & \infty & \infty \\ \infty & \infty & \infty \\ \infty & \infty & \infty \end{bmatrix}$$

(a) *i_table* (b) *o_table* (c) *g_table*

Fig. 2. The initialize *i_CCU* of a 3×3 3D-VOQ switch (t=0)

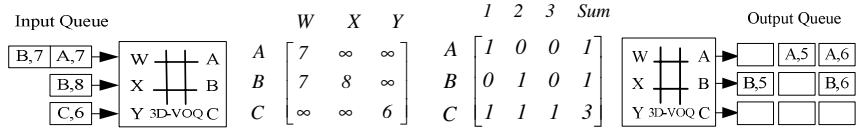
(1) **Request:** Each input sends requests of HOL cell of VOQ TL (Time to Leave) value to *i*_CCU.
(2) **Grant:** Decide whether time slot *t* of input port's cell can be transferred to M-OQ(*j,k*) after receiving the requests from input. First, check all elements of *o_table*. If *o_table*(*j,k*) = 1, then M-OQ(*j,k*) corresponding to output port *j* its *k'th queue* is available (ON).
(3) **Update:** Each M-OQ(*j,k*) sends a feedback represents the state of each M-OQ(*j,k)* to *i*_CCU. The feedback can be used to make a correct decision for next time slot.

Figure 3 illustrates a snapshot of a 3×3 3D-VOQ switch at time slot *t*. Cell (**P**,**T**) is destined to output port **P**, which depart in time slot **T** according to the emulated OQ switch. Figure 3(a) illustrates that *i_table* receives a request from input. The *i*_CCU received requests from HOL cell of each VOQ, which can be recorded as TL in the *i_table* matrix.

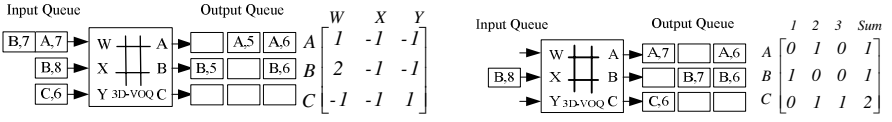
Figure 3(b) demonstrates that *o_table* records the ON/OFF state of each M-OQ(*j,k*). Column (M+1) contains the sum of each row.

Figure 3(c), the row of *i_table* was sorted and used to compare with the *o_table*. We can obtain *g_table* to respond to each VOQ of input. For example, (B,7) and (B,8) are located in the same row in *i_table*, but output B only receives one cell (row B of *o_table* has only one queue in ON state, or the sum of *o_table* is 1). Cell (B,7) is chosen to transmit to output after sorting *i_table*, because cell (B,7) is a smaller TL than cell (B,8). The *g_table* is calculated after comparing two tables, and *i*_CCU responds with grant to input. The HOL cell (A,7) receives grant value 1; cell (B,7) receives grant value 2, and cell (C,6) receives grant value 1. The grant value is the number of the *k*th output queue. If the input and output do not match, then *i*_CCU responds with grant value −1.

In Fig. 3(d), after matching cells, *o_table* requires the latest state of M-OQ(*j,k*) at time slot *t*. The M-OQ(*j,k*) feeds back the state (ON/OFF) at the next time slot to update the *o_table* of *i*_CCU in order to make a correct decision for the next run.



(a) i_table snapshot at time *t* (*request state*) (b) o_table snapshot at time *t* (*grant state*)



(c) g_table snapshot at time *t* (*grant state*) (d) o_table snapshot at time *t*+1(*update state*)

Fig. 3. A snapshot of a 3×3 3D-VOQ switch (t=5)

The operation mechanism of *o*_CCU, which is much simplified than that of *i*_CCU. Only one N × N table, called *d_table* representing departure, is required. The mechanism has two steps: (1) **request:** each HOL cell of M-OQ(*j,k*) sends a request to *d_table* of the *o*_CCU, and (2) **grant:** after receiving requests from each M-OQ(j,k), *o*_CCU only sends one grant to each output port. Only one cell for each output port can be departed from a switch in one time slot. To do it by each output port, the minimum entries in all rows can be searched in parallel. With parallel search, the complexity of computing the *d_table* of the *o*_CCU is *O(n)*.

3.2 Analyze Algorithm and Proof

The following analysis proves that a 3D-VOQ switch with our proposed STCF algorithm can emulate an OQ switch exactly without speedup.

Definition 1: Input Priority List (**IPL**): Each VOQ maintains an ordered list of all queued cells, which can be ordered according to various input ordering schemes.

Definition 2: Output Priority List (**OPL**): Each M-OQ(j,k) also maintains an ordered list. Each queue has an associated OPL, which sets the departure order of cells.

Definition 3: The "time to leave" for cell c, **TL(c)**, is the time slot at which c leaves the shadow OQ switch.

Definition 4: The "output cushion of a cell c", given by **OC(c)**, is the number of cells waiting in the output buffer at cell c's output port with less time to leave than cell c.

Definition 5: The "input thread of cell c", given by **IT(c)**, is the number of cells of smaller time-to-leave than cell c in the input side.

Definition 6: The "slackness of cell c", **L(c)**, equals the difference between the output cushion and input thread of cell c, i.e., L(c) = OC(c) – IT(c)

The proceeding definitions are similar to those presented in [11].

Lemma 1: No input/output contention occurs in the 3D-VOQ switch when N = M, where N denotes the switch size, and M is the number of (M-OQ(j,k), $1 \le j \le N$, $1 \le k \le M$) in STCF algorithm with any traffic.

Lemma 2: A cell with smaller time-to-leave value is transmitted from a 3D-VOQ (N=M) switch much faster than a cell of greatly time-to-live value using the STCF algorithm.

Theorem 1: A 3D-VOQ switch that uses STCF algorithm should have identical behavior to an OQ switch under any traffic.

Proof: *Lemma 1* and *Lemma 2* show that the time needed for a cell to enter and depart 3D-VOQ can be found if the designated TL has the same sequence and time as the sequence and time of the OQ switch. Thus, cells entering the 3D-VOQ and OQ switches always have identical behavior.

Lemma 3: The slackness L of Cell c waiting on the input is never less than zero in any time slot.

Theorem 2: Regardless of the incoming traffic pattern, a 3D-VOQ switch using STCF without speedup can emulate an OQ switch exactly.

Proof: Assume that the 3D-VOQ switch has successfully emulated the OQ switch up until time slot $t-1$. Consider the beginning of time slot t (arrival phase). We must indicate that any cell reaching its time to leave is either (i) already at the output side of the switch, or (ii) to be transferred to the output during time slot t. *Lemma* 3 shows that the slackness L of cells waiting on the input is never non-negative. Consequently, if a cell has reached its time to leave (i.e., its output cushion and input thread both equal zero), then either (i) it is already at its output, and may depart on time, or (ii) it is at the HOL and has the smallest TL. In case (ii), the STCF algorithm is guaranteed to transfer the cell to its output during the time slot, so the cell departs on time.

4 System Simulation and Performance Analysis

The 3D-VOQ cell delay in an input queue was first analyzed. Then, the performance of the OQ, CIOQ and 3D-VOQ switches were analyzed under different traffic models. Finally, a simulation was performed to show the exactly emulative OQ switch.

4.1 Analysis of 3D-VOQ Delay

The switch interconnection is an architecture of N layers and N×M crossbar switches. The model and switch analysis are based on the following assumptions:

- The switch operates synchronously.
- Cells arrive at the beginning of a time slot, and depart only at the end of a time slot.
- The arrival of cells follows the independent and identically distributed (i.i.d.) Bernoulli process, and cell destinations are uniformly distributed over all outputs.
- Every VOQ in an input has the same buffer size B.
- The M-OQ queues of output ports are assumed to consist of M queues with size L.

Under these assumptions, the probability of a request is first derived from a VOQ, which is denoted as Ps and which can be successfully serviced by the scheduler. The request is created from a VOQ at a rate φ. As cell competition on 3D-VOQ switch occurs at input ports, the calculation of theoretical values becomes focused on input

ports. Equations (1)–(6) show the theoretical derivation of various states within the switch, and their convergence values are obtained by iterative calculation. The following are lists of notations and their corresponding formulae.

(1) **λ**: offered load for every VOQ.

(2) **ρ**: occupancy of VOQ, namely, the probability of cells staying at VOQ. The calculation formula is as follows:

$$\rho = \frac{\lambda (1 - Ps)}{Ps (1 - \lambda)} \tag{1}$$

(3) **φ**: probability required by VOQ to transfer the cells, namely, the probability of possessing cells within VOQ.

$$\varphi = \lambda + \rho - \lambda\rho \tag{2}$$

(4) Blocking rate: the probability that the cells are blocked at VOQ when they enter the switch.

$$B_{(N*M\_switch)} = \frac{1}{N\varphi} \sum_{i=m+1}^{N} (i-m) \frac{N!}{i!(N-i)!} \varphi^i (1-\varphi)^{N-i} \tag{3}$$

(5) Service rate: the rate of unblocking that cells successfully send out from the input ports. The formula is as follows:

$$Ps_{(VOQ\_service)} = 1 - B_{(N*M\_switch)} \tag{4}$$

(6) VOQ delay: B denotes the length of a queue. According to the well-known M/M/1/B model [14], the VOQ delay results are as follows:

$$D_{(Delay\_VOQ)} = \frac{\rho\left[1 - (B+1)\rho^B + B\rho^{B+1}\right]}{(1-\rho^{B+1})(1-\rho)} \tag{5}$$

(7) Throughput: the throughput per input port is the total number of N queues sent out with request rate φ plus a successful service rate Ps, i.e., throughput of every input port can be given as follows:

$$T = N * \varphi * Ps \tag{6}$$

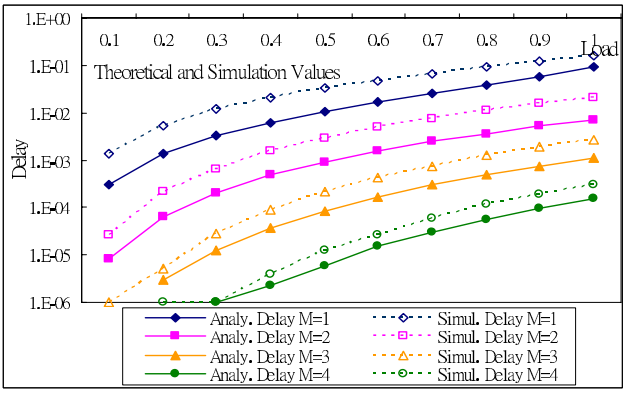


Fig. 4. Theoretical values and Simulation Values

Figure 4 compares the numerical results obtained from the proposed analysis model in comparison with simulation results under switch size of N=16 and length of VOQ=10, and shows that the average delay in simulations has a higher value than that derived from Eqs. (1) to (5), because theoretical values are calculated under optimum state. In this graph, the queueing delay in input buffers decreases as the M value of M-OQ rises. Significantly, the average delay of a cell in input buffers is less than 10^{-4} in a 3D-VOQ switch where M-OQ = 4. That is, only one in every 10^4 cells is delayed.

4.2 Performance Evaluation of OQ, CIOQ and 3D-VOQ Switches

When confronting different traffic patterns, the OQ, 3D-VOQ and CIOQ switches tend to show dissimilarity. Three traffic models were applied in the simulation to observe their differences.

Figure 5 depicts Bernoulli arrival traffic, on-off traffic and polarized traffic. The figure shows that the throughput of 3D-VOQ and OQ switches better than the CIOQ switch. In the case of bursty traffic, the throughput of the OQ, 3D-VOQ and CIOQ switches decreases slightly, while that of the CIOQ switch decreases more steeply than others. Hence, the CIOQ switch performance is unstable among all three switches.

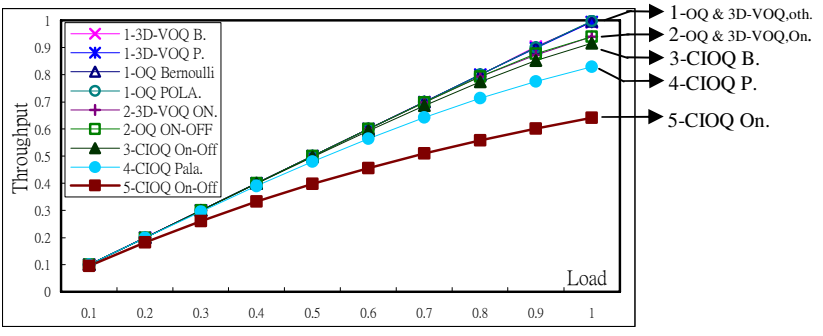


Fig. 5. Comparison diagram of throughput among 32×32 OQ, 3D-VOQ and CIOQ switch

Figure 6 illustrates the average cell delay under different traffic load. OQ switch and 3D-VOQ switch signify the highest value regarding the average cell delay. This result arises from the fact that when input cells need to be transmitted to a specific output port and only one cell is allowed to leave the switch output port at one time-slot. The average cell delay of CIOQ switch is smallest. But, the average cell delay under Bernoulli traffic is greater than that under busty traffic for CIOQ switch. This phenomenon is opposite to that observed from both OQ switch and 3D-VOQ switch, which can be attributed to the drop rate.

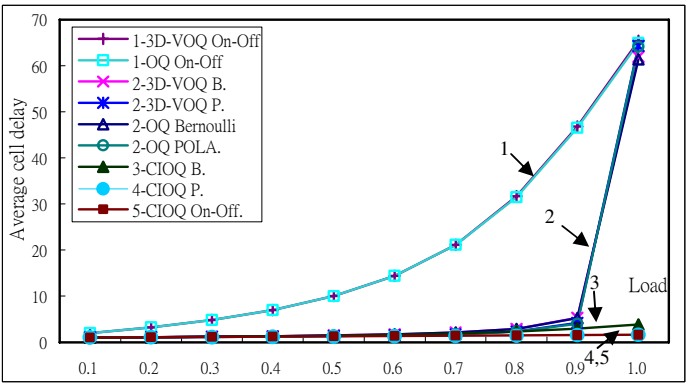


Fig. 6. Comparison diagram of delay among 32×32 OQ, 3D-VOQ and CIOQ switch

In Fig. 7, it can be seen that CIOQ switch has the highest drop rate. This finding results from the fact that VOQ of input sides can only provide limited memory space. So, cells will be dropped before they enter the input sides, and then average cell drop rate will be increased as well.

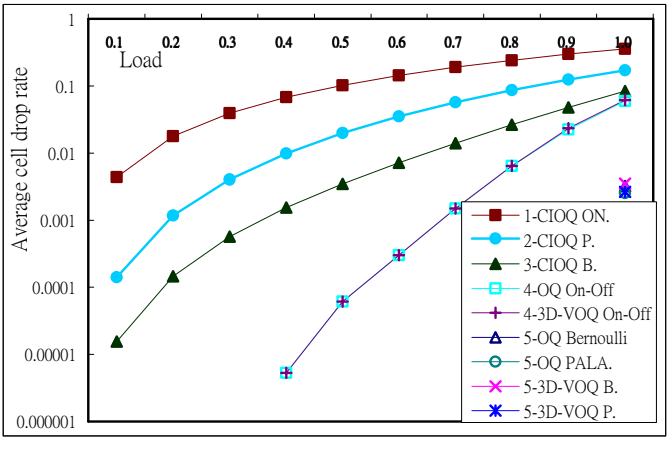


Fig. 7. Comparison diagram of drop rate among 32×32 OQ, 3D-VOQ and CIOQ switch

4.4 3D-VOQ Emulate OQ Switch by Simulation

Section 3 analyzes the STCF algorithm, which can emulate an OQ switch via a 3D-VOQ switch. This section simulator was used to show cells entering the 3D-VOQ and OQ switches. Each point in Fig.8, denotes one cell. Figure 8 reveals that the cell with ordering will has identical delay times in both the OQ and 3D-VOQ switches in every case. In conclusion, for the same traffic to enter into 3D-VOQ switch and OQ switch their behaviors are identical.

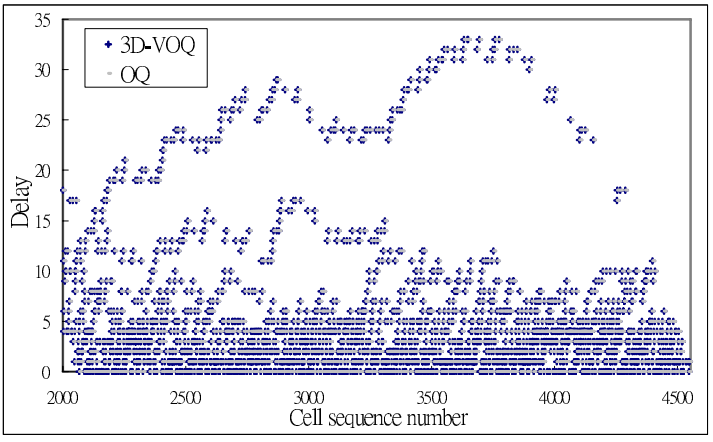


Fig . 8. Comparison of OQ and 3D-VOQ cell when entering and departing switch

5 Conclusions

The 3D-Virtual Output Queue switch adopts the 3D concept to improve the original switches of the plane structure. Moreover, packet switching within the switches can be operated efficiently, thus reducing competition within original switch architecture.

This study shows that the 3D-VOQ switch can emulate exactly an output queued switch with no speedup. This result holds for all arriving traffic patterns. That is, any size of switches and a broad class of service scheduling algorithms including FIFO, WFQ and strict priority queueing are applicable using this design. An N × N 3D-VOQ switch with sufficient separate output queues was found to make switching an input/output contention-free architecture. This study also proposes the STCF algorithm that can produce a stable many-to-many assignment. Additionally, the 3D-VOQ switch was found to be able to emulate an exact OQ switch.

The performance of 3D-VOQ was verified by analysis and simulation. Furthermore, the primary concept of QoS, of applying cells of different priorities and strict priority mechanism to achieve the desired packet switching, was incorporated into 3D-VOQ.

References

1. Hyoung-Il Lee and Seung-Woo Seo: Matching Output Queueing with a Multiple Input/Output-Queued Switch. *IEEE INFOCOM 2004. Vol. 1,* (2004) 07-11
2. B. Prabhakar, N. Mckeown: On the speedup required for combined input and output queued switching. Automatica, Vol. 35, (1999)
3. S. T. Chuang, A. Goel, N. Mckeown, B. Prabhakar: Matching output queueing with a combined input output queued switch. IEEE J. S. Areas in Commun. v17,(1999) 1030-1039
4. I. Stoica and H. Zhang: Exact emulation of an output queueing switch by a combined input output queueing switch. *Proc. 6th IEEE/IFIP IWQoS'98,* (1998) 218-224

5. N. Mckeown: Scheduling Algorithms for Input-queued Cell Switches. Ph. D. dissertation, Univ. California at Berkeley, (1995)
6. N. Mckeown: The iSLIP Scheduling Algorithm for Input-Queued Switches. IEEE/ACM Transactions on Networking, Vol. 7, No. 2, (1999) 188-201
7. R. O. LaMaire and D. N. Serpanos: Two Dimensional Round-Robin Schedulers for Packet Switches with Multiple Input Queues. IEEE/ACM Trans. Networking, v. 2 (1994) 471-482
8. H. J. Chao, Saturn: A Terabit Packet Switch using Dual Round-Robin. IEEE Commun. Magazine, Vol. 38, (2000) 78-84
9. N. McKeown, V. Anantharam, J.Walrand: Achieving 100% Throughput in an Input-Queued Switch. *INFOCOM '96,* (1996) 296-302.
10. M. Karol, M. Hluchyj, S. Morgan: Input Versus Output Queueing on a Space Division Switch. IEEE Trans. Comm, vol.35, no.12, (1987) 1347-1356
11. T. Anderson, S. Owicki, J. Saxe, and C. Thacker: High-Speed Switch Scheduling for Local-Area Networks. ACM Transactions on Computer Systems, Vol. 11, No. 4, (1993) 319-352
12. Mei Yang and S.Q. Zheng: An Efficient Scheduling Algorithm for CIOQ Switches with Space-Division Multiplexing Expansion. *IEEE INFOCOM 2003, Vol.3* (2003) 1643-1650
13. Ding-Jyh Tsaur, Xian-Yang Lu, Chin-Chi Wu, Woei Lin: 3D-VOQ Switch Design and Evaluation" *IEEE 19th International Conference on AINA, vol.2* (2005) 359-362.
14. D. Gross and C. M. Harris: Fundamentals of Queueing theory. 3rd Edition. Wiley, (1998)

BGP Route Selection Notice*

Wang Lijun, Xu Ke, and Wu Jianping

Department of Computer Science and Technology,
Tsinghua University, Beijing, China
{wlj, xuke}@csnet1.cs.tsinghua.edu.cn, jianping@cernet.edu.cn

Abstract. The present Internet is not trustworthy, partially because the routing system forwards packets only according to destination IP address. Forged packets with mendacious source IP address will also be brought to the destination, which can be utilized to compromise the destination machine. In this paper, we propose to enhance BGP by adding Route Selection Notice functionality. With BGP Route Selection Notice, Autonomous Systems can validate the authenticity of incoming IP packets and filter out improper packets to make routing infrastructure offer support to trustworthy service. BGP Route Selection Notice does not impair the routing function of BGP and with proper design its bandwidth cost and convergence delay is acceptable which is proved by our simulation.

1 Introduction

The next generation Internet is necessary to be more trustworthy to sustain many security-sensitive applications. As an important Internet infrastructure, the present routing system provides *best effort* service which forwards packets only according to destination IP address. When a host receives a packet from network, it identifies the remote sender by the source IP address. However, the host can not tell the source IP address is mendacious or genuine, which makes Internet not trustworthy. To meet the need of Internet development, service provided by the routing system should be enhanced to be trustworthy, which will includes two aspects: forwarding IP packets to destination properly and guaranteeing the packets forwarded are genuine.

Internet routing has two levels, intra-domain routing and inter-domain routing. BGP [1] is the *de facto* standard of the latter which is used among Autonomous Systems(ASes).To construct the trustworthy Internet, routing system firstly should be able validate the genuineness of packets on a coarse granularity, that is ASes only permit in or transmit packets coming from the ASes where the packets should come from. This paper concentrates on inter-AS validation which makes hosts can not disguise hosts in other ASes or use other improper source address, such as not assigned IP address, to serve the devil. Indeed, this

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB314801, and Hi-tech research and development program of China under Grant No. 2005AA112130.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 440–449, 2006.

can only guarantee the packet comes from the true AS but not the true host, but we think this is the first step toward the trustworthy Internet.

We think over to guarantee packets validity according to the hierarchy of Internet routing system. Compare to intra-domain routing, inter-domain routing is not so much dynamic in despite of the instability observed in [2]. So routing-based Distributed Packet Filtering(DPF) [3] is a more practical method used in inter-domain routing than in intra-domain routing. On another aspect, routing system mainly can be divided into data plane and control plane. The packets forwarding function of data plane is support by routing protocols. The packets validation is implemented in data plane, it is also reasonable that the control plane provides the information used by the validation. So, we prefer to extend BGP to provide the validation criterion to be used in border routers.

Route is propagated like a rumor in BGP: the receiving AS is not sure the received route is true or not and the propagating AS is not sure whether the routes sent to neighbors are selected or not. For the latter, if a route of specific destination is not selected by a neighbor, then no packets to the destination should come from the neighbor, otherwise, packets going to the destination should be forwarded through the propagator. But in current BGP, the propagator knows nothing about the route sent to other ASes, including whether is selected and if selected, selected by ASes with what IP address space. A border router just receives whatever sent from neighbor ASes and forwards them to destinations properly. So the present inter-domain routing has no guard against potential threat which makes Internet untrustworthy.

In this paper we bring forward extending BGP with Route Selection Notice function. The main idea is if an AS selects a route from a neighbor, it sends a message containing the destination prefix of the route and the IP address space of its AS to the propagator, which informs the propagator that packets to the destination are valid only if the source addresses is in that address space. We design a new BGP message, SelectionNotice, to complete this function. SelectionNotice messages pass through the reverse path of BGP route propagation until reach the original AS of the route. Each border router along the path records the address space in the message and all such information store in border routers as Route Selection Information which is used to construct packets validation criteria. In the design, bandwidth cost and convergence time of Route Selection Information are the main concerns we think over. Selection Notice timer is introduced to improve the performance by piggyback more source address space of different ASes for a route. In simulation, we found the extension has little negative impact on routing function of BGP.

The rest of this paper is organized as follows. In the next section, we give a summary of related works. The design principles of Route Selection Notice are introduced in Section 3. In the following Section, we present the architecture of the extended BGP in detail. Simulation result will be presented in Section 5. Several issues relating to Route Selection Notice is discussed in Section 6. Finally, we conclude this paper in the last Section.

2 Related Work

IP traceback [4] is a type of method to cope with the harm of untrustworthy event by tracing back packets to find the real origin. However, on one hand IP traceback is a remedy taken after harm produced, on the other hand it engenders significant computation and storage overhead to routers.

Packet filtering is a network mechanism to eliminate the questionable packets in the networks. Filtering decision is usually based on some fields of IP packet. [5] brings forth forwarding table based filtering, but unfortunately, routing asymmetry causes this method disabled since many legitimate packets may be dropped. Ingress filtering [6] puts traffic filters on edge routers only allowing packets with source address of the edge networks. The main concern is Ingress filtering requires to be broadly deployed in Internet to take effect while there is no incentive for every AS to be compliant.

Spoofing Prevention Method(SPM) [7] is brought out which provides protection for destination networks. In SPM a unique temporal key is associated with each ordered pair of source/destination networks. Each packet leaving a source network S is tagged with the key $K(S; D)$, where D is the destination network. Upon arrival at the network S, the key is verified and removed. Thus the method verifies the authenticity of packets carrying the addresses which belongs to network S.

In [3] the authors systematically present the route-based distributed packets filtering method and analyze the performance of DPF in Internet topology with power-low property. ASes judge the validity of packets according to the information provided by routing system indicating packets with specific source IP address come from which router port. Through simulation the authors indicate that deploying DPF in minority of ASes will eliminate majority of spoofing packets. Nevertheless, the authors do not point out how the routing system provides the information used to make filtering decision.

Source Address Validation Enforcement(SAVE) [8] is a new protocol in which every router sends a SAVE update with its local address space to each destination in the forwarding table. The routers on the traveling path of a SAVE update will record the address space and construct a source address table mapped to corresponding port. On receiving a packet, the router validates it according to the source address table of the incoming port.

3 Design Principles

In contrast with designing a new protocol, we consider to extend the present routing protocol to provide necessary information to validate packets. We present how Route Selection Notice enables packets validation in the topology of Figure 1 with six ASes, with AS number 1, 2, 3, 4, 5, and 6, each denoted by a circle. To simplify the problem, each AS has only one border router (named A, B, C, D, E, and F) and one IP address space (p1, p2, p3, p4, p5, and p6 respectively). The edges between two ASes denote physical connections. Directed edges denote

provider-customer relationship between two ASes and nondirected edges denote peer-to-peer relation.

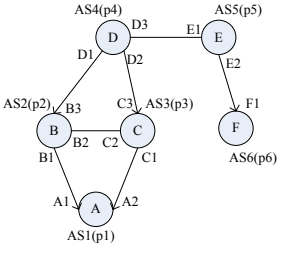


Fig. 1. A simple topology

Table 1. Route selection information of router B

Prefix	Next hop	AS path	Route selection info
p1	A1	1	4p4
p2		local	1(p1),3(p3),4(p4)
p3	C2	3	
p4	D1	4	1(p1)
p5	D1	45	1(p1)
p6	D1	46	1(p1)

After a border router has propagate a route to its neighbors, it does not know whether the route is selected or not and whom the route is further propagate to if selected by specific neighbor. So, if a packet with specific source IP address coming to the border router, it has no knowledge to judge whether the source IP address of the packet is authentic or not. Our main idea is if a border router selects a route from a neighbor, it informs the neighbor the selection of that route and the IP address space of local AS. To support this functionality a new message type is add to BGP4, named SelectionNotice. We call all the SelectionNotice messages recorded by a border router Route Selection Information.

We choose AS2 to explain the procedure of acquiring the Route Selection Information. In the following discussion ⟨m⟩ eans a BGP message and [ ] means a sequence of IP prefix. Router B will receive Update message ⟨p1, A1, 1⟩ from customer AS1, ⟨p1, D1, 431⟩ , ⟨p3, D1, 43⟩ , ⟨p4, D1, 4⟩ , ⟨p5, D1, 45⟩ , ⟨p6, D1, 456⟩ from provider AS4, and ⟨p1, C2, 31⟩ , ⟨p3, C2, 3⟩ from peer AS4. The first part in Update message is the IP prefix of destination network, the second part is the IP address of nexthop, and the third part means the AS path. BGP route decision process installs selected routes in BGP Loc-Rib. For each route in Loc-Rib, B sends a SelectionNotice message to inform the neighbor which has sent the route, that is, ⟨p1, p2⟩ to AS1, ⟨p3, p2⟩ to AS 3, ⟨[p4,p5,p6], p2⟩ to AS4. The former part of SelectionNotice message means the destination address of the selected route and the latter means the IP address space of AS2. Also B send Update messages to its neighbors, ⟨p2, B1, 2⟩ , ⟨p3, B1, 23⟩ , ⟨p4, B1, 24⟩ , ⟨p5, B1, 245⟩ , ⟨p6, B1, 2456⟩ to AS1, ⟨p1, B2, 21⟩ , ⟨p2, B2, 2⟩ to AS3, and ⟨p1, B3, 21⟩ , ⟨p2, B3, 2⟩ to AS4. The neighbors send to AS2 SelectionNotice messages after they have selected the routes from AS2. That is, AS1 sends ⟨[p2, p4, p5, p6], p1⟩ , AS3 sends ⟨p2, p3⟩ , and AS4 sends ⟨[p1, p2], p4⟩ . Router B's Route Selection Information is illustrated as column 4 in 1, in which 4(p4) means AS4 has selected the route to p1 from AS2 and will send packets to p1 with source IP address belonging to p4. With this information, if AS3 forwards a packet to AS2 with source IP address not belonging to p3, router in AS2 can pick it out as forged packet

because it knows AS3 only sends packets to it whose destination IP addresses belong to p2 and source IP addresses belong to p3.

However, after router D has selected the route from B, it will send Update ⟨p2, D3, 42⟩ , ⟨p1, D3, 421⟩ to AS5 and AS5 will further send ⟨p2, E2, 542⟩ , ⟨p1, E2, 5421⟩ to AS6. Then packets coming from AS5 and AS6 destine to p1 and p2 will be forwarded to B. If B executes packets validation, the packets from AS5 and AS6 will be picked out as forged. The key point of the problem is as a route spreads, SelectionNotice message is only transmitted to nexthop AS and the more anterior ASes do not know the selection information. So we conclude such a rule for the propagation of SelectionNotice message: AS x should forward a SelectionNotice Message to the nexthop AS if the route does not originate from it. (Here the nexthop AS is the AS sent the route to AS x. We call it nexthop AS because AS x will forward packets to this AS as nexthop.) According to this rule, router D will forward ⟨[p1, p2], p5⟩ , ⟨[p1, p2], p6⟩ from AS5 and AS6 to AS2.

After all the BGP Update message and SelectionNotice message have been transmitted and processed, each border router has stable BGP Loc-Rib and Route Selection Information. Even in 1 there is chance for forged packets to evade from validation. However, in BGP without Route Selection Notice, hosts in every AS can forge arbitrary source IP address.We think that is a great progress and through extending BGP with Route Selection Notice the effectiveness indicates in [3] can be achieved and the work in this paper is the first step to perfection.

4 Extension Design

The architecture of extended BGP is illustrated as the part enclosed by solid lines in 2. And the part enclosed by dashed line is the process of data plane. The control plane provides packet validation information to data plane. Update message and SelectionNotice message are handled in parallel. Update messages are processed as specified in [1]. The difference is if a new route selected, at the same time advertising the new route to neighbors a SelectionNotice message is sent to the neighbor from whom receives the route. As to SelectionNotice message, BGP process will affirm whether the route acknowledged by the SelectionNotice message is indeed sent to that peer. If so, the address space in the SelectionNotice message will be recorded in Route Selection Information. If the route acknowledged does not origin from local AS, the router will forward the SelectionNotice message to the nexthop AS who sent the route to it.

If the border router is configured to execute packets validation, when generating packet forwarding table from BGP Loc-Rib, it also generates packets validation criteria from the Route Selection Information. How to validate the packet using this information is another topic which has been discussed adequately in [3]. If an AS aggregates its SelectionNotice message with the ones sent by "Downstream" ASes, then the AS locating "Upstream" will process and transmit fewer messages. For example in 1, without aggregation totally 11 SelectionNotice messages will be sent to acknowledge the route originate from AS1, otherwise, the best case is 5. To aggregate SelectionNotice messages we also

design Selection Notice timer. After BGP process has selected a new route into Loc-Rib and sent Update message to peers, it do not send the SelectionNotice message immediately to the nexthop AS, but startup a timer to wait for the SelectionNotice message coming from "Downstream" ASes. Until having received SelectionNotice message from all "Downstream" peers or timeout event happening, BGP process sends a SelectionNotice message to the nexthop AS with local address prefix and the ones from "Downstream" ASes.

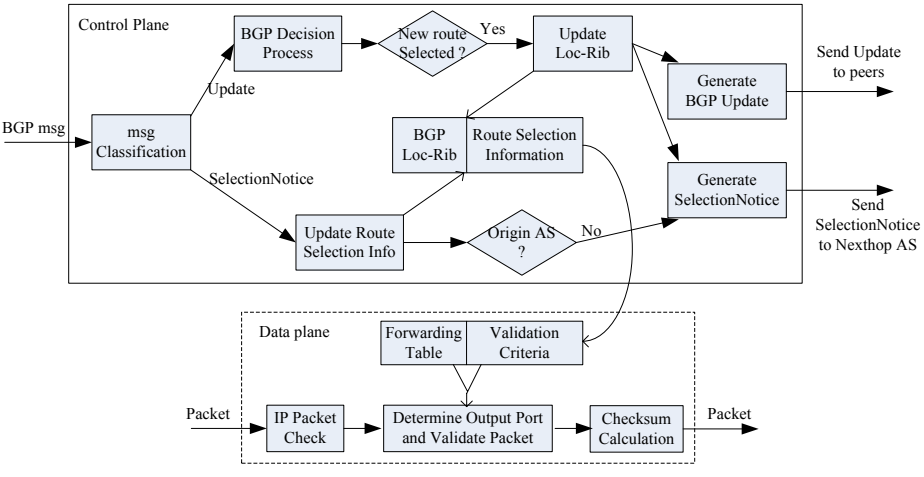


Fig. 2. Architecture of BGP Extended with Route Selection Notice

5 Simulation

We have implemented Route Selection Notice in the BGP protocol of SSFNet [9] simulator. The simulation is to observe the communication overhead of the extension and its effect on BGP convergence property. Also we want to find out how to set the Selection Notice timer properly to balance the communication overhead of SelectionNotice message and convergence time of Route Selection Information. In the simulation, we use net29 and net110 topology [10] as the Internet topology which are generated from a BGP table dump.Every AS in net29 and net110 consists of only one border router whose process delay for each BGP message uniformly distributes between 0.1 second and 0.5 second. The transmission delay between border routers is 1.0 second.

5.1 Effect on BGP Convergence

We first compare the convergence time of BGP route (not including Route Selection Information) and the total number of Update message transmitted in the whole network before and after extension. Three instances are observed for the comparison: (1) standard BGP without Route Selection Notice extension;

(2) BGP extended with Route Selection Notice, but without Selection Notice timer; (3) BGP extended with Route Selection Notice, with Selection Notice timer set. The result of three instances have totally the same Update message number and convergence time. This means the Route Selection Notice extension and the Selection Notice timer have not at all any effect on BGP convergence. In the architecture of our design (Figure 2), the disposal of SelectionNotice message is parallel with the disposal of Update message, so it is not surprising that the extension has no effect on BGP convergence behavior.

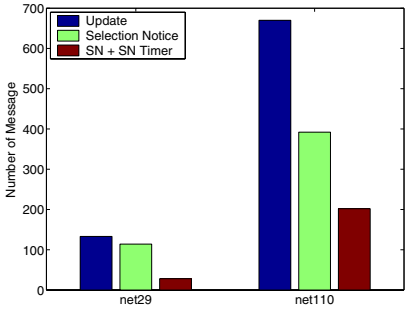


Fig. 3. Route Selection Notice communication overhead

5.2 Bandwidth Cost

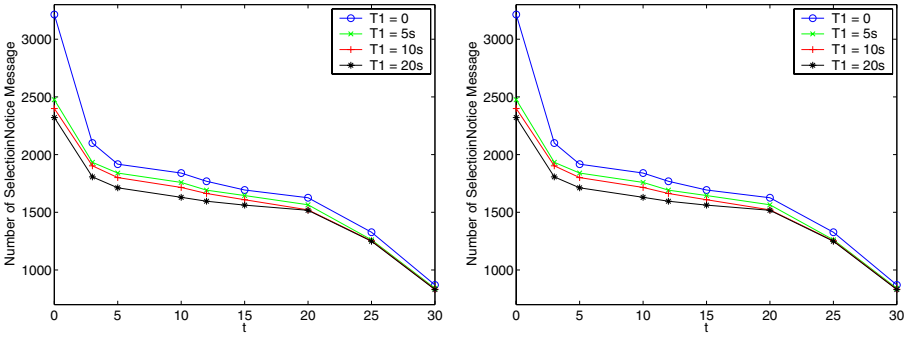
To assess the bandwidth requirements of Route Selection Notice, we compared the bandwidth cost of SelectionNotice message with BGP Update message. In the experiment, we log the Update messages and SelectionNotice messages sent while the network reaches a stable state after a route change. In Fig.3 the number of Update message and SelectionNotice message for net29 and net110 is illustrated. It can be seen that the number of SelectionNotice message is obviously less than the number of Update message and if enabling Selection Notice timer (SN + SN Timer) the difference is more obvious. The format of a SelectionNotice message is more compact than that of Update message, so we think the total Bandwidth cost is acceptable.

5.3 Effect of Selection Notice Timer

To piggyback SelectionNotice message, we have designed Selection Notice timer. We divide the value of Selection Notice timers into two parts: BGP route convergence time T_1, and transmission and process delay time T_2. When a route is announced, the receiving router may receive multiple routes from different neighbor successively. Each newly received route will trigger decision process and a Route Selection Notice message may be sent. The function of T_1 is to wait the router reach the final route, thus the router will only sent ONE Route SelectionNotice.

The T_2 timer is used to estimate the time for the SelectionNotice message to return back. A severe question is the "Upstream" AS does not know whether a neighbor will select the route and whether the route will be further propagate to other ASes. To piggyback more SelectionNotice message, T_2 should be set according to the last SelectionNotice message, which is still hard to estimate. In implementation the value of T_2 is determined by $T2 = t * (L_{max} - L_{as\_path})$, in which L_{max} means the longest AS path in the BGP Loc-rib of the border router, L_{as_path} means the length of the AS_PATH attribute of the route and t is the guess value of average transmission and process delay between neighboring ASes. This is certainly not an accurate estimate, however, form the viewpoint of the AS it is reasonable to think the route will further go through at most $L_{max} - L_{as\_path}$ ASes.

So the convergence process of Route Selection Information is determined by the value of t and T_1. To control routing traffic overhead when sending Updates BGP uses MRAI timer whose default value is 30 seconds. The MRAI timer will prolong the convergence time of BGP route [11] also make Selection Notice timer longer. The influence of Selection Notice timer is illustrated in Fig.4. It can be seen that when $t < 5$ the number of SelectionNotice message reduces evidently as t increases. When $t > 5$ the reduction of message is not very obvious. And the influence of T_1 is not as notable as t.



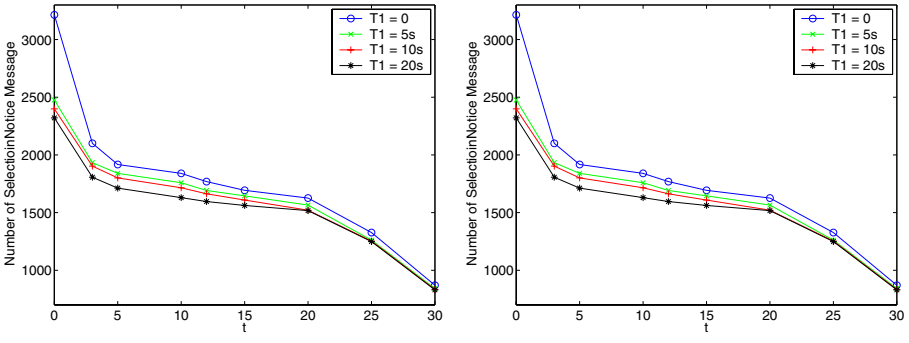
(a) net29 (b) net110

Fig. 4. The influence of Selection Notice timer on the Communication Overhead

5.4 Convergence Time

Though the Selection Notice timer reduces the number of SelectionNotice message, it also brings some problems: (1) adding overhead to the border routers; (2) making the Route Selection Information convergence more slowly. In the simulation we have observe the relation between the communication overhead and convergence time. Fig.5 illustrates when T_1 is 5 seconds and 10 seconds how the number of SelectionNotice message and the convergence time of Route Selection Information change with t. In this experiment, we produce a new-route event

to AS1 in net29 and then log the SelectionNotice messages and the convergence time of Route Selection Information. In Fig.5, we can see when the value of t increases, the convergence time increases while the number of SelectionNotice message reduces. Furthermore, when t is small the convergence time increase slowly and the number of SelectionNotice message reduces evidently and when t exceeds specific value the convergence time increases rapidly and the number of SelectionNotice message reduces very few. From this relation we can get some optimized value of t for the network to make convergence time and communication overhead acceptable simultaneously.



(a) $T_1 = 5$ seconds (b) $T_1 = 10$ seconds

Fig. 5. Relationship between communication overhead of Route Selection Notice and the convergence time of Route Selection Information

6 Discussion

As an extension to BGP, Route Selection Notice supports packet validation, and at the same time is detached from packet validation. That is, to an AS, it is the administrator who determines whether and which type of packets validation is executed. Furthermore, some peers may be requested to send SelectionNotice messages while some do not and router may send Route Selection Notice messages to selected peers. So this extension is very flexible in configuration which may reflect the different relationship between ASes.

The packets validation does not need to be executed by all ASes as points out in [3]. In [3] the authors compare the effectiveness of DPF on with specific proportion ASes between selecting ASes randomly and to form a vertex cover, and find the latter excels the former. Finding a minimal vertex cover in a graph is NP-Complete. The AS vertex cover size of 1997 Internet is about 18.9% because of the power law property of Internet topology [12]. Using vertex cover to select validation ASes, even if 20% ASes in the Internet execute packet validation, above 80% forged packets can be picked out.

Selecting the "center" ASes with more peers to execute packets validation will leads to a smaller AS coverage ratio. So, it is certain to ask whether the

"centers" have incentive to deploy packets validation. If an ISP can provide more trustworthy service and protect servers and host in security, it is reasonable that more customer networks will prefer it. Attracting more customer network is the main incentive for large ISPs. For a stub network, if it does not support Route Selection Notice, some part of the internet may be unreachable, which is the incentive for such network.

The transmission and storage overhead of Route Selection Information which brings more burdens to border routers is proportional to the number of IP address prefix allocated. However, the next generation Internet adopts 128bit-long IPv6 address which has a tremendous address space. IPv6 prefix needed by an AS will be cut down, so the overhead will be much smaller.

7 Conclusion

In this paper, we bring forward to construct a inter-domain routing system which not only forwards packets properly to the destination but also validates the forwarded packets are the ones should be forwarded. We walk out the first step by extending BGP mechanism to support packets validation, which we name as Route Selection Notice. We design the extension according to several principles, such as no impairment to routing, efficiency, and promptness. Through simulation we prove the performance of Route Selection Notice is acceptable.

References

1. Rekhter, Y., Li, T.: A Border Gateway Protocol 4(BGP-4). RFC 1771 (1995).
2. Labovitz, C., Malan, G. R., Jahanian, F.: Internet routing instability. IEEE/ACM Transactions on Networking. vol. 6, no. 5, (1998) 515 – 527.
3. Park, K., Lee, H.: On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. Proceedings of ACM SIGCOMM. vol. 31, no. 4, (2001) 15 – 26.
4. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for IP traceback. Computer Communication Review. vol. 30, no. 4, (2000) 295 – 306.
5. Baker, F.: Requirements for IP Version 4 Routers. RFC 1812 (1995).
6. Ferguson, P., Senie, D.: Network Ingress Filtering:Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (1998).
7. Bremler-Barr, A., Levy, H.: Spoofing prevention method. Proceedings of IEEE INFOCOM. (2005) 536 – 547.
8. Li, J., Mirkovic, J., Wang, M., Reiher, M., Zhang, L.: SAVE: Source address validity enforcement protocol. Proceedings of IEEE INFOCOM. vol. 3 (2002) 1557 – 1566.
9. SSFNet project. `http://www.ssfnet.org/`.
10. Premore, B.: Multi-as topologies from BGP routing tables. `http://www.ssfnet.org/Exchange/gallery/asgraph/index.html`.
11. Labovitz, C., Ahuja, A., Bose, A., Jahanian, F.: Delayed Internet routing convergence. IEEE/ACM Transactions on Networking. vol. 9, no. 3, (2001) 293 – 306.
12. Siganos, G., Faloutsos, M., Faloutsos, P., Faloutsos, C.: Power laws and the AS-level Internet topology. IEEE/ACM Transactions on Networking. vol. 11, no. 4, (2003) 514 – 524.

Unicast and Multicast RWA Algorithms in DWDM-Based OVPN Backbone Networks

Jeong-Mi Kim[1], Jin-Ho Hwang[1], Jae-Il Jung[2], and Sung-Un Kim[1,⋆]

[1] Pukyong National University, 599-1 Daeyeon 3-Dong Nam-Gu, Busan, 608-737, Korea
{kimjm, jhhwang, kimsu}@pknu.ac.kr
[2] Hanyang University, 17 Haengdang-Dong Seongdong-Gu, Seoul, 133-791, Korea
jijung@hanyang.ac.kr

Abstract. OVPN (Optical Virtual Private Network) based on DWDM (Dense Wavelength Division Multiplexing) backbone framework has been regarded as a favorable approach for the future VPN. In DWDM-based OVPN, the RWA (Routing and Wavelength Assignment) problem has been an important and challenge issue for network performance improvements. In this paper, we propose new RWA algorithms in both cases of unicast and multicast approaches heading toward minimizing congestion by avoiding interference for requested connections. Also, we verify the performance of the proposed algorithms in terms of blocking probability and resource utilization.

1 Introduction

VPNs (Virtual Private Networks) are well-recognized as one of the critical applications of the future Internet market and have gained increased acceptance due to the economic benefits, scalability and reliability[1][2]. Given the increasing demand for high bandwidth services, OVPN using DWDM technology has been regarded as a favorable approach for the future VPN.

One of the critical issues in OVPN is the RWA (Routing and Wavelength Assignment) problem which is embossed as very important and plays a key role in improving the global efficiency for capacity utilization.

In previous unicast RWA, the routing scheme has been recognized as a more significant factor on the performance of the RWA problem than the wavelength assignment scheme[3][4]. And existing routing schemes that do not consider potential traffic demands can lead to serious network congestion by inefficiently utilizing wavelengths in terms of traffic-engineering.

In previous multicast RWA, some multicast routing algorithms[5][6] have some defects such as the long delay incurred in constructing the multicast tree. And also we need to have a simple procedure to add or delete a node from a session, because it changes the structure of the multicast tree.

⋆ Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 450–459, 2006.

To overcome these limitations, [7] proposed VS (virtual source)-based tree generation method. But as the number of VS nodes increases, the congestion due to the resources reserved for paths between VS nodes also increases[8].

So, we propose a new concept to minimize the congestion in critical links by choosing a wavelength route that does not interfere too much in accordance with potential future connection requests. And we apply this concept in both unicast and multicast.

The rest of the paper is organized as follows. Section 2 presents a generic architecture for DWDM-based OVPN. In section 3, we propose RWA algorithms in both cases of unicast and multicast, respectively. Section 4 verifies the performance of the proposed algorithms. Finally, section 5 concludes this paper.

2 Architecture of DWDM-Based OVPN for Unicast and Multicast RWA Algorithms

As shown in figure 1, a generic OVPN reference architecture is composed of VPNs in the electric control domain and the DWDM-based backbone network in the optical control domain. We assume that external VPNs aggregate IP packets (the same destined packets at the CE nodes (Client Edge)) to make operations simple. The internal OVPN backbone network consists of the PE nodes (Provider Edge) and the P core nodes (Provider).

As illustrated in figure 1, different VPNs may provide different services, i.e., point-to-point (unicast), point-to-multipoint (multicast). The congestion in a network is defined as the maximum offered traffic on any link. The congestion can be partially reduced by using an appropriate routing scheme with consideration of the current status of the network in the unicast manner and by constructing multicast tree efficiently in the multicast manner.

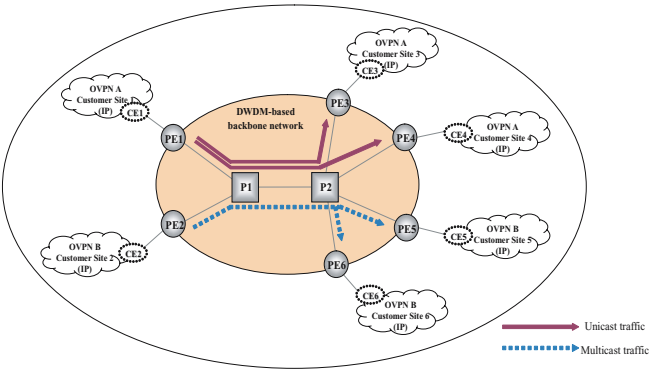


Fig. 1. OVPN reference architecture with unicast and multicast connectivities

In this paper, we propose a routing scheme that chooses a wavelength route minimizing interference for not only current traffics but also future potential traffics. Therefore, the objective is to improve blocking probability for requested connections and to decrease the congestion in OVPN backbone networks.

3 Unicast and Multicast RWA Algorithms

In this section, we propose new RWA algorithms for both unicast and multicast traffics during connection establishment procedure in OVPN backbone networks. In order to restrict the congested links and segments from the network situation, we consider the interference concept[9] in accordance with potential future connection requests.

To formulate unicast and multicast RWA algorithms, let the directed graph *G=(V,E)* represents the network with the n-element set of vertices V and the l-element set of directed edges $E=\{e_i | 1 \leq i \leq l\} \subseteq \{(u,v) | u,v \in V, (u \neq v)\}$. We assume k connection requests for potential demands in which the set of unicast connection requests (potential source-destination pairs in the future) is $Mu=\{m_i^u=(s_i,d_i) |\ 1 \leq i \leq k\}$. And the set of multicast connection requests is $Mm=\{m_i^m=(s_i,d_{ij}) |\ 1 \leq i \leq k,\ 1 \leq j \leq k\}$ where s_i and $d_i(d_{ij})$ are the source and the destination nodes, respectively. And we also assume that there are h connection requests for current demands. So, the set of current demands for both unicast and multicast connection requests is $Pu=\{p_i^u=(a_i,\ b_i) |\ 1 \leq i \leq h\}$ and $Pm=\{p_i^m=(a_i,\ b_{ij}) |\ 1 \leq i \leq h,\ 1 \leq j \leq h\}$, where a_i and $b_i(b_{ij})$ are the source and the destination nodes.

The objective function is given by

$$max \sum_{(s,d) \in M \setminus P} \alpha_{sd} \cdot F_{sd}. \tag{1}$$

Here α_{sd} is a generalized parameter representing link weight for the unicast traffic where $\forall(s,d) \in Mu$ (or segment weight for multicast traffic where $\forall(s,d_m) \in Mm$). And F_{sd} is the number of available wavelengths on the bottleneck link or path that has the smallest residual wavelengths. The equation (1) shows the maximum available wavelength problem, that is, the network accepts as many connections as possible and improves connection acceptance ratio.

For the wavelength assignment problem on the route selected by the proposed algorithms, we use the FF (First-Fit) scheme due to its small computational overhead and low complexity[10].

3.1 Unicast Routing Algorithm

In this sub-section, we propose an unicast routing algorithm which considers potential blocking possibilities for future traffic demands. This algorithm chooses a route that does minimize interference for potential future connection requests by avoiding congested links.

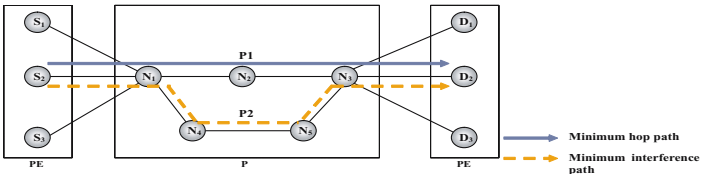


Fig. 2. Illustration of the proposed unicast routing algorithm

In most of routing algorithms, a rather generally used routing algorithm is the min-hop algorithm. This can lead to high blocking probability by inefficiently using the resources due to traffic concentration on the minimum-hop paths. As a simple example as shown in figure 2, there are three potential source-destination pairs such as (S_1, D_1), (S_2, D_2), and (S_3, D_3). The connection between the (S_2, D_2) pair is set along path P1 selected by min-hop routing as demanded. When the capacity of link (N_1-N_2-N_3) is not large enough, consequently this route may block the path between (S_1, D_1) as well as (S_3, D_3). Therefore, it is better to pick route P2 that has a minimum effect for other future connection requests, even though the path is longer than P1.

As a critical constraint factor, we define C^u_{sd} as the set of critical links which are shared with other node pairs at the same time. These links have higher congestion possibility than other links for potential future requests.

Additionally, the number of available wavelengths on a link is regarded as an important factor to improve network performance in terms of blocking probability. So, we add a new notation $\triangle^u$ as a threshold value of the available wavelengths on a link. Based on notations C^u_{sd} and $\triangle^u$, we define the congestion link as given in equation (2), where $\forall e \in E$, and $R(e)$ is the number of residual wavelengths on link e.

$$CL_{sd}\colon (e \in C^u_{sd}) \cap (R(e) < \triangle^u) \tag{2}$$

In this equation, the appropriate choice for threshold value $\triangle^u$ is very important for efficient wavelength utilization. If $\triangle^u$ is chosen to be large, then many pre-reserving wavelengths for future connection requests can cause wavelength waste. On the other hand, if $\triangle^u$ is set too small, then the potential blocking probability for upcoming traffic may be high. In this paper, we set the threshold value $\triangle^u$ within 30% of the total wavelength number on a link. This ratio is assumed by our simulation results regardless of the number of wavelengths per link.

To solve the problem described above, our unicast algorithm gives an appropriate weight to each link based on the amount of available wavelengths on a link e. The link weight is estimated by the following procedure. Firstly, let $\partial F_{sd}/\partial R(e)$ the variation of available wavelengths on the bottleneck link in accordance with the potential connection requests, and F_{sd} the set of available wavelengths of the critical links. With respect to the number of residual wavelengths of the link, the weight $Wu(e)$ of a link e is set to

$$Wu(e) = \sum_{(s,d) \in Mu \backslash Pu} \alpha_{sd}^{u}(\partial F_{sd}^{u}/\partial R(e)), \forall e \in E. \tag{3}$$

Equation (3) determines the weight of each link for all (s,d)-pairs in the set Mu (except the current request setting up between the (a,b)-pair), i.e., $(s,d) \in Mu \backslash Pu$. Using this formula we can calculate the sum of each link weight compared with other (s,d)-pairs. However computing weight for all links is very difficult in a wide area network environment. So, we define more restricted links than other links for routing by using following equation (4).

$$\begin{cases} \partial F_{sd}^{u}/\partial R(e)=1 [if (s,d): e \in CL_{sd}] \\ \partial F_{sd}^{u}/\partial R(e)=0 [otherwise] \end{cases} \tag{4}$$

$$Wu(e) = \sum_{(s,d): e \in CL_{sd}} \alpha_{sd}^{u}. \tag{5}$$

Consequently, computing link weight is simplified as shown in equation (5). Here if the value of α_{sd}=1 for all (s,d)-pairs, $Wu(e)$ designates the number of source-destination pairs containing critical link e. Once the weight of each link e is determined, traffics are routed between the (a,b)-pair along the path with the smallest $Wu(e)$ to achieve equation (1).

3.2 Multicast Routing Algorithm

In this sub-section, we propose a multicast routing algorithm based on VS-rooted approach that chooses minimum interference segment. The algorithm overcomes the limitation of VS-based method[7] and provides an efficient use of wavelengths.

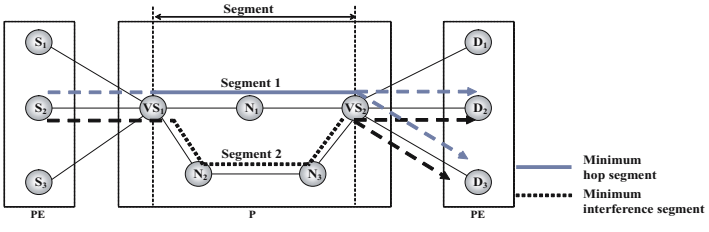


Fig. 3. Illustration of the proposed multicast routing algorithm

We define that a segment means a path between VS nodes. And each segment must follow the wavelength continuity constraint[4] because we assume VS nodes can only have a wavelength splitting and conversion capability. If there are three potential source-destination pairs such as (S_1, D_1 &D_2), (S_2, D_2 &D_3), and (S_3, D_1 &D_3). When the segment 1 is chosen for the first multicast session between (S_2, D_2 &D_3) pairs, other multicast sessions may share the same link

that can lead to high blocking probability due to the traffic concentration on the minimum-hop paths (VS_1-N_1-VS_2). If the connections for the (S_1, D_1 &D_2) and (S_3, D_1 &D_3) pairs are set along segment 1 (selected by min-hop routing as demanded), then this route may block the previous path when the capability of segment 1 is not large enough. Even though segment 2 is longer than segment 1, it is better to pick segment 2 that has a minimum effect for other future connection requests.

In the multicast routing algorithm, Equation (1) can be utilized to decide the minimum interference wavelength path between the VS-nodes. We apply the same concept of the unicast case (as given in equation (2)) to the multicast case (as shown in equation (6)). It determines the possible congested path in accordance with potential future connection requests between the VS nodes and calls it CP_{sd}.

$$CP_{sd}\colon (S^n_{sd}\colon e \in C^m_{sd}) \cap (R(S^n_{sd}) < \triangle^m) \tag{6}$$

where S^n_{sd}(the nth segment) represents the set of minimum hop segments between the VS nodes and C^m_{sd} indicates the set of critical links between the VS nodes, that is, C^m_{sd} are shared on the minimum hop paths of other VS node pairs at the same time. Also, we add a new notation $\triangle^m$ as a threshold value of the available wavelengths on S^n_{sd} (30% of the total wavelengths in S^n_{sd}). So, based on the notations and $\triangle^m$ and C^m_{sd}, we can define the congestion path as given in equation (6).

Equation (7) determines the weight of each segment for all (v_s, v_d)-pairs (where (v_s, v_d)-pair indicates a pair of VS nodes) in the set Mm except the current request setting up between the (a,b)-pair, i.e., (v_s, v_d) $\in Mm \backslash Pm$.

$$Wm(S^n_{sd}) = \sum_{\forall (v_s,v_d) \in Mm \backslash Pm} \alpha^m_{sd}(\partial F^m_{sd}/\partial R(S^n_{sd})) \tag{7}$$

where α^m_{sd} statistically represents the weight of each segment for the connection requests between the VS-nodes. Here F^m_{sd} is the number of available wavelengths on the bottleneck segment and $R(S^n_{sd})$ indicates the number of residual wavelengths on the segment S^n_{sd}.

$$\begin{cases} \partial F^m_{sd}/\partial R(S^n_{sd}) = 1 [if (v_s, v_d) : S^n_{sd} \in CP_{sd}] \\ \partial F^m_{sd}/\partial R(S^n_{sd}) = 0 [otherwise] \end{cases} \tag{8}$$

$$Wm(S^n_{ij}) = \sum_{(v_s,v_d):S^n_{sd} \in CP_{sd}} \alpha^m_{sd}. \tag{9}$$

Equation (8) allocates the differentiated values to the nth segment between VS nodes which were determined by the previous multicast session requests and corresponding available wavelengths. Same as the unicast case, calculating the weight of all segments is difficult, so we apply equation (8) to equation (7), and then computing segment weight is simplified as shown in equation (9). And if the value of α^m_{sd}=1 for all (v_s, v_d)-pairs, $Wm(S^n_{ij})$ represents the number

of source-destination pairs containing critical segment S_{sd}^n. Therefore, the algorithm decides a lightpath that has a minimum value of segment weight $Wm(S_{ij}^n)$.

3.3 Coordinated Unicast and Multicast Approaches

In figure 4, we illustrate an overall procedure of unicast and multicast approaches.

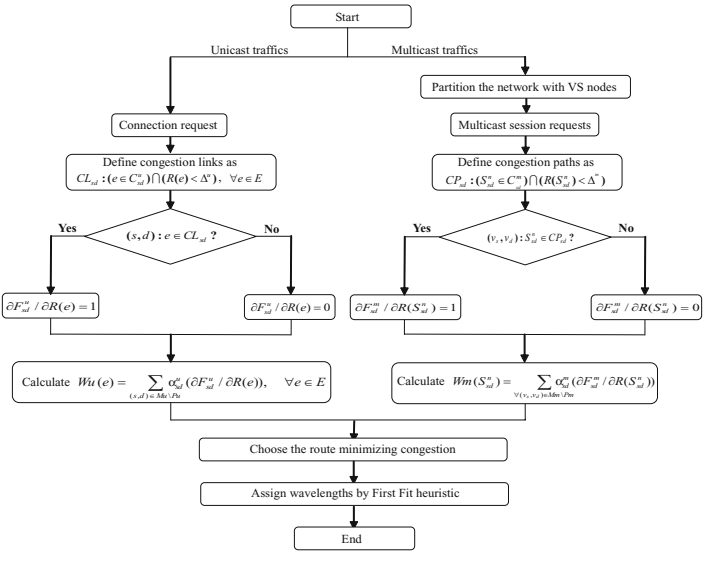


Fig. 4. Overall procedure of unicast and multicast RWA algorithms

4 Numerical Results

In order to evaluate the performance of the two proposed schemes in NSFnet environment, we assume that each link contains two uni-directional fibers (16 wavelengths), one in each direction and the traffic pattern is dynamic. And the connection requests arrive randomly according to the Poisson process, with negative exponentially distributed connection times with unit mean.

Firstly, we compare the proposed unicast RWA to the existing routing (DR (Dynamic Routing) and FR (Fixed Routing)) algorithms. The result is illustrated in figure 5. According to the simulation results, the proposed scheme has lower blocking probability than DR (improved by about 10%) due to selecting the minimum interference path which takes into consideration potential future setup requests.

The group size (GS) that determines the number of members to construct a multicast session is 0.2 (20%) and 0.3 (30%)[8]. Figure 5 reveals that the blocking probability of the proposed multicast scheme has better performance (improved

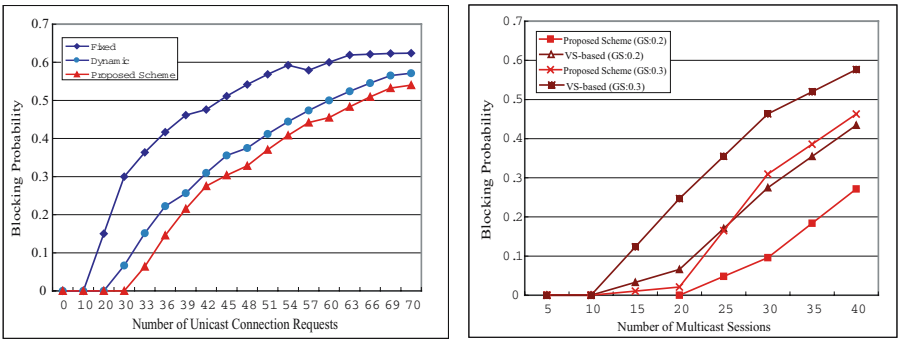


Fig. 5. Blocking probability of the proposed unicast RWA and multicast RWA in the uniform environment

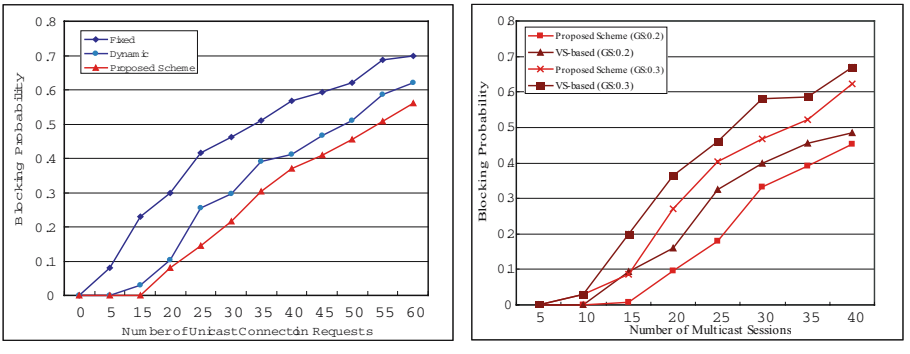


Fig. 6. Blocking probability of the proposed unicast RWA and multicast RWA in the non-uniform environment

about 10%-15%) than previous VS-based scheme in both cases of GS 0.2 and 0.3. We can find that the overall blocking probability is also increased in the non-uniform environment applying dynamic number of wavelengths on each link, as shown in figure 6. Therefore, we conclude that the proposed schemes outperform the previous methods in both case of unicast and multicast (improved about 5%-10%, 5%-15%, respectively) in the non-uniform environment.

Finally, we carried out a simulation for the network utilization of the proposed schemes. As shown in figure 7, in terms of the number of wavelengths the proposed unicast scheme has better performance (improved about 5%-20%) than Dynamic scheme. The loss ratio of the number of wavelength channels does not exceed 7%.

In figure 8, we find the improvement of the proposed multicast RWA (approximatly 25% and 26%) in terms of the gain of the number of wavelengths. And also we identify that the loss of the number of wavelength channels does not exceed 8% in both cases of GS 0.2 and 0.3.

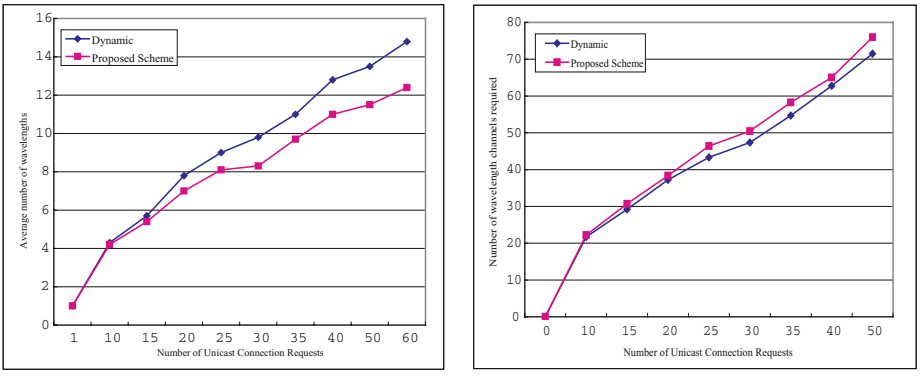


Fig. 7. The average number of wavelengths and the number of wavelength channels in the proposed unicast RWA

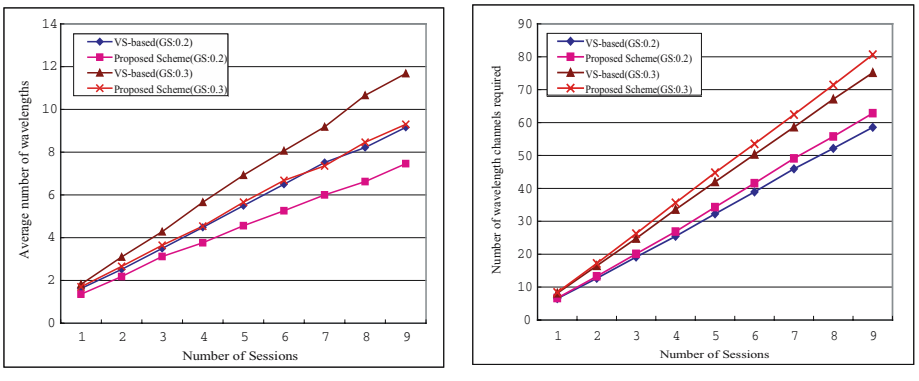


Fig. 8. The average number of wavelengths and the number of wavelength channels in the proposed multicast RWA

5 Conclusion

In this paper, we proposed new unicast and multicast RWA algorithms in DWDM-based OVPN backbone network. The objective of the proposed schemes is to choose a wavelength route that does minimize interference in accordance with potential future connection requests by avoiding congested links or paths. As a result of simulations, for the blocking probability, the proposed schemes (in both cases of unicast and multicast approaches) achieved better performance than the existing routing algorithms. Whereas we observed that the proposed RWA algorithms slightly need more numbers of wavelength channels due to the detour paths to avoid congestion links or paths. However, we experienced that the proposed schemes significantly improve the utilization of the number of wavelengths comparing with the previous methods.

Acknowledgment

This work was supported by grant No.(R01-2003-000-10526-0) from Korea Science & Engineering Foundation and the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF- 2005-213-D00060).

References

1. Stephen French, Dimitrios Pendarakis, Optical Virtual Private Networks: Applications, Functionality and Implementation, Photonic Network Communications, vol.7, no.3, pp. 277-238, 2004.
2. Mi-Ra Yoon, Ju-Dong Shin, Chang-Hyun Jeong, Jun-Mo Jo, Oh-Han Kang, Sung-Un Kim, Optical-LSP Establishment and a QoS Maintenance Scheme Based on Differentiated Optical QoS Classes in OVPNs, Photonic Network Communications, vol.7, no.2, pp.161-178, 2004.
3. H. Zang et al., A Review of Routing and Wavelength Assignment Approaches for Wavelength Routed Optical WDM Networks, Optical Networks Magazine, vol.1, no.1, pp.47-60, Jan. 2000.
4. S. Ramamurthy, B. Mukherjee, Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks, Proceedings of IEEE GLOBECOM 1998, vol.4, pp.2295-2302, Nov. 1998.
5. Xijun Zhang, Wei J.Y., Chunming Qiao, Constrained multicast routing in WDM networks with sparse light splitting, Journal of Lightwave Technology, vol.18, no.12, pp.1917-1927, Dec. 2000.
6. N. Sreenath, K.Krishna Mohan Reddy, G. Mohan, C.S.R. Murthy, Virtual source based multicase routing in WDM optical networks, Proceedings of IEEE International Conference on, pp.385-389, Sep.2000.
7. N. Sreenath, C. Siva Ram Murthy, G. Mohan, Virtual Source Based Muticast Routing in WDM Optical networks, Photonic Network Communications, vol.3, no.3, pp.213-226, July 2001.
8. Jong-Gyu Hwang, Jae-Il Jung, Yong-Jin Park, Jung-Hyun Bae, Hyun-Su Song, Sung-Un Kim, A RWA Algorithm for Differentiated Services with QoS Guarantees in the Next Generation Internet based on DWDM Networks, Photonic Network Communications, vol.8, no.3, pp. 319-334, Nov. 2004.
9. Figueiredo, G.B., da Fonseca, N.L.S., Monteiro, J.A.S., A minimum interference routing algorithm, 2004 IEEE International Conference on Communications, Volume 4, pp.1942-1947, June 2004.
10. X. D. Hu, T.P. Shuai, Xiaohua Jia, and M.H. Zhang, Multicast Routing Wavelength Assignment in WDM Networks with Limited Drop-offs, INFOCOM, IEEE Computer and Communications Societies, vol.1, no.7-11, pp.487-494, Mar. 2004.

QoS and Resource Management

Improving Delay Characteristics of Real-Time Flows by Adaptive Early Packet Discarding

Kazumi Kumazoe[1], Masato Tsuru[2], and Yuji Oie[2]

[1] Kitakyushu JGNII Research Center, NICT,
AIM-7F, 3-8-1 Asano, Kokurakita-ku, Kitakyushu-city, Fukuoka, 802-0001, Japan
kuma@kyushu.jgn2.jp
[2] Department of Computer Science and Electronics, Kyushu Institute of Technology,
680-4, Kawazu, Iizuka-city, Fukuoka, 820-8502, Japan
{tsuru, oie}@cse.kyutech.ac.jp

Abstract. The quality of real-time applications is significantly affected by the delay of packets traversing a network. Some real-time applications set limits for acceptable network delay, and thus, a packet delayed longer than this limit before arriving at its destination is not only worthless but also harmful to the quality of the application because it may increase the queuing delay of other packets. Therefore, we propose an adaptive scheme for real-time applications in which such packets are discarded early. In this scheme, packets experiencing too much delay are discarded at intermediate nodes based on the delay limit for the application and the delay experienced by each packet. Such early discarding of packets is expected to improve the overall delay characteristics of real-time flows competing for network resources shared only by those flows. Simulation results showed that our scheme is effective.

1 Introduction

The quality of real-time applications is significantly affected by delay and packet loss as packets traverse a network. In fact, some real-time applications set limits for acceptable network delay. For example, VoIP defines service classes based on an end-to-end packet delay limit for a flow in a network and the rate of packet loss [1]. In those applications, packets delayed longer than the acceptable limit are invalidated by their applications when they reach their destinations, even though they successfully arrive at the receiver. Such packets are useless for the applications, and thus they are an excess load in the network.

Therefore, we propose an adaptive early packet discarding scheme; in this scheme, those packets not contributing to the quality of real-time applications are discarded in advance at intermediate nodes. Note that we assume that we can predict a fixed amount of delay (e.g., propagation delay and transmission delay) for packets traversing between nodes before packets are sent, and thus, only the variable part of the delay (mainly, queuing delay) is taken into account in our scheme. For example, if the acceptable total network delay for an application is 50 [ms] and the fixed delay on an end-to-end path is 30[ms], the acceptable

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 463–472, 2006.

total queuing delay is 20[ms]. We also assume that real-time application flows are separated from other elastic traffic, such as TCP flows, in terms of the bandwidth (queuing buffer) shared by the flows. Thus, hereafter, we focus only on real-time flows in a network. We present two kinds of router-supported mechanisms in the early packet discarding scheme: one is called Maximum Transmission Queue Delay (MTQ) which resembles the concept of a Maximum Transmission Unit (MTU), and the other is Queue Delay To Live (QTL), which is analogous to the mechanism for Time To Live (TTL); both of these require the use of an additional header field in IP or UDP packets (e.g., an IPv6 optional header) to convey queuing delay information for each packet.

Early packet discarding, in which some packets might be dropped even though the queuing buffer is not full, is not a very new concept. For example, in Random Early Detection (RED) [2] or its variants, an intermediate node probabilistically discards incoming packets based on its queue length to mitigate instability and unfairness in the throughput of TCP flows traversing the node by introducing random losses instead of burst losses. In contrast, our scheme aims to reduce the delay of real-time application flows by reducing the number of worthless packets that wastefully consume network resources. In the context of TCP over ATM (more generally, the case that one long higher-level packet is fragmented into many short lower-level cells) in which if one cell is dropped then all remaining cells belonging to the same packet will be useless and harmful at the higher-level performance even though they are not yet dropped, Early Packet Discard (EPD) [3] was proposed to improve TCP performance against the fragmentation problem. Our scheme is similar to EPD in terms of the basic idea of discarding packets in advance if they are likely to be useless to the final application.

To reduce the delay of real-time flows, a variety of packet scheduling policies, such as Earliest Deadline First (EDF) [4], have been developed along with some resource reservation schemes to optimally reorder the sequence of packets belonging to various flows. On the other hand, our scheme, does not deal with such scheduling policies, but it can be combined with them. However, we note that complex packet scheduling schemes are not scalable in general, while our scheme is so light-weight that it is applicable to heavily loaded core routers. In fact, for MTQ and QTL schemes, the expected queuing delay of a packet in an intermediate node can be easily obtained from the queue length, which is readily managed at the node, when the packet arrives at the node. Then, the delay is simply compared with a MTQ/QTL value in the packet header or subtracted from the QTL value in the header. The cost of updating the QTL value (subtracting), which is similar to that of TTL mechanism, is negligible.

This paper is organized as follows. In Section 2, we introduce the mechanisms of our early packet discarding schemes (MTQ and QTL). The network simulation and traffic models are described in Section 3, and Section 4 presents simulation results that demonsrate the effectiveness of our scheme. Concluding remarks are presented in Section 5.

2 Adaptive Early Packet Discarding at Intermediate Nodes

MTQ and QTL schemes were originally proposed in a light-weight practical framework for the active network [5], and the effectiveness of using those techniques in network time synchronization was described [6]. We, on the other hand, adopted MTQ and QTL schemes to improve the delay characteristics of real-time application flows. Because the MTQ scheme limits the queuing delay at each node, when a packet arrives at a node, the expected queuing delay is calculated from the length of the output queue and the bandwidth of the output link; if the expected delay is longer than the MTQ value in the packet header, the packet is discarded. Therefore, setting the MTQ parameter is equivalent to limiting the size of the queuing buffer for each packet. Because the QTL scheme limits the total queuing delay along a path, when a packet arrives at a node, the expected queuing delay is calculated; the QTL value in the packet header is reduced by the calculated delay. If the updated QTL value is not positive, the packet is discarded. Thus, the initial value of the QTL parameter corresponds to the total queuing delay limit.

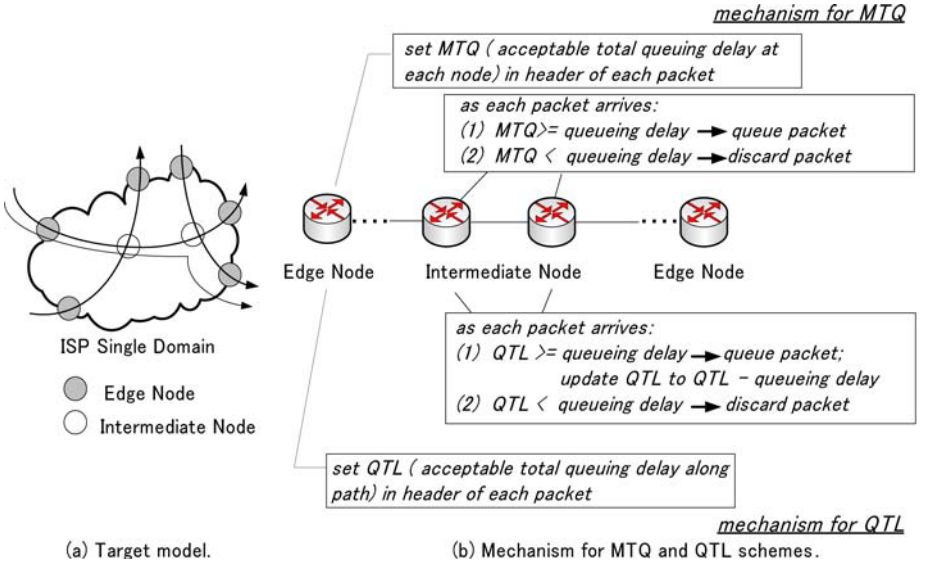


Fig. 1. Target network model and MTQ/QTL schemes on model

Our target model in which real-time application flows compete for resources in a single domain network is shown in Fig.1(a). At the ingress-edge nodes, MTQ and/or QTL parameters are set in the header of each packet incoming to the domain, and the packets are forwarded to the intermediate nodes. At the intermediate nodes, every time a packet arrives, it is queued or discarded according to the MTQ and/or QTL schemes shown in Fig.1(b).

3 Simulation Model

The simulation model we used is illustrated in Fig. 2. Three flow groups (0, 1, and 2) are in the network, and each flow group consists of 120 real-time application flows. Each node has a buffer of 200 [packets].

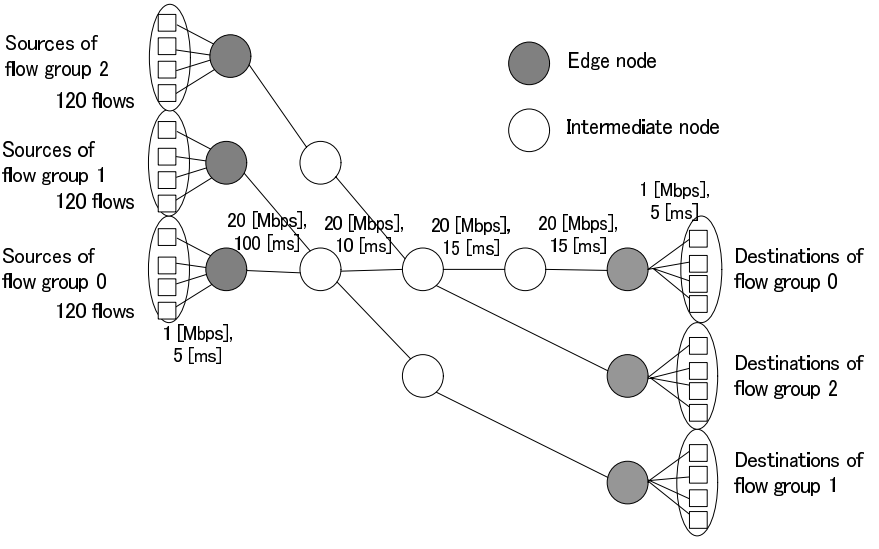


Fig. 2. Simulation model

As a traffic model, we applied an ON-OFF burst model that uses an exponential distribution of its ON and OFF periods [7]. We targeted two types of traffic: one with its average ON/OFF durations set to 350/650 [ms], and the other with its durations set to 3500/6500 [ms]. The results we observed were similar for the two types; therefore, in the following, we show the characteristics when the degree of traffic burstiness was lower, i.e., when the average ON/OFF duration was set to 350/650 [ms]. The average rate was fixed at 80 [kbps], and the packet length was 200 [bytes].

In the simulation model presented in Fig. 2, the maximum queuing delay for one packet at each node is 16 [ms] because at most 200 200 [byte] packets are waiting to be sent into a link with a bandwidth of 20 [Mbps]. The packets belonging to flow group 0 compete with the traffic of the other flow groups in the two nodes along their path; therefore, the total queuing delay increases to 32 [ms]. The maximum total queuing delay that packets of flow groups 1 and 2 experience along their paths will be 16 [ms]; their paths each have one node where they compete with the traffic of the other flow groups. Hence, in our model, we suppose three kinds of acceptable total queuing delay limits: 10 (strict), 20 (moderate), and 30 (loose) [ms], which may correspond to three different application delay constraints or may be derived from three different network environments that have different fixed delays. In our simulation results, we observed similar

tendencies when the acceptable queuing delays were 20 or 30 [ms]. Therefore, we will only show the results for 10 and 20 [ms] due to space limitations.

As a metric for evaluating performance, we adopted the effective packet loss rate, *Ploss*. *Ploss* is the sum of $Ploss_{net}$ and $Ploss_{apl}$, where $Ploss_{net}$ is the ratio of the number of packets discarded at the nodes in the network to the total number of packets, and $Ploss_{apl}$ is the ratio of the number of packets discarded by the application at the destination (because the queuing delay in the network exceeded the application's acceptable limit) to the total number of packets.

We ran the simulation using ns release 2.27 [8] with added MTQ and QTL mechanisms to manage queues.

4 Simulation Results

We first evaluated our scheme through network simulations in a homogeneous environment in which all flows had an identical delay limit (delay requirement). The resulting *Ploss*es in cases with MTQ only, QTL only, and QTL plus MTQ are reported in Sections 4.1, 4.2, and 4.3. In our simulation model (where each node has an output queuing buffer corresponding to a maximum queuing delay of 16 [ms] for the targeted application flows), if neither MTQ nor QTL are adopted, very high *Ploss*es are seen for acceptable total queuing delays of both 10 and 20 [ms] as shown in case (1) of Figs. 6(a) and (b). For example, $Ploss_{net}$ is 2% and $Ploss_{apl}$ is 37% for flow group 0 when the acceptable queuing delay is 10 [ms].

Note that a MTQ value of more than 16 [ms] is meaningless because the queuing delay at each node is at most that value, while a QTL value of more than 32 [ms] might be meaningless because the total queuing delay in traversing at most two congested nodes along a path is likely to be less than that value. In Section 4.4, we examine our scheme in a heterogeneous environment in which two types of flows (each type with its own delay limit) are multiplexed on each path.

4.1 Effectiveness of Adopting MTQ Scheme at Intermediate Nodes

Figure 3 (a) shows *Ploss* when the acceptable total queuing delay was 10 [ms]. The figure shows that adopting the MTQ scheme significantly improved the *Ploss*es of real-time application flows if the MTQ value was set within an appropriate range. However, setting a MTQ parameter (< 1.8[ms]) that was too small greatly increased $Ploss_{net}$; accordingly, a high *Ploss* was observed.

Conversely, when a value of more than 5[ms] was set, the effect of MTQ on flow group 0 was very limited. Furthermore, when a value of more than 10[ms] (the acceptable total queuing delay) was adopted, a very high $Ploss_{app}$ was observed for every flow group similarly to the case without MTQ.

Figure 3 (b) shows *Ploss* when the acceptable queuing delay was 20 [ms], which also indicates an appropriate range of MTQ values.

In general, a packet experiencing a large queuing delay at a congested node is likely to finally exceed the total queuing delay limit even if the delay experienced at that node does not exceed its limit. Thus, setting MTQ to an appropriate value (at least smaller than the total queuing delay limit of the application and smaller

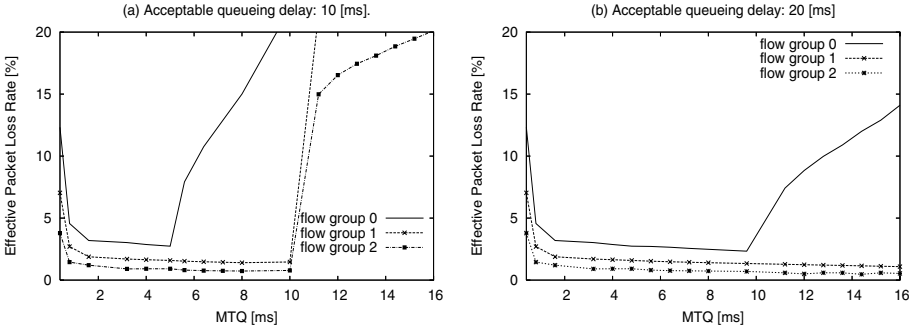


Fig. 3. *Ploss*(effective loss rate), adopting MTQ

than the maximum queuing delay at each node) may be effective for aggressively discarding in advance those packets that will probably exceed the limit before reaching their destinations.

However, the optimal value for MTQ at a node directly depends on how long of a queuing delay will be experienced by the packet in succeeding nodes; therefore, this value is hard to predict. Furthermore, because the value of MTQ set in a packet traversing nodes does not change node by node, the value might be ineffective when, for example, multiple congested nodes are in a path. Therefore, it seems that adopting only the MTQ scheme in a network is not always effective.

4.2 Effectiveness of Adopting QTL at Intermediate Nodes

The *Ploss*es resulting from adopting QTL for the acceptable total queuing delays of 10 and 20 [ms] are shown in Fig. 4 (a) and (b).

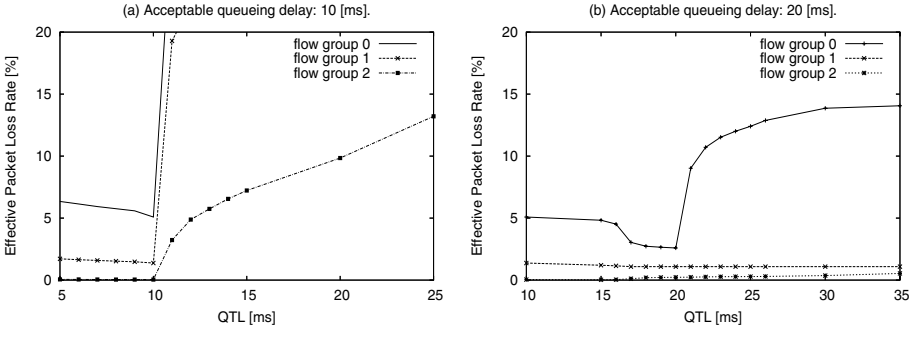


Fig. 4. *Ploss*(effective loss rate), adopting QTL

From Fig. 4 (a), one can easily see that *Ploss* improved most when QTL was set exactly to the acceptable total queuing delay for the application. Observing the results in other simulation scenarios including those shown in Fig. 4(b), we found that setting QTL exactly to the acceptable total queuing delay limit was

always optimal regardless of the value of the delay limit itself and the degree of traffic burstiness of the flows. The optimal setting of QTL suppress the *Ploss* of every flow (e.g., case (3) in Fig. 6) much more than no QTL setting (e.g., case (1) in Fig.6). This effectiveness must increase as the number of congested nodes increases. This simple rule for setting the value of QTL is of practical importance from the operational standpoint.

Note that setting QTL exactly to the acceptable total queuing delay limit conservatively discards packets that have exceeded the limit, and the above results indicate that setting QTL to some value smaller than the limit to aggressively discard packets in advance (which are likely to exceed the limit before reaching their destinations) is harmful, unlike the case of MTQ. Therefore, in the following subsection, we investigate the effects of setting QTL and MTQ simultaneously (QTL plus MTQ) on *Ploss* to exploit the combination of conservative discarding using QTL and aggressive discarding using MTQ.

4.3 Effectiveness of Setting MTQ and QTL Simultaneously

Figure 5 (a) shows *Ploss* when the acceptable total queuing delay is 10 [ms]. QTL is set to 10 [ms], equal to the delay requirement, and MTQ is set to the range from 0 to 16 [ms] (that is, QTL plus MTQ). As shown in Fig. 5(a), using QTL plus MTQ improved *Ploss* for a wider range of MTQ values than did setting only the optimal QTL. In the case in which the acceptable total queuing delay is 20 [ms], as shown in Fig. 5 (b), because the queuing buffer (=16[ms]) is originally shorter than the acceptable total queuing delay, the effectiveness of setting MTQ is not clear because such a small queuing buffer already exploits the effectiveness of the MTQ scheme when combined with the QTL scheme.

Combining QTL and MTQ, that is, setting QTL to the total queuing delay limit and setting MTQ to some value smaller than the acceptable limit (e.g., 75% of the limit), seemed to result in good delay characteristics in general.

To summarize the effectiveness of MTQ, QTL, and QTL plus MTQ, in Figs. 6 (a) and (b), we compare the *Ploss* that resulted from adopting (1) neither MTQ nor QTL, (2) the optimal MTQ (hard to predict), (3) the optimal QTL (equal to the acceptable total queuing delay), (4) the optimal QTL and the optimal

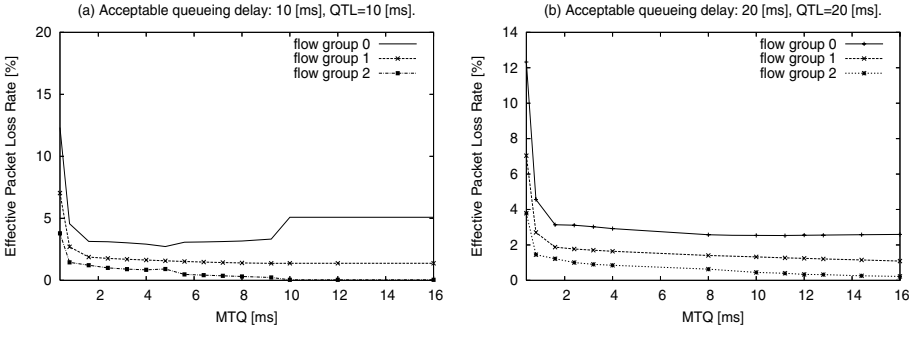


Fig. 5. *Ploss*(effective loss rate), setting both MTQ and QTL

MTQ, and (5) the optimal QTL and a moderate MTQ, when the acceptable total queuing delay is either 10 or 20 [ms]. Because the optimal MTQ is hard to predict in general and may not always exist , cases (2) and (4) are not realistic.

Obviously, in both cases, *Ploss* improved drastically by setting MTQ and/or QTL (cases (2) through (5) in the figure) compared with the case where neither MTQ nor QTL was set (case (1)). As shown in Fig. 6 (a), using QTL only (case (3)) was improved by adding MTQ, that is, case (5). When *Ploss* in case (3) is compared with that in case (5), the results of (5) are preferable because the balance of *Ploss* for all flow groups was improved, and the *Ploss* of the worst flow group was reduced. On the other hand, in Fig. 6 (b), QTL by itself (case (3)) performed nearly optimally.

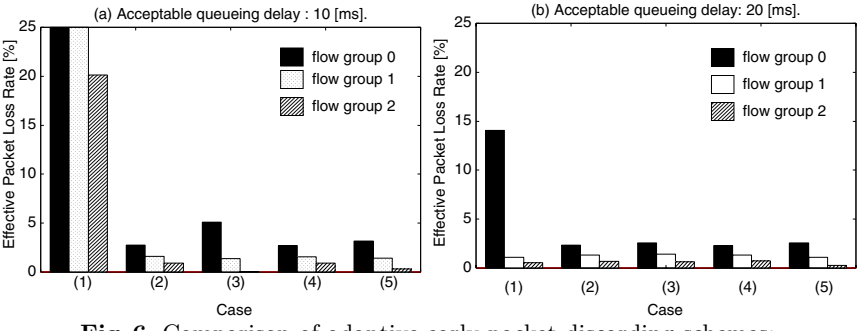


Fig. 6. Comparison of adaptive early packet discarding schemes: (1) no parameters set, (2) MTQ : 5 [ms], QTL : acceptable queuing delay [ms], (3) QTL : acceptable queuing delay [ms], (4) MTQ : 5 [ms], QTL : acceptable queuing delay [ms], (5) MTQ : 7.5 [ms], QTL : acceptable queuing delay [ms]

4.4 Effectiveness of MTQ/QTL Schemes in a Heterogeneous Environment

We investigated the effect of setting QTL/MTQ on *Ploss* when coexisting flows have different acceptable total delay requirements by using the following configuration. The flows in each flow group shown in Fig. 2 are divided into two subgroups, referred to as subgroups 1 and 2, each of which has 60 flows. As shown in Table 1, the flows of subgroup 1 have stricter delay requirements than those in subgroup 2.

Figure 7 shows *Ploss* for each subgroup, when parameters were set as in Table 2. Comparing the performance of (1) with those of other cases makes obvious that setting only MTQ, only QTL, or both MTQ and QTL improves *Ploss* just as in the homogeneous circumstance in which all flows had identical queuing delay requirements. In Figure 7, (2)–(4) show *Ploss* results from setting MTQ parameters. The MTQ parameters adopted in (2) were found in a trial and error manner suited to the delay condition of subgroup 1, and MTQ in (3) was set to a longer value than that set in (2). The MTQ adopted in (4) was found by trial and error based on the delay conditions of each subgroup. Cases (5)–(7) show results when QTL was adopted. Case (5) shows *Ploss* when QTL

Table 1. Acceptable queuing delay limit for each subgroup flow

flow group	subgroup	indication	acceptable queuing delay [ms]
0	1	f0-1	10
	2	f0-2	20
1	1	f1-1	10
	2	f1-2	20
2	1	f2-1	10
	2	f2-2	20

Table 2. Setting parameters

#	parameters [ms]	#	parameters [ms]
(1)	set no parameters	(6)	f0-1–f2-2: QTL=20
(2)	f0-1–f2-2 : MTQ=5	(7)	f0-1,f1-1,f2-1: QTL=10
(3)	f0-1–f2-2 : MTQ=7.5		f0-2,f1-2,f2-2: QTL=20
(4)	f0-1,f1-1,f2-1: MTQ=5	(8)	f0-1–f2-2: MTQ=7.5, QTL=10
	f0-2,f1-2,f2-2: MTQ=10	(9)	f0-1–f2-2: MTQ=7.5, QTL=20
(5)	f0-1–f2-2: QTL=10	(10)	f0-1–f2-2: MTQ=7.5, QTL=10/20

was set based on the delay condition of subgroup 1, and (6) shows that when QTL was set based on the delay conditions of subgroup 2. The QTLs adopted in (7) were based on the delay conditions of each subgroup. The results of cases (2)–(4) and (5)–(7) show that when either MTQ or QTL was adopted, setting the parameters for all flows based on the strictest queuing delay requirement of all flows was necessary for achieving small *Ploss*es of all flows.

Cases (8)–(10) show the *Ploss* setting for both MTQ and QTL parameters. In these cases, we supposed that the proper value for MTQ was unknown, so

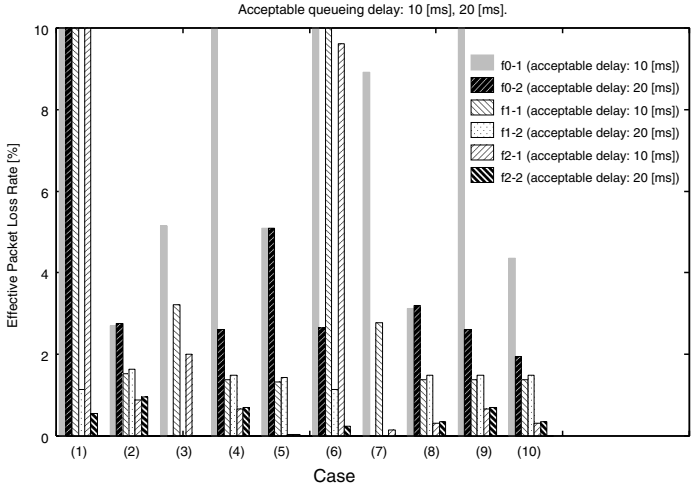


Fig. 7. *Ploss* when coexisting flows have different queuing delay requirements

we set MTQ to 75 % of the strictest acceptable queuing delay requirement in the network, i.e., 7.5 [ms]. Case (8) shows *Ploss* when QTL was set based on the delay requirement of subgroup 1, the QTL in (9) was set based on the delay requirement of subgroup 2, and (10) is a case of setting QTL based on the delay requirements of each subgroup. We found from the results for (8)–(10) that when both of QTL and MTQ were adopted, setting QTL of each flow to the strictest acceptable queuing delay requirement among flows is preferable; i.e., QTL should be set to 10 [ms] in this case.

5 Concluding Remarks

We have proposed an adaptive early packet discarding scheme (MTQ and QTL) for real-time application flows. In our scheme, a packet experiencing too much queuing delay is discarded at intermediate nodes based on a limit for the total queuing delay the packet experiences along the path (QTL) and/or a limit for the local queuing delay the packet experiences at each node (MTQ).

We evaluated our scheme through network simulations in a homogeneous environment in which all flows had an identical delay limit (delay requirement) and in a heterogenious environment in which two types of flows (each type had its own delay limit) were multiplexed on each path.

We found that using the QTL plus MTQ, that is, setting QTL to the total queuing delay limit and MTQ to some value smaller than the limit (e.g., 75% of the limit), resulted in relatively good delay characteristics, while it was practical.

This work was supported in part by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (A) (15200005).

References

1. Schmitter, A., Schwarzbacher, A.T., Smith, T.D.: Analysis of network conformity with voice over IP specifications. In: ISSC2003. (2003) 82–86
2. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking **1** (1993) 397–413
3. Romanow, A., Floyd, S.: Dynamics of TCP traffic over ATM networks. IEEE Journal on Selected Areas In Communications **13** (1995) 633–641
4. Georgladls, L., Guerin, R., Parekh, A.: Optimal multiplexing on a single link: Delay and buffer requirements. IEEE Transactions on Information Theory **43** (1997) 1518–1535
5. Tsuru, M., Kitaguchi, Y., Fukuoka, H., Oie, Y.: On the practical active network with the minimal functionality. In: The second International Workshop on Active Network Technologies and Applications. (2003) 45–52
6. Kitaguchi, Y., Machizawa, A., Tsuru, M., Oie, Y., Hakozaki, K.: The advanded network time synchronous system by self-discarding packet technique. Information Processing Society of Japan **46** (2005) 1017–1024
7. Fiorini, P.M.: Voice over ip (voip) for enterprise networks: Performance implications and solutions. In: International CMG Conference. (2000) 545–556
8. The Network Simulator ns-2,*http://www.isi.edu/nsnam/ns/*.

Voice Traffic Characterization Models in VoIP Transport Network

Ilyoung Chong[1], Chul-Woon Jang[1,⋆], and Hyun-Kook Kahng[2]

[1] Dept. of Information and Communications Eng., Hankuk Univ. of FS
Seoul , Korea
{iychong, jcw21}@hufs.ac.kr
[2] Dept. of Electronics Information Engineering, Korea University
Seoul , Korea
kahng@korea.ac.kr

Abstract. The motivation for characterization of VoIP traffic is that VoIP quality is mainly affected by some impairments in transport network in terms of delay, jitter and packet loss. It is shown at the paper that the prediction of transport network resource to satisfy the VoIP QoS is important to find an perceptual optimization of playout buffer, and is able to provide effecient way to compute resource consumption in VoIP transport network. This paper shows two models to caharcterize VoIP traffic in transport network, and proposes the novel mechanism to compute an amount of resource consumption. The mechanism evaluates its availability of VoIP call arrived newly in order to sustain a stable VoIP quality level. The applicability of the proposed mechanism in the paper will be stressed in terms of computational efficiency of dynamic real time computation algorithm.

1 Introduction

The VoIP quality is mainly affected by network impairments such as delay, jitter and packet loss. Playout buffer at the receiving side can be used to compensate for the effects of jitter based on a tradeoff between delay and loss. It is shown that the prediction of transport network resource to satisfy the VoIP QoS is important to find an perceptual optimization of playout buffer. The contributions of the paper are three-fold. First, we show a characteristic model of VoIP traffic in transport network in terms of buffer length, traffic intensity and VoIP packet loss. The model will also be used for the computation of VoIP traffic resources and for QoS estimation in terms of VoIP packet loss probability. Second, we characterize the upper bound of VoIP traffic characteristics and required network resources in the transport network. This estimation approach is more simple than using traditional approaches [1][7]. Third, we show that the

⋆ This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment). and this research was supported by Korea Science and Engineering Foundation(R01-2003-000-10562-0).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 473–482, 2006.

novel algorithm is proposed to approximate required bandwidth of VoIP traffic in transport network. It is considered as better characterization approach in reduction of computational cost. The paper analyzes few traffic characterization approaches among related works and their features. Let a group of D contiguous s1ots be ca11ed $frame$. The voice packet interva1 in a burst is then considered to be deterministic and is equal to D slots. If we consider a s1otted mode1 where the transmission time of a voice packet is taken as unit time of a s1ot, D contiguous s1ots, ca11ed a $frame$, are considered as servers of a transport resources in VoIP transport network. The arriva1 process in a frame is considered a $semi - Markov$ process. [5] has proposed the $SM/D/D$ model for frame scale model of multip1exer. In this model, voice packets are generated in interva1 of D s1ots within a burst , and the change of system status at every D-s1ot frame is observed. So, $SM/D/D$ can be considered as the approximation nlode1 of $SM/D/1$. The $SM/D/1$ mode1 has been analyzed by using intricate matrix analysis and numerica1 evaluations. [7] provided the exact and approximation model of $SM/D/1$. [5] has a1so analyzed the bursty voice packet traffic as $SM/D/D$ mode1 in frame sca1e. By using the resu1ts of the ana1ysis, the paper takes into account the overa11 picture of voice bursty traffic characteristics in frame sca1e for multiplexor in VoIP transport network.

2 Characterization of VoIP Bursty Traffic

2.1 Characteristic Elements of VoIP Bursty Traffic

Fig.1 (a) shows two different curves at each different traffic 1oad in VoIP transport network. The voice packet 1oss probabi1ity over VoIP transport network with small buffer capacity ($< D$) is mainly affected by the VoIP packet component fluctuation. If the transmission system in VoIP transport network has an avai1ab1e buffer capacity 1arger than D, the voice packet 1oss probabi1ity is varied with the average rate of arriving voice packets. That is, it is dominated by the 1ong-term behavior. In the VoIP packet traffic characteristics, a sharper slope occurs than in burst leve1 component since a short-term variation at the VoIP packet leve1 affects its VoIP packet 1oss probability significant1y. In Fig. 1, the most significant variation with increasing 1oad is the vertical trans1ation of the burst 1eve1 component. When the available buffer space in the VoIP transport network is 1ess than the value of D, the saturation probabi1ity to the finite buffer capacity is main1y varied by the voice packet level congestion property.

In the effect of burst level characteristics of VoIP packets, Fig.1 (b) shows that the s1ope of the burst leve1 component is proportiona1 to the mean burst 1ength whi1e the VoIP packet 1evel component remains unchanged. From Fig. 1 (b), the most significant fact can be found. The curve for burst 1ength with 240 PTT (Packet Transmission Time) is insensitive on the variation of buffer capacity in VoIP transport network resource. That is, the packet 1oss probabi1ity of long bursty traffic sources in VoIP transport network wil1 not be affected by the variation of buffer capacity un1ess an avai1able buffer capacity is larger than the burst length.

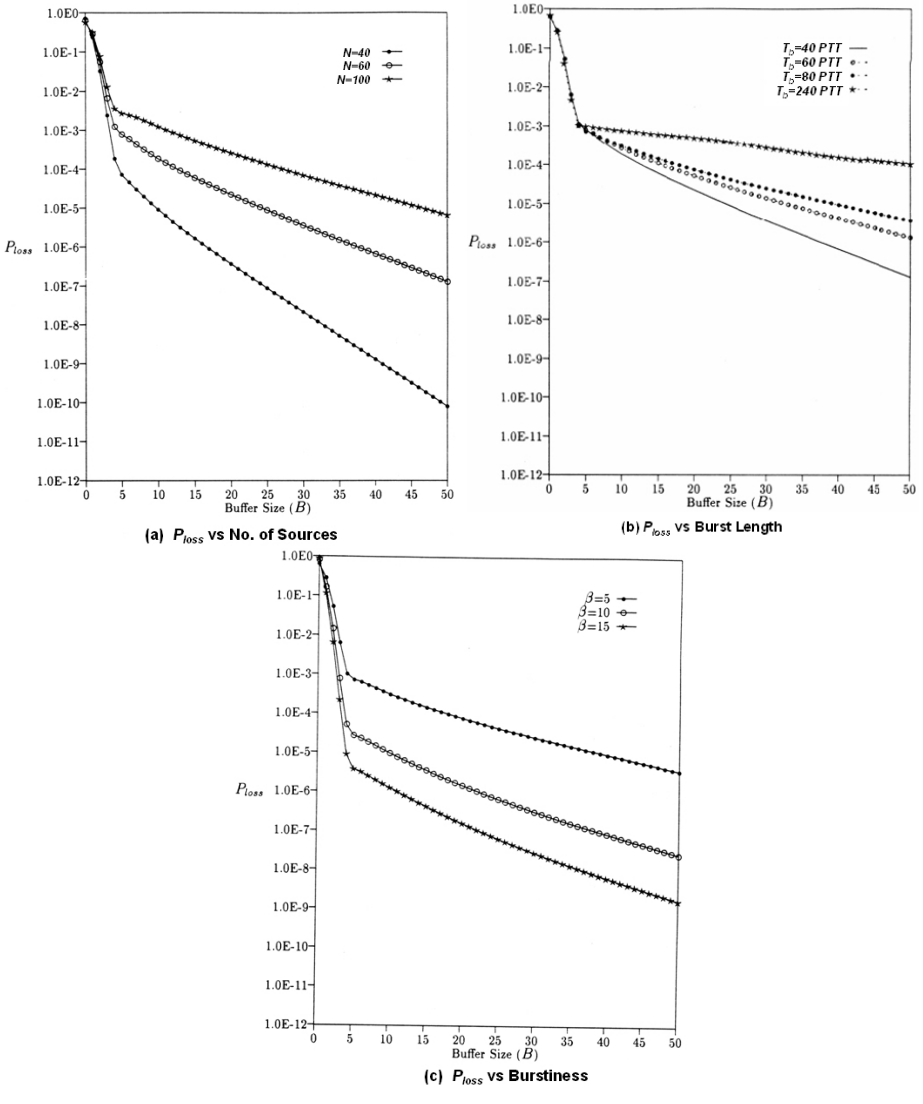


Fig. 1. Traffic Characteristics of VoIP Packet on Loss Probability(number of sources, burst length and burstiness)

And in the effect of burstiness to VoIP packet in transport network, Fig.1 (c) indicates that a traffic stream with different burstiness (β) and same burst 1ength a1so shows the vertical translation of the burstiness variation with increasing burstiness.

In the burst congestion consideration, it is more convenient to ignore the discrete nature of the VoIP packet arriva1 process in the burst period. Various mode1s have been proposed in the literature for analyzing queues with correlated

arrivals within burst time scale. Among them, the fluid-flow model is appealing since arriva1 rate fluctuation are accurately represented but the work to be accomp1ished by the server is assumed to arrive in a continuous flow rather than in discrete unit of packets. The simp1ifcation appears reasonab1e when the voice packet interarrival times are very sma11 compared to the time sca1e of changes in the arrival rate. [10] shows that the fluid-flow model may be viewed as a means to eva1uate a burst component, which constitutes by far the most significant part of the queue when there is a non-neg1igib1e probabi1ity that the instantaneous arrival rate exceeds the link capacity. [9],[10], and[12] show that the fluid-flow approximation is one approach for the analysis of burst level congestion.

We now turn to the mathematical nlode1 of the system proposed by [7]. Let $B(t)$ and $n(t)$ denote the buffer contents and the number of active sources, respectively, at time t. And let

$$P(t,x) = Pr\{n(t) = i, B(t) \leq x\}, 0 \leq N, t \geq 0, x \geq 0.$$

The partial differential equation for $P(t,x)$ is as follows:

$$(i - \frac{C}{P})\frac{\partial F(x)}{\partial x} = (N-i+1)\frac{\lambda}{\mu}F_{i-1}(x)\{(N-i)\frac{\lambda}{\mu} + i\}F + (i+1)F_{i+1}(x) \quad (1)$$

where, C and P are outlink capacity and peak rate respectively.

The equation(1) can be rewritten in matrix form as

$$D\frac{d}{dx}F(X) = MF(X), 0 < x < q \quad (2)$$

where D is a diagonal matrix whose elements contains the increasing or decreasing queue length rates, M is the transition matrix of the active sources, and q is the maximum buffer size. The solution of (2) is of the form

$$F(X) = F(\infty) + \sum_{k=0}^{N-[\frac{C}{P}]-1} exp(z_k x)a_k\phi_k, 0 < x < q \quad (3)$$

where, the a_k are coefficients that must be found by defining and solving suitable boundary equations. z_k is an eigenvalue of the matrix $\mathrm{D}^{-1}\mathrm{M}$. F($\infty$)is an ($N$+1) vector in which i^{th} component is the stochastic equilibrium probability that i sources are active. ϕ is the eigenvector of $\mathrm{D}^{-1}\mathrm{M}$ corresponding to z. The N - [$\frac{C}{P}$] eigenvalues, z, such that Re(z) < 0 are called *stable* eigenvalues. The complementary buffer occupancy distribution, $Q(x) = P_r B > x$, is then given by $Q(x)$ = 1 - e′F(X), where e′ is [1,1,$\cdots$,1].

$$Q(x) = -\sum_{k=0}^{N-[\frac{C}{P}]-1} exp(z_k x)a_k e'\phi_k, 0 < x < q \quad (4)$$

where it is used the fact that e′F(∞) = 1 since the process must be in some state. The details to compute the buffer overflow probability are summarized at (6).

Fig. 2 (a) shows VoIP packet loss probability at burst time-scale varying buffer size. In the observation of an effect of burst length. Fig. 2 (b) shows that bursty traffic sources with long burst length are less sensitive to buffer size than with short burst length. In particular, Fig. 2 (b) indicates that a long-term behavior of VoIP bursty traffic sources can be characterized by the burst level analysis.

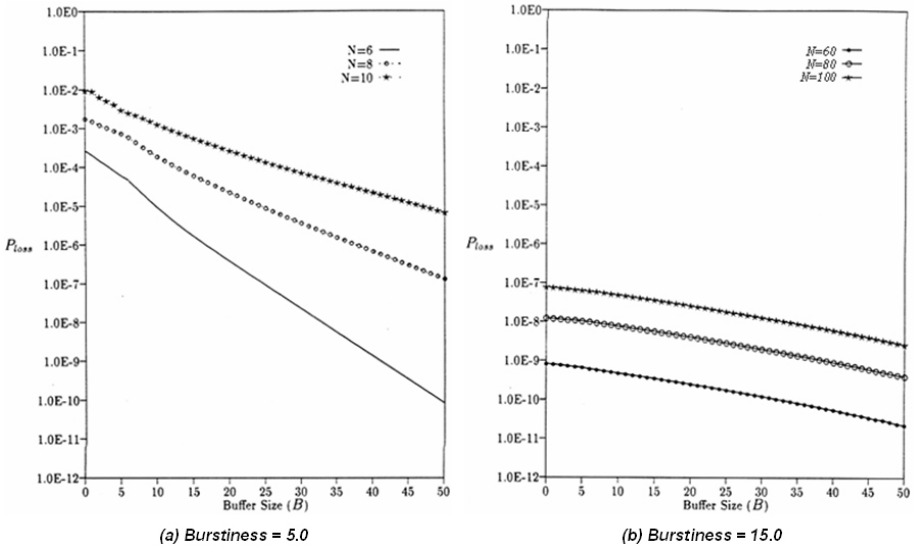


Fig. 2. Upper Bound Characteristics of VoIP Packet Loss Probability

2.2 Upper-Bound Characteristics of VoIP Packet Traffic

We characterize traffic source by the following parameter: P: peak bit rate, A: mean bit rate, $L[VoIPpacket] : meanburstlength.The$burstiness(b) is defined as the peak to mean rate ratio ($b = P/A$). The average burst duration (T) relates to the peak rate (P) and VoIP packet length (l_{VoIP}) through the following equation:

$$T = \frac{L \cdot l_{VoIP}}{P}$$

It is assumed that both active and silence periods are exponentially distributed with average T and $T(b-1)$ respectively.

The equilibrium queue distribution is described by a set of differential equations together with a set boundary equation describing the queue behavior as its limit. In the activity model of active source, the number of active sources k is modeled by a continuous-time birth-death process where the transition states are given by

$$p(k, k+1) = (N-k) \cdot \lambda \; for \; 0 \leq k < N$$
$$p(k, k-1) = k \cdot \mu \; for \; 0 \leq k < N$$

where, the silence length(in number of VoIP packets) is $1/\lambda = (L \cdot (b-1) \cdot l_{VoIP})/P$, the active length is $1/\mu = (L \cdot l_cell)/P$ and N is the number of input traffic sources. The equilibrium probability of k sources being active P_k given by the binomial distribution:

$$P_k = \binom{n}{k} \left(\frac{\lambda}{\lambda+\mu}\right)^k \left(\frac{\mu}{\lambda+\mu}\right)^{N-k} \tag{5}$$

The equilibrium probability that k sources are active and the queue length does not exceed x, represented by $F_k(x)$, is obtained by solving the matrix equation of (2). From the equation (3) the loss probability can be derived,

$$P_{loss} = \frac{1}{\alpha \cdot N} \sum_{k=w}^{N} (k-w) u_k \tag{6}$$

where, $\alpha = 1/b, w = C/P = [w]$, and $u_k = P_k - F_k(q^-)$. The VoIP packet loss which occurs when the queue is held at its limits, expressed as a fraction of the total VoIP packets, can be determined from u_k. $F(q^-)$ is the probability that k sources are active and the queue is full. And the eigenvectors and eigenvalues of (3) are obtained by the use of numerical methods[1], while coefficients a_j's are obtained by solving the following set of boundary conditions:

$$F_k(0) = \sum_{j=0}^{N} a_j \phi_k = 0, \ for\ w < k \leq N \tag{7}$$

$$F_k(q^-) = p_k, \ for\ 0 \leq k < w \tag{8}$$

(6) can be rewritten as:

$$P_{loss} = \frac{P}{A \cdot N} \sum_{k=[C/P]}^{N} \left\{ (k - \frac{P}{C}(p_k - F_k(q^-)) \right\} \tag{9}$$

In the case with identical bursty traffic sources, the general upper bound on the buffer overflow probability, $Q(x)$, in the burst level approach the burst scale term of the $logicalbufferless$ model as opposed to a queuing model with finite buffer. For the upper bound it is sufficient to calculate the binomial tail distribution. Under assumption of $ergodicity$ the VoIP packet loss probability of the equation (9) and boundary condition of (7), the upper bound of VoIP packet loss probability can be obtained as shown below.

$$P_{loss} \leq \frac{P}{A \cdot N} \sum_{k=[W/P]}^{N} (k - \frac{W}{P}) p_k$$

$$= \frac{P}{A \cdot N} \sum_{k=[W/P]}^{N} \left[(k - \frac{W}{P}) \binom{N}{K} \left(\frac{\lambda}{\lambda+\mu}\right)^k \left(\frac{\mu}{\lambda+\mu}\right)^{N-k} \right] \tag{10}$$

And[13] shows that analytical result under a bufferless model based on the binomial distribution provides the reasonable queuing result to find the upper bound

of the VoIP packet loss probability in the stationary queuing behavior. It will be used for the upper bound in the computation of VoIP packet loss probabilities.

3 Proposed Characterization Mechanism Through Loss Period Computation of VoIP Traffic

3.1 Loss Period Characterization of VoIP Traffic

This subsection describes a discrete-time Markov BD process model and proposes a novel concept derived from a conventional first passage time [1], which is a time duration that state i reaches to state j (i j) in Markov BD process. The computed result is applied to decide the admission control in terms of VoIP packet loss probability in the transport network. For the computation algorithm of expected occupied resource amount of connections, this paper uses a discrete-time Markov BD process model and proposes a novel concept derived from a conventional first passage time [1], which is a time duration that state i reaches to state j (i j) in Markov BD process.

The proposed passage time in the paper is conditioned by initial state, and its transition path is diversified rather than a conventional first passage time. The novel concept, which is a special case of the first passage time, the first up-passage time (FUT) and the first down-passage time(FDT), is introduced in [4]. Using the concepts (FUT and FDT), as shown in [4], the novel concepts to find the computation algorithm of VoIP packet loss probability in the transport network.

Firstly, the loss period to the virtual capacity of overlay network is computed from the results, FUT and FDT and U_i. The expected offered load period ($\overline{W}$)is computed by using the heap occurrence probability (U_i) and the property of probabilistic similarity in expected path length computation.

$$E[W] = \sum_{i=1}^{N} \bar{W}_i \cdot U_i = \sum_{i=1}^{N} \left(\bar{W}_{i,up} + \bar{W}_{i,down}\right) \cdot U_i$$
$$= \sum_{i=1}^{N} E[FUT_{1,i}^{(1)}] \cdot U_i + \sum_{i=1}^{N} E[FUT_{i,0}^{(N)}] \cdot U_i \qquad (11)$$

Define E[W_{up}] and E[W_{down}] an expected upward-path length and an expected downward-path length for all heaps, respectively. The upward-path length indicates the total path length of when the number of active sources takes journey from state 0 to the top state k. There may be fluctuations going down and up, and the total length of upward-path is a summation of those paths. The downward-path length is also the total path length of while the number of active sources reach to top state from state 0. First E[W_{up}] is rewritten by using the concept and some formula in [4] as:

$$E[W_{up}] = \sum_{i=1}^{N-1} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_{i+1} = \sum_{i=1}^{N-1} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{q_j} \right) \cdot q_k \cdot \bar{S}_k \right] \cdot U_{i+1} \quad (12)$$

From (2), we can see that E[W_{up}] is a function of S_k(k=1,2,...,N-1). So, we can rewrite (2) by $A_k(S_k)$ as:

$$E[W_{up}] = A_1(\bar{S}_1) + A_2(\bar{S}_2) + A_3(\bar{S}_3) + ... + A_{N-1}(\bar{S}_{N-1}) \tag{13}$$

where,

$$A_k(\bar{S}_k) = \sum_{i=1}^{N-1} X_{k,(1,i)} \cdot U_{k+1}, X_{k,(1,i)} = \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot \frac{1}{q_k} \cdot \bar{S}_k$$

In the computation of E[W_{down}],the similar procedure with E[W_{up}] will be applied.

$$E[W_{down}] = \sum_{i=1}^{N} \bar{W}_{i,down} \cdot U_i = \sum_{i=1}^{N} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{p_l}{q_l} \right) \cdot p_j \cdot \bar{S}_j \right] \cdot U_i \tag{14}$$

As we see(4), E[W_{down}] is also a function of $\bar{S}_k$ (k=1,2,...,N-1), (4) is rewritten by $A_k(\bar{S}_k)$, which is the summation of $Y_{k,(i,1)}$.

$$E[W_{down}] = B_1(\bar{S}_1) + B_2(\bar{S}_2) + B_3(\bar{S}_3) + ... + B_N(\bar{S}_N) \tag{15}$$

where,

$$B_1(\bar{S}_1) = \sum_{i=1}^{N} Y_{1,(i,1)} \cdot U_i, , B_2(\bar{S}_2) = \sum_{i=2}^{N} Y_{2,(i,1)} \cdot U_i, B_k(\bar{S}_k) = \sum_{i=k}^{N} Y_{k,(i,1)} \cdot U_k$$

$$B_N(\bar{S}_N) = Y_{N,(N,1)} \cdot U_N \quad and\ Y_{k,(i,1)} = \sum_{j=1}^{k} \left(\prod_{l=j}^{k} \frac{p_l}{q_l} \right) \cdot \frac{1}{p_k} \cdot \bar{S}_k \tag{16}$$

As shown so far, the expected offered load period ($\bar{W}$) is expressed by the expected path length ($\bar{S}_i$) of each state. The expected offered packets during load period can be computed by product of arriving packets at each state during one frame period (D). And It will be noted that the expected *Loss Period* ($E[T_{loss}]$) at this model is simply calculated without any further computation labor. That is, from (6), the overflow period is counted. The expected *Loss Period* ($E[T_{loss}]$) is as follows:

$$\begin{aligned} E[T_{loss}] &= \sum_{i=D+1}^{N-1} A_i(\bar{S}_i) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \\ &= \sum_{i=D+1}^{N-1} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_{i+1} + \sum_{i=1}^{N} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_i \\ &= \sum_{i=D+1}^{N-1} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot q_k \cdot \bar{S}_k \right] \cdot U_{i+1} \\ &+ \prod_{i=D+1}^{N} \left[\sum_{k=l}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot q_j \cdot \bar{S}_j \right] \cdot U_i \end{aligned} \tag{17}$$

And the expected offered packets during load period can be computed by product of arriving packets at each state during one frame period (D). Let $\bar{N}_{offered}$be the expected offered packets during offered load period.

$$\bar{N}_{offered} = \sum_{i=1}^{N-1} A_i(\bar{S}_i) \cdot i + \sum_{i=1}^{N} B_i(\bar{S}_i) \cdot i \tag{18}$$

The mean number of lost packets during loss period ($E[T_{loss}]$) is computed from (7). Let $\bar{N}_{lost}$ the expected lost packets during loss period ($E[T_{loss}]$).

$$\bar{N}_{lost} = \sum_{i=D+1}^{N-1} A_i(\bar{S}_i) \cdot (i - D) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \cdot (i - D) \tag{19}$$

Finally, we can obtain the PLR (Packet Loss Ratio of VoiP Traffic) from the results of (18) and (19).

$$\begin{aligned} PLR &= \frac{lost\_packets}{offered\_packets} = \frac{\bar{N}_{lost}}{\bar{N}_{offered}} \\ &= \frac{\sum_{i=D+1}^{N-1} A_i(\bar{S}_i) \cdot (i - D) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \cdot (i - D)}{\sum_{i=1}^{N-1} A_i(\bar{S}_i) \cdot i + \sum_{i=1}^{N} B_i(\bar{S}_i) \cdot i} \end{aligned} \tag{20}$$

3.2 Applicability of Proposed Mechanism

As anticipated VoIP traffic demand and network availability estimates are forecasts, they should be treated as such by the traffic management and service layer functions. Actual offered traffic will fluctuate around the forecast values in the long term. The algorithm introduced in the paper is fluid-flow approximation algorithm to make a decision of incoming new call request on VoIP transport network. Many approximation algorithms for bursty traffic sources have been proposed, but they take limited applications in real world due to complexity in computation or due to too much approximation. As shown in the paper, the proposed fluid-flow approximation algorithm provides a less computational cost for real-time application than the existing approximation approach. In our proposed approximation mechansm, $A_{N-1}(\bar{S}_{N-1})$ and $B_k(\bar{S}_k)$are used as function of λ, μ and N. The values of $A_{N-1}(\bar{S}_{N-1})$ and $B_k(\bar{S}_k)$ can be computed and tabulated according to input traffic parameters in advance. The tabulation is constructed as a function of a number of sources. The approximation algorithm can make a table considering few adjacent possible situations in advance, and it reduces the computation cost (e.g., $O(n)$) to estimate and reserve system resource for newly incoming traffic. It should be more useful approximate for network resource computation with VoIP traffic with similar pattern. If we consider the numerical results, the algorithm shows that results is very likely to fluid-flow approach in [3].

4 Conclusion

Currently VoIP is emerging to deply for public services with QoS guarantee, and it is stressed that our proposed mechanism will provide characterization of VoIP traffic in transport network well. These are summarized as follows:

- The proposed mechanism and model on VoIP traffic characterization makes a clear separation to the real time computation of physical characteristics in transport network through dynamic computation algorithm.
- The proposed model to charaterize VoIP traffic will enable to make the measurement in transport network resources with real-time, and the management of VoIP traffic resources will be performed in advance, and its feature will make an expectation of the resource consumption in VoIP transport network accordingly.
- The paper has proposed the approximated algorithm to compute resource capacity and to control admission for incoming VoIP calls. The algorithm computes a virtual capacity with dynamic, and its algorithm is shown to provide scalability in resource management for large scale VoIP transport network.

References

1. Danel P. Heyman and Matthew J. Sobel, "Stochastic Models in Operations Research - Volume I," McGraw Hill Book Company pp 38-104, 1988
2. Ilyoung Chong, "Cost Minimization Allocation (CMA) Algorithm", Technical Report, Univ. of Massachusetts, 1992.
3. Ilyoung Chong, "Traffic Control at Burst Level", Ph.D. Thesis, Univ. of Massachusetts, 1992
4. D. Scott Alexander, William A. Arbaught, Angelos D. Keromytis, Steve Muir, and Jonathan M. Smith, "Secure Qu laity of Service Handling: SOS", IEEE Communications Magazine 2000
5. B Kim, "Analytical Appoarch in Discrete–Time System," McGraw Hill, 1992
6. Panos Trimintzios, Geoge Pavou et al, "Service-Driven Traffic Engineering for Intradomain Qulaity of Service management," IEEE Network, May/June 2003
7. D. Anick , D. Mitra and M. M. Sondhi, " Stochastic Theory of a Data-Handling Systems with Multiple Sources," Bell Systems Technical Journal vol 61. No. 8 pp 1971-1894, Oct, 1982
8. S P Miller and C B Cotner, "The Quality Challenge Now and For the Future", ICDSC, May 1995
9. Fumio Ishizaki and Testuya Takine, "Cell Loss probability Approximation and Their Application to call Admission Control," Advances in Performance Analysis Vol. 2 No. 3, pp225-258, 1999
10. Roger C. F. Tucker, "Accurate Method for Analysis of a Packet Speech Multiplexor with Limited Delay," IEEE Trans. On Communications, vol 36, No. 4, April 1988
11. Hans Kroner, "Statistical Multiplexing of Sporadic Sources - Exact and Approximate Performance Analysis, " Proceedings of 13th ITC '91, pp723-729, 1991
12. Roch Guerin, Hamid Ahmadi and Mahmoud Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High Speed Networks," IBM Research Report RC16317, 1990

On Flow Distribution over Multiple Paths Based on Traffic Characteristics

Yoshinori Kitatsuji[1], Satoshi Katsuno[2], Masato Tsuru[3], Tetsuya Takine[4], and Yuji Oie[3]

[1] National Institute of Information and Communications Technology, Kitakyushu-shi, Fukuoka, 802-0001, Japan
kitaji@kyushu.jgn2.jp
[2] KDDI R&D Laboratories, Inc., Fujimino-shi, Saitama, 356-8502, Japan
katsuno@kddilabs.jp
[3] Kyushu Institute of Technology, Iizuka-shi, Fukuoka, 820-8502, Japan
{tsuru@ndrc, oie@cse}.kyutech.ac.jp
[4] Osaka University, Suita-shi, Osaka, 565-0871, Japan
takine@comm.eng.osaka-u.ac.jp

Abstract. In traffic engineering to effectively distribute traffic flows over multiple network paths, taking into account traffic characteristics for individual flows is vital in appropriately assigning flows to network paths for achieving better delay performance of the total traffic. All flows assigned to a path, in which some flows are highly bursty, equally experience a significant queuing delay when the path is highly utilized. We analyze the mean, variance, and 99.5^{th} percentile of queuing delay when two types of flows are multiplexed into a single path using an infinite buffer model with on-off state fluid flows. This is carried out to find a better traffic distribution strategy over multiple paths so as to lower queuing delay incurred to multiplexed flows with distinct traffic characteristics. From the analytical results, we found that trivial strategies, such as dividing individual types of flows in a proportional manner to the bandwidth of each path, or segregating distinct types of flows as much as possible, do not always achieve a good delay performance, e.g., in terms of max-min fairness. Thus, the flows should be distributed considering their traffic characteristics, the number of flows, and the path bandwidths.

1 Introduction

Internet service providers are facing the challenge of dynamically and adaptively designing their networks to satisfy customers' demands for fast, reliable and differentiated services at the minimal cost. Internet traffic engineering (TE) [1] is a key tool for accomplishing the goal by effectively mapping traffic demands onto the network topology and adaptively reconfiguring the mapping according to the change of network conditions. More specifically, TE adaptively distributes traffic flows over networks using an effective network resource allocation.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 483–492, 2006.

The fundamental philosophy to make the Internet scalable is that algorithmically complex processing should be pushed to the edge of the network whenever possible. Designing a backbone network supporting TE with two levels of components – a high-speed core network and edge routers surrounding the core – reflects that philosophy. Consider that the individual pairs of edge routers establish multiple paths to convey traffic from one edge (ingress) router to the other edge (egress) router. Effective traffic distribution can be achieved by balancing traffic flows between a pair of edge routers among several paths [2] [3].

In distributing traffic flow, however, an inadequate traffic aggregation (multiplexing) onto a path results in degrading delay performance significantly, i.e., a large averaged queuing delay and/or a large delay variation, especially when the path is highly utilized [4][5][6]. Specifically, assigning flows based on their average path utilization without considering their burstiness may incur the dire performance degradation, when highly bursty flows are included. In this situation, non-bursty flows multiplexed in the same path also encounter such a delay performance drop. Thus, given a set of flows having distinct traffic characteristics, and multiple paths having different bandwidths, distributing those flows over paths in order to achieve a good overall delay performance of flows is not trivial.

In this paper, therefore, we consider the basic case where flows having two types of traffic characteristics are assigned to two paths. We analyze the mean, variance, and 99.5^{th} percentile of the queuing delay when distinct types of flows are multiplexed into a single path using an infinite buffer model with on-off state fluid flows. Using such an analytical method, we exhaustively examine the delay performance of possible flow assignments to two paths, and find an optimal assignment to minimize the greater queuing delay in two paths to achieve max-min fairness in terms of delay performance of these paths.

A flow that we discuss in this paper can be defined as a set of packets distinguished by protocol header fields, such as a pair of IP addresses, a protocol number, and port numbers. However, we do not limit outselves to only this definition. Our suggestion is applicable to flows that are grouped into traffic classes distinguished by their traffic characteristics.

The remainder of this paper is organized as follows. In Section 2, we derive the mean, variance, and 99.5^{th} percentile of the queuing delay for an infinite buffer model accommodating two types of on-off state fluid flows. In Section 3, we analytically examine the queuing delay on all possible combinations in dividing flows into paths to find better delay performance. In addition, we compare the flow distribution combinations minimizing the mean, variance, or 99.5^{th} percentile of the queuing delay. Finally, we conclude this work in Section 4.

2 Analytical Model

We assume that the core of a high-speed network accommodates paths established between pairs of edge routers and guarantees the bandwidth of individual paths. The ingress router balances two types of flows forwarded to an egress

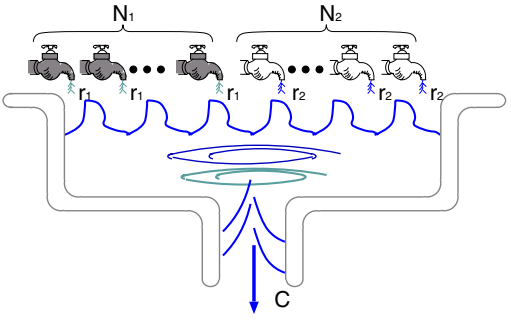


Fig. 1. Infinite buffer accommodating two types of on-off fluid flow

router over two paths to minimize the queuing delay encountered by the traffic in a buffer belonging to each path. In addition, the traffic consists of UDP flows and its characteristics are stable, even though the queuing delay fluctuates. Moreover, delay fluctuation encountered by traffic traversing the core network is very small compared to the queuing delay incurred in ingress routers.

First, we derive the mean, variance, and 99.5^{th} percentile of queuing delay from the fluid flow model. Next, we derive the metrics representing the delay performance for two paths.

2.1 Fluid Flow Model

The fluid model was first proposed by Anick *et al.* to model data network traffic [7]. In the fluid simulation paradigm, traffic is modeled in terms of a (time) discrete or continuous-rate-based model, rather than discrete packet instances. A fluid simulator technique keeps track of the fluid rate changes at traffic sources and network nodes.

We illustrate the model in which an infinite buffer receives two types of on-off state fluid flows, as shown in Figure 1. N_k denotes the number of fluid flows of class k ($k = 1, 2$) and C denotes the fluid output rate. The individual fluid flow of class k comes down at an input rate r_k when it is in the on state, and repeats the on and off state independently. The on and off terms are exponentially distributed where the means are $1/\mu_k^{(on)}$ and $1/\mu_k^{(off)}$, respectively. The traffic characteristics of UDP flows mentioned above are represented by the mean of on and off state periods, input rate, and number of fluid flows. The mean rate, ρ_k, of a fluid flow in class k and mean rate ρ of the total flows in both classes are represented as $\rho_k = r_k p_k^{(on)}$, and $\rho = \sum_{k=1}^{2} N_k \rho_k$, respectively, where $p_k^{(on)}$ is the steady-state probability in the on state of a class k fluid flow. Here, we assume $\rho < C$, and $\sum_{k=1}^{2} N_k r_k > C$. The first assumption makes the fluid flow model stable, and the second ensures the existence of queue length.

Let $(i, j) \in \mathcal{S}$ denote the number of on-state fluid flows of classes 1 and 2, $U(t)$ denote the fluid level in the buffer at time t, and $c_{(i,j)} = ir_1 + jr_2 - C$

denote the rate of variance of $U(t)$. To simplify the calculations, we assume that any $c_{(i,j)}$ is not 0. We define

$$\mathcal{S}_+ = \{(i,j);\ (i,j) \in \mathcal{S},\ c_{(i,j)} > 0\} \text{ and}$$
$$\mathcal{S}_- = \{(i,j);\ (i,j) \in \mathcal{S},\ c_{(i,j)} < 0\}\ ,$$

where $\mathcal{M}_+, \mathcal{M}_-$, and $\mathcal{M}$ denote the number of states of $\mathcal{S}_+, \mathcal{S}_-$, and $\mathcal{S}$, respectively. Furthermore, the first $\mathcal{M}_+$ states among $\mathcal{S}$ (numbered from 1 to $\mathcal{M}_+$), are included in $\mathcal{S}_+$ and the remaining $\mathcal{M}_-$ states (numbered from $\mathcal{M}_+ + 1$ to $\mathcal{M}$), are included in $\mathcal{S}_-$.

This fluid flow model is an infinite buffer model modulated by the continuous-time Markov chain with $\mathcal{M}$ states, so the transition rate matrix $\mathbf{T}$ and the steady-state probability vector $\boldsymbol{\pi} = (\boldsymbol{\pi}_+, \boldsymbol{\pi}_-) = (\pi_1, \pi_2, \cdots, \pi_{\mathcal{M}})$ hold

$$\boldsymbol{\pi}\mathbf{T} = \mathbf{O} \text{ and } \boldsymbol{\pi}\mathbf{e} = 1\ ,$$

where $\mathbf{e}$ is a vertical vector in which all the elements are 1.

To derive the probability distribution $\Pr[\tilde{U} > x]$, and n-th moment $E[\tilde{U}^n]$ for the fluid level that the fluid arrivals see and that is denoted by the steady-state random variable $\tilde{U}$, we define matrices $\mathbf{C}$, $\boldsymbol{\Pi}$, $\tilde{\mathbf{T}}$, and alignment of submatrices including $\mathbf{T}$, as follows

$$\mathbf{T} = \begin{matrix} \\ \mathcal{M}_+ \\ \mathcal{M}_- \end{matrix} \begin{matrix} \begin{matrix} \mathcal{M}_+ & \mathcal{M}_- \end{matrix} \\ \begin{pmatrix} \mathbf{T}_{+,+} & \mathbf{T}_{+,-} \\ \mathbf{T}_{-,+} & \mathbf{T}_{-,-} \end{pmatrix} \end{matrix}, \mathbf{C} = \begin{matrix} \\ \mathcal{M}_+ \\ \mathcal{M}_- \end{matrix} \begin{matrix} \begin{matrix} \mathcal{M}_+ & \mathcal{M}_- \end{matrix} \\ \begin{pmatrix} \mathbf{C}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \end{pmatrix} \end{matrix}, \boldsymbol{\Pi} = \begin{matrix} \\ \mathcal{M}_+ \\ \mathcal{M}_- \end{matrix} \begin{matrix} \begin{matrix} \mathcal{M}_+ & \mathcal{M}_- \end{matrix} \\ \begin{pmatrix} \boldsymbol{\Pi}_+ & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\Pi}_- \end{pmatrix} \end{matrix},$$

$$\text{and } \tilde{\mathbf{T}} = \begin{matrix} \\ \mathcal{M}_+ \\ \mathcal{M}_- \end{matrix} \begin{matrix} \begin{matrix} \mathcal{M}_+ & \mathcal{M}_- \end{matrix} \\ \begin{pmatrix} \tilde{\mathbf{T}}_{+,+} & \tilde{\mathbf{T}}_{+,-} \\ \tilde{\mathbf{T}}_{-,+} & \tilde{\mathbf{T}}_{-,-} \end{pmatrix} \end{matrix} = \boldsymbol{\Pi}^{-1}\mathbf{T}^T\boldsymbol{\Pi},$$

where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. The probability distribution and n-th moment for the fluid level are derived, as follows [8],

$$\Pr[\tilde{U} > x] = \frac{\boldsymbol{\alpha}\exp(x\mathbf{V})\mathbf{r}_+}{\pi\mathbf{r}} \text{ and} \tag{1}$$

$$E[\tilde{U}^n] = \frac{n!\boldsymbol{\alpha}\{(-\mathbf{V})^{-1}\}^n\mathbf{r}_+}{\pi\mathbf{r}}\ , \tag{2}$$

respectively, where $\mathbf{V}$ is a $\mathcal{M}_+ \times \mathcal{M}_+$ matrix in which all the diagonal elements are negative and nondiagonal elements are positive, $\boldsymbol{\alpha} = \boldsymbol{\pi}_+ + \boldsymbol{\pi}_-\mathbf{W}$, and $\mathbf{r} = (\mathbf{r}_+, \mathbf{r}_-) = (r_1, r_2, \cdots, r_{\mathcal{M}})$ is a vertical vector in which an element is the total input rate of each state m ($m \in \{1, 2, \ldots, \mathcal{M}\}$). $\mathbf{V}$ and $\mathbf{W}$ satisfy the following

$$\mathbf{V} = \mathbf{C}_+^{-1}\tilde{\mathbf{T}}_{+,+} + \mathbf{C}_+^{-1}\tilde{\mathbf{T}}_{+,-}\int_0^\infty \exp\{y(-\mathbf{C}_-)^{-1}\tilde{\mathbf{T}}_{-,-}\}(-\mathbf{C}_-)^{-1}\tilde{\mathbf{T}}_{-,+}\exp(y\mathbf{V})dy$$

$$\mathbf{W} = \int_0^\infty \exp\{y(-\mathbf{C}_-)^{-1}\tilde{\mathbf{T}}_{-,-}\}(-\mathbf{C}_-)^{-1}\tilde{\mathbf{T}}_{-,+}\exp(y\mathbf{V})dy\ .$$

Table 1. Path combinations over which traffic flows are balanced

Path combination	1	2
Path 1	1000 Kbit/s	700 Kbit/s
Path 2	1000 Kbit/s	1300 Kbit/s

Table 2. Traffic characteristics and combinations of two traffic classes

Traffic combination	1		2	
Traffic class	H	L-1	H	L-2
Average period of on-state [ms]	1	1	1	1
Average period of off-state [ms]	3	5	3	3
Probability of on-state	1/4	1/6	1/4	1/4
Rate of on-state [Kbit/s]	256	16	256	16
Average rate of a flow [Kbit/s]	64	2.67	64	4

2.2 Delay Metrics Minimized in Balancing Flows over Paths

We treat the mean, variance, and 99.5^{th} percentile of the queuing delay as delay performance for a set of flows assigned to a path. In flow distribution, we pay attention to the path having the worse delay performance, i.e., the larger mean of queuing delays in two paths. Hereafter, we term such a higher value of mean, variance, and 99.5^{th} percentile of the queuing delay in two paths the *mean delay*, *delay variance*, and 99.5^{th} *percentile delay*, respectively.

The *mean delay*, A, *delay variance*, D, and 99.5^{th} *percentile delay*, M, are respectively represented as

$$A = \max_l \frac{E[\tilde{U}_l]}{C_l}, \quad D = \max_l \frac{E[\tilde{U}_l^2] - E[\tilde{U}_l]^2}{{C_l}^2}, \text{ and } M = \max_l \frac{x_l}{C_l},$$

where $\tilde{U}_l$ denotes the steady-state random variable of queue length of path l $(= 1, 2)$, C_l denotes the bandwidth of path l, and queue length x_l of path l satisfies $\Pr[\tilde{U}_l > x_l] = 1 - 0.995$.

3 Numerical Analysis

We analyze the *mean delay*, *delay variance*, and 99.5^{th} *percentile delay* of every combination of divided flows assigned to two paths with respect to the various numbers of flows and the different combinations of path bandwidths. In the analysis we used two combinations for the path as shown in Table 1. Combination 1 and 2 consist of the same and different bandwidths of paths, respectively. We also used two traffic combinations, as shown in Table 2. The traffic combinations consist of the same rate flows at 256 Kbit/s in the on state for high-rate flows, labeled 'H', and at 16 Kbit/s for low-rate flows, while the mean of the off state

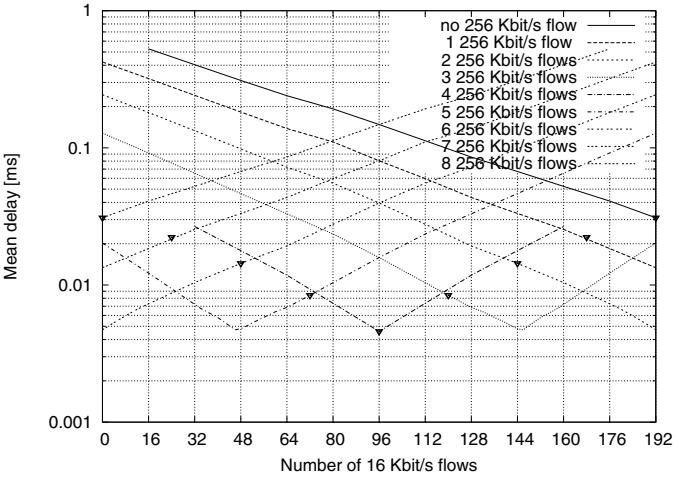


Fig. 2. *Mean delay* for traffic combination 1 distributed over two 1-Mbit/s paths

period for low-rate flows differs by 5 ms for combination 1, labeled 'L-1' and by 3 ms for combination 2, labeled 'L-2'. In both traffic combinations, the high-rate flows have the same traffic parameters.

3.1 Mean of Queuing Delay in Distributing Flows over Same-Bandwidth Paths

First, with same-bandwidth paths (path combination 1), we analyzed the *mean delay* of a sufficient number of flows. The numbers of high-rate and low-rate flows for traffic combination 1 (H and L-1) are 8 and 192, respectively, and, those for traffic combination 2 (H and L-2) are 8 and 128, respectively. The average rate for each traffic class is 512 Kbit/s and ρ is 1 Mbit/s in total. The *mean delay* for traffic combination 1 is shown in Figure 2. The legend and X-axis indicate the number of high-rate and low-rate flows, respectively, that are assigned to the path 1, i.e., "2 256 Kbit/s flows" at 32 on the X-axis indicates that 2 high-rate flows and 32 low-rate flows are assigned to path 1 and the remainings to path2.

From the figure, there are multiple flow assignments making the *mean delay* low. The assignment equally dividing flows in both traffic classes also makes the *mean delay* low. The $\bigtriangledown$ marks in the figure indicate combinations of the number of flows that lead to equal utilization between paths. $\bigtriangledown$ points incur the greater *mean delay*, as the difference in the number of low-rate flows between paths becomes greater. Shifting the number of low-rate flows from such distributions to make the utilization excessively unbalanced between paths makes the delay lower. These tendencies: the existence of multiple low-mean-delay combinations and shifting low-rate flows from equal utilization combinations, were also found in traffic combination 2.

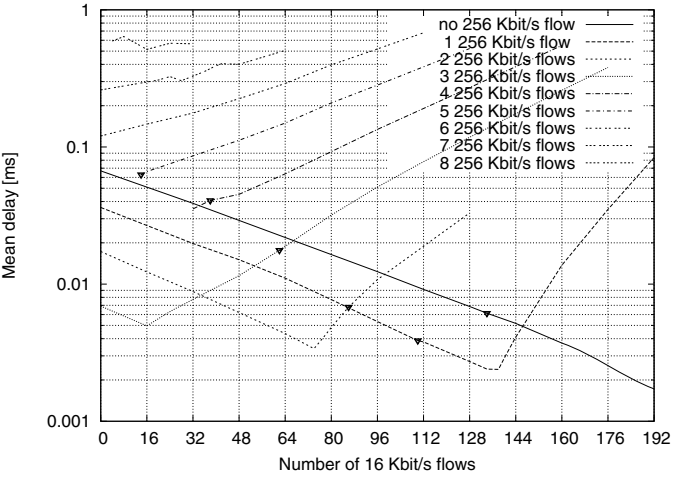


Fig. 3. *Mean delay* for traffic combination 1 distributed over 700-Kbit/s and 1300-Kbit/s paths

In terms of network operation, the equal flow division in both classes is much simple and, therefore, desirable. On the other hand, the existence of multiple flow assignments lowering the delay is also promising to apply the flow assignment in the case where either or both traffic classes cannot be equally divided. For example, flows in a class require a short one-way delay, and only one path can satisfy the requirement.

3.2 Mean of Queuing Delay in Distributing Flows over Different-Bandwidth Paths

We analyzed the *mean delay* in distributing flows over different-bandwidth paths, path combination 2. The number of flows is the same as that of path combination 1, where the average rate for each traffic class is 512 Kbit/s. The *mean delay* is shown in Figure 3. The legend and X-axis indicate the number of flows assigned to path 1, the narrow-band path.

The figure indicates that the flow assignment equalizing the utilization between two paths cannot minimize the *mean delay*. The flow assignment making the utilization between two paths unbalanced excessively by shifting low-rate flows from an equal-utilization division ($\bigtriangledown$) is as effective as path combination 1 for lowering the *mean delay*. These tendencies were also found in traffic combination 2. The assignment: all the low-rate flows are in the narrow-band path, and all the high-rate flows are in the broadband path, minimizes the *mean delay* in Figure 3. However, Such a complete separation of traffic classes could not minimize the *mean delay* in the case of traffic combination 2. With respect to

Table 3. Number of flows classified based on peak rate

Traffic class	Number of flows		
	Small	Middle	Large
H	2	4	8
L-1	40	64	192
L-2	40	64	128

Table 4. Flow assignment minimizing the *mean delay* for each number of flows for individual traffic combinations

Number of flows		How to divide	
H	L-1 or L-2	Assignment of H and L-1	Assignment of H and L-2
Large	Large	(broad, narrow)	(broad, both)
Large	Middle	(both, both)	(both, both)
Middle	Large	(both, both)	(both, both)
Middle	Middle	(both, both)	(both, both)
Large	Small	(both, narrow)	(both, both)
Small	Large	(narrow, both)	(narrow, both)

traffic combination 2, The best flow assignment was: 125 low-rate flows were in the narrow-band path, and remainings were in the broadband path.

To find features such that flow assignment minimizes the *mean delay* over the different-bandwidth paths, we analyzed the other combinations of the number of flows with path combination 2. This was carried out by classifying the peak rates, $r_k N_{(k,l)}$ of aggregated flows, where $N_{(k,l)}$ denotes the number of flows of traffic class k assigned to path l. The labels, 'small,' 'middle,' and 'large' in Table 3 represent whether the peak rate of an individual traffic class is less than the narrow-band path, between the narrow-band and broadband paths, or more than the broadband path, respectively. In all combinations of the number of flows from H and L-1, or H and L-2, there are six combinations incurring the queuing delay in each traffic class combination. Instances of flow assignment minimizing the *mean delay* for those combinations are presented in Table 4. The labels 'narrow', 'broad', and 'both' expressed as (A, B) in the columns indicate that all the flows are assigned to the narrow-band path, the broadband path, and divided into both paths, respectively, with the high-rate flows denoted by A and the low-rate flows denoted by B.

The table indicates that all the high-rate flows should be assigned to the narrow-band path in the case where the peak rate of all the high-rate flows is less than that of the narrow-band path and that of low-rate flows is larger than that of the broadband path. Although, with respect to other combinations, both or one traffic classes should be divided into both paths, we found that the strategy to compel the utilization to be unbalanced between two paths, in the manner that the path having the greater number of low-rate flows had the higher utilization, was capable fo the minimizing the *mean delay*.

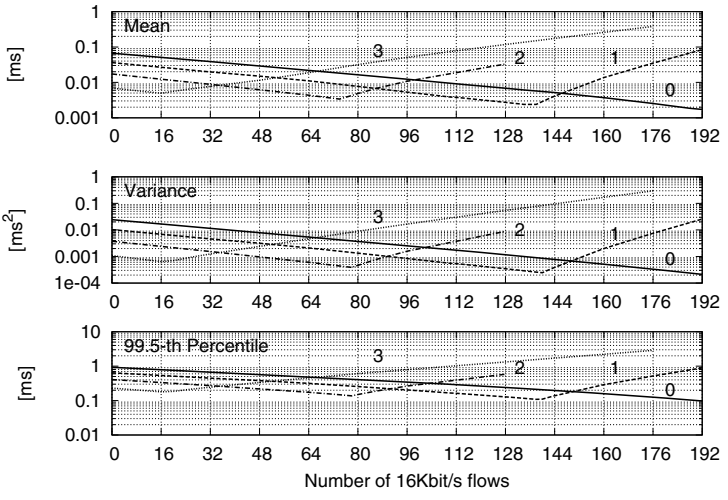


Fig. 4. Comparison of *mean delay*, *delay variance*, and 99.5^{th} *percentile delay* on traffic combination 1 distributed over 700 Kbit/s and 1300 Kbit/s paths

It is future work to devise a more concrete method to specify the number of flows to be divided based on the bandwidth and utilization of paths, and the traffic characteristics of flows.

3.3 Comparison of Flow Assignments Based on Mean, Variance, and 99.5^{th} Percentile of Queuing Delay

To demonstrate the distinct difference achieved by flow assignments lowering delay metrics, we compare flow assignments minimizing *mean delay*, *delay variance*, and 99.5^{th} *percentile delay* for flow combination 1 with same-bandwidth and different-bandwidth paths. The numbers of high-rate and low-rate flows are 8 and 192, respectively, so the average rate of both traffic classes is 512 Kbit/s. In the case of same-bandwidth paths (path combination 1), equal flow division of both traffic classes can also minimize the *delay variance* and 99.5^{th} *percentile delay* (the figure is omitted). In addition, other flow assignments approximately minimizing the *mean delay* also result in lowering the *delay variance*, and 99.5^{th} *percentile delay*.

However, in the case of different-bandwidth paths (path combination 2), the numbers of flows minimizing the *mean delay*, *delay variance*, and 99.5^{th} *percentile delay* differ, although the difference is small, as shown in Figure 4. The number near the lines represents the number of high-rate flows. The number of low-rate flows for both the *delay variance* and 99.5^{th} *percentile* slightly differs from the *mean delay*. Although we also analyzed on different combinations of the number of flows, we could not find any other remarkable behavior than this small difference among the delay metrics.

Consequently, flow assignments based on the *mean delay*, *delay variance*, or 99.5^{th} *percentile delay* can approximately minimize other metrics.

4 Conclusion

To find a traffic distribution strategy achieving a good overall delay performance in assigning two types of flows having distinct traffic characteristics over two paths, we analytically examined the queuing delay of paths for every combination of flow assignment. As a result, we found three features in lowering the queuing delay in distributing flows: 1) in the case of same-bandwidth paths, equal division of both types of flows could lower (or minimize in some cases) the queuing delay, and there were multiple flow assignments similarly lowering the queuing delay; 2) in the case of different-bandwidth paths, a trivial flow-distribution strategy, such as equalizing utilization between paths or separating flows completely by their traffic characteristics could not always lower the queuing delay; 3) the flow assignments lowering the mean of queuing delay could approximately lower both the variance and 99.5^{th} percentile of queuing delay.

The topics remaining as future work are devising a method with the ability to distribute m types of flows over n paths ($n,\ m > 0$), based on their traffic characteristics of flows and the bandwidth of paths, and developing a practical method to measure the traffic characteristics in time.

Acknowledgements

This work was supported in part by the Japan Society for the Promotion of Science, a Grant-in-Aid for Scientific Research on (A) (15200005).

References

1. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao: Overview and Principles of Internet Traffic Engineering. RFC 3272, (2002)
2. A. Elwalid, C. Jin, S. Low and I. Widjaja: MATE: MPLS adaptive traffic engineering. Proc. of the Infocom Anchorage (2001) 1300–1309
3. T. Guven, C. Kommareddy, R. j. La, M. A. Shayman, B. Bhattacharjee: Measurement Based Optimal Multi-path Routing. Proc of IEEE Infocom 2004, Hong-Kong, (2004) 187–196
4. P. Siripongwutikorn, and S. Banerjee: Per-flow Delay Performance in Traffic Aggregates. Proc. of IEEE Globecom, vol. 21, no. 1, Taipei, (2002) 2641–2645
5. V. Trecordi, and G. Verticale: Per-flow Delay Performance in a FIFO Scheduler fed by Policed UDP Sources. Comp. Commun., vol. 23, (2000) 309–316
6. Y. Xu, and R. Guerin: Individual QoS versus Aggregate QoS: A Loss Performance Study. in Proc. IEEE Infocom 2002, vol. 3, New York, (2002) 1170–1179
7. D. Anick, D. Mitra, and M. M. Sondhi: Stochastic theory of a data-handling system with multiple sources. The Bell system Technical Journal, vol. 61, no. 8, (1982) 1871–1894
8. V. Ramaswami: Matrix analytic methods for stochastic fluid flows. Proc. of ITC 16, Amsterdam (1999) 1019–1030

Open and Association MCTAs Access and Allocation Scheme by Staggering Algorithm in IEEE 802.15.3

Eui-Seok Hwang[1], You-Chang Ko[2], Choong-Ho Cho[3], Hyong-Woo Lee[4], and Sumit Roy[1]

[1] Univ. of Washington Dept. of Electrical Engineering
{eui, roy}@ee.washington.edu
[2] LG Electronics Inc. Mobile Handset R&D Center
ycko@lge.com
[3] Korea Univ. Dept. of Computer & Information Science
chcho@korea.ac.kr
[4] Korea Univ. Dept. of Electronics & Information Engineering
hwlee@korea.ac.kr

Abstract. The IEEE 802.15.3 medium access control (MAC) protocol is standard for high bit rate wireless personal area network (WPAN). The open or association management channel time allocations (MCTAs) are used by devices(DEVs) for sending command messages or association request command to piconet coordinator (PNC) by means of slotted aloha random access manner. Based on slotted aloha scheme the binary back-off algorithm has been considered as a primary contention resolution candidate due to its simple operation. However it is not appropriate for the future wireless networks because of low throughput and high delay and delay variance. Without loss of generality the goals of the random multiple access algorithm are to maximize the throughput and to minimize the average packet delay. In this paper, we propose a new multiple access protocol named staggering algorithm working on top of slotted aloha scheme. The performance results by NS-2 simulation show that the proposed algorithm achieves the maximum throughput up to 0.54 and guarantees the QoS in terms of delay and delay variance of realtime multimedia traffic.

1 Introduction

IEEE 802.15.3 working group [1] is working on technologies targeted at enabling high bit rate multimedia applications operating in WPAN. These technologies include both MAC and PHY protocols that enable WPAN to support up to 243 DEVs operating at least 20Mb/s [2]. IEEE 802.15.3 piconet is a wireless ad hoc data communications system which allows a number of independent data DEVs to communicate with each other. To provide multimedia QoS, a TDMA-based superframe structure is adopted. The superframe is composed of three major parts: the beacon, the optional contention access period (CAP) and the channel

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 493–501, 2006.

time allocation period (CTAP), as shown in Figure 1. Any DEV associated in the piconet may attempt to send a command frame to the PNC in an open MCTA. Any DEV not currently associated in the piconet also may attempt to send an association request command to the PNC in an association MCTA. It is the PNC's responsibility to determine the number of MCTAs to use for each superframe. It is desirable that the number of MCTAs is dynamically adapted by the PNC depending on the current traffic conditions. As a random access scheme slotted aloha was proposed for an open MCTA or an association MCTA in [2]. However, this mechanism in line with the exponential back-off algorithm for a collision resolution has been challenged by some prior works in order to enhance realtime multimedia traffic access efficiently in terms of throughput, delay, and delay variance[4-6]. In this paper, we propose a new open and association MCTAs access and allocation scheme named staggering algorithm to deal with collision resolution which is inevitable in random access environment such as slotted aloha scheme. NS-2 simulation results show that the proposed scheme provides lower mean delay/delay variation and higher throughput than previous schemes. The throughput of the proposed scheme is remarkably increased up to 0.54.

This paper is organized as follows: Section 2 provides slotted aloha access scheme for open and association MCTAs followed by a description of previous proposed schemes. Section 3 describes the proposed scheme. We present simulation results in section 4. Section 5 concludes the paper.

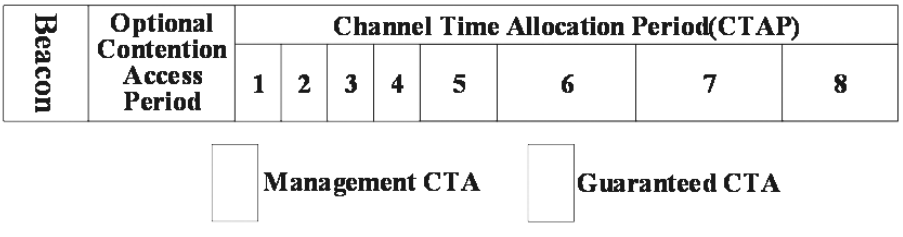


Fig. 1. Superframe structure

2 Related Works

2.1 Slotted Aloha Access for Open and Association MCTAs [2]

The access to an open or association MCTA shall be controlled by a contention window CW_a maintained by each DEV. Each DEV decides CW_a by the number a, where a is the number of retransmission attempts made by the DEV. For the first access attempt, a shall be set to zero. The size of the contention window, CW_a is defined as follows,

$$CW_a = \begin{cases} 256 & 2^{a+1} \geq 256 \\ 2^{a+1} & 2^{a+1} < 256 \end{cases} \quad (1)$$

The open or association MCTA used for the a^{th} retransmission attempt shall be chosen by a uniformly distributed random integer value r within the interval $[1, CW_a]$. The DEV shall start counting r beginning with the open or association MCTA in the current superframe and continue across superframes. The open or association MCTA with number equal to r is the MCTA that the DEV shall access. The DEV shall not access the MCTA before its counter has reached the open or association MCTA with the number equal to r. After receiving the ACK, a will be reset to 0. This retransmission scheme based on slotted aloha is simple but inappropriate for the future wireless networks because of low throughput and high delay variance.

2.2 Previous Scheme [3]

Recently, extensive research has been devoted to IEEE 802.15.3 [6-10]. In this paper we introduce a new random access scheme in IEEE 802.15.3. The random access scheme being used by HIPERLAN/2 is much similar to that of IEEE 802.15.3 such a way that both of them use the centralized random access method controlled by PNC or access point (AP) base on slotted aloha scheme. So, we simply compare one of recently developed random access schemes in HIPERLAN/2 with the proposed scheme.

In [3], an AP controls the number of random channels (RCHs), which works as open or association MCTA in IEEE 802.15.3 does, based on the binary splitting algorithm. By the binary splitting algorithm the number of RCHs of $(t+1)^{th}$ MAC frame is given by

$$r(t+1) = \min\{N_a + 2 \times N_f(t), R_{\mathrm{MAX}}\} \tag{2}$$

where

- $r(t)$: the number of allotted RCHs at MAC frame t
- $N_f(t)$: the number of collided RCHs at MAC frame t
- N_a : the fixed number of RCHs allocated for newly arriving packets
- R_{MAX} : the maximum number of RCHs per MAC frame

For each collided RCH in the previous frame, additional two RCHs are allocated for the collision resolution. For initial attempt, a mobile terminal (MT), which corresponds to a DEV in IEEE 802.15.3, randomly accesses one RCH within the interval [1, N_a] as Eq. 3. The collided MTs in the previous MAC frame may choose the RCH to access based on the location information of contention slots where collisions occur. That is, the MTs in the i^{th} RCH among the collided RCHs in the previous MAC frame would randomly access either $(2 \times i - 1 + N_a)^{th}$ or $(2 \times i + N_a)^{th}$ RCH as Eq. 4. In addition, if there are not enough RCHs in the current MAC frame, the retransmission occurs within the interval $[1, N_a]$ after a random delay by frame unit as Eq. 5.

◇ Initial attempt: Random access within $[1, N_a]$ (3)
◇ Retransmission:

Random access either $2i-1+N_a$ or $2i+N_a$ if $2i+N_a \leq R_{\mathrm{MAX}}$ (4)

Random access within $[1, N_a]$ after a random delay if $2i+N_a > R_{\mathrm{MAX}}$ (5)

where i is the location of collided RCH in the previous MAC frame.

3 Proposed Scheme: Staggering Algorithm

In the proposed scheme the structure of time slot is changed such a way that each slot is separated into two areas, the front part and the rear one, and most of their parts are overlapped each other. The collided DEVs by accessing the front part of a slot in the previous superframe would access any part of the first slot of additional two slots prepared by binary splitting for the collision resolution. The collided DEVs by accessing the rear part would access any part of the second slot of two additional slots. If all collided DEVs are in the same part of slot, they choose any part of any two additional slots randomly.

The PNC controls the number of MCTAs based on the splitting algorithm. The number of MCTAs of $(t+1)^{th}$ superframe is given by

$$r(t+1) = \min\{N_a + 2 \times N_f(t), R_{\mathrm{MAX}}\} \tag{6}$$

where

- $r(t)$: the number of allotted MCTAs at superframe t
- $N_f(t)$: the number of collided MCTAs at superframe t
- N_a : the fixed number of MCTAs allocated for newly arriving packets
- R_{MAX} : the maximum number of MCTAs per superframe

There are N_a MCTAs for newly arriving request packets in each superframe. For each collided MCTA in the previous superframe, additional two MCTAs are allocated for the collision resolution. For the initial attempt, a DEV randomly accesses any part of one MCTA within the interval $[1, N_a]$ as Eq. 7. If a collision occurs, the collided DEVs in the previous superframe choose the MCTA to access again based on the location information of contention slots where collisions occur. That is, if all collided DEVs are in the same part of the i^{th} MCTA among the collided MCTAs in the previous superframe, they randomly access any part of either $(2\times i-1+N_a)^{th}$ or $(2\times i+N_a)^{th}$ MCTA as Eq. 9. Otherwise the collisions are occurred in both parts of slot, the collided DEVs in the front part of slot access any part of $(2 \times i - 1 + N_a)^{th}$ MCTA as Eq. 10. and the others access any part of $(2 \times i + N_a)^{th}$ MCTA as Eq. 11. In addition, if there are not enough MCTAs in the current superframe, the retransmission occurs within the interval $[1, N_a]$ after a random delay by frame unit as Eq. 8.

◇ Initial attempt: Random access within $[1, N_a]$ (7)

◇ Retransmission:

– If $2 \times i + N_a > R_{\mathrm{MAX}}$
access any part of slot within $[1, N_a]$ after a random delay, (8)
where i is the location of the collided MCTA in the previous superframe.

- Else if all collided DEVs are in the same part
  access any part of either $i \times 2 - 1 + N_a$ or $i \times 2 + N_a$ (9)
- Else if collided DEVs are in the front of slot
  access any part of $i \times 2 - 1 + N_a$ (10)
- Else if collided DEVs are in the rear of slot
  access any part of $i \times 2 + N_a$ (11)

Fig. 2 shows an example of how the number of MCTAs would be decided by the staggering algorithm, in which the number of MCTAs for new requests is assumed by two and R_{MAX} is eight. If four MCTAs are collided in the previous superframe, the number of MCTAs will be 10 in the current superframe(two MCTAs for new requests and eight MCTAs that are splitted by four collided MCTAs). However, the number of MCTAs in the current superframe should be eight because R_{MAX} is limited by eight.

In the current superframe, new DEVs can access the first or second MCTA. The DEVs collided on the front part of the first slot,C_{11} of the previous superframe access any part of the 3^{rd} slot. The DEVs collided on the rear part of the first slot, C_{12} of the previous superframe access any part of the 4^{th} slot. Similarly, the DEVs collided on C_{21} of the previous superframe access any part of the 5^{th} slot. The DEVs collided on C_{22} of the previous superframe access any part of the 6^{th} slot. Since all collisions are occurred in the front part of the 4^{th} slot of the previous superframe, the DEVs of C_{31} randomly access either the 7^{th} or the 8^{th} slot. However, the DEVs involved in C_{42} in the previous superframe cannot access any MCTA slot in the current superframe because the maximum number of MCTAs is limited by eight. In this case, these DEVs should access MCTAs in N_a area as if they are newly arrived packets after a random delay. After all, the proposed scheme can resolve collisions quickly by means of reducing randomness by staggering algorithm when DEVs access MCTAs.

4 Simulation Results

This section presents some NS-2 simulation results on the throughput, average delay, and delay variance. For simulation, we assume that there is no transmission error due to radio channel environment, there are one PNC and 50 DEVs as a whole, and each DEV generates the message according to Poisson process with rate λ per superframe. We also assume a DEV cannot generate a new request message until the access attempt succeeds. Throughout this section the (normalized) throughput, ρ, is defined as the fraction of MCTAs in which successful transmissions occur.

In Figs. 3, we observe that the maximum throughput of the proposed scheme is 54%, while that of [3] is 47% and fixed MCTA allocation scheme is 36%. Slotted aloha access mechanism [2] is adapted for fixed MCTA allocation scheme in Fig. 3. Therefore, the proposed scheme utilizes the MCTAs more efficiently than previously proposed schemes. The number of allocated MCTA is one of [1, 2, 4, 8, 16, 32] in the fixed MCTA allocation scheme. The maximum throughput can be got when the number of MCTA is small if the offered load is low and

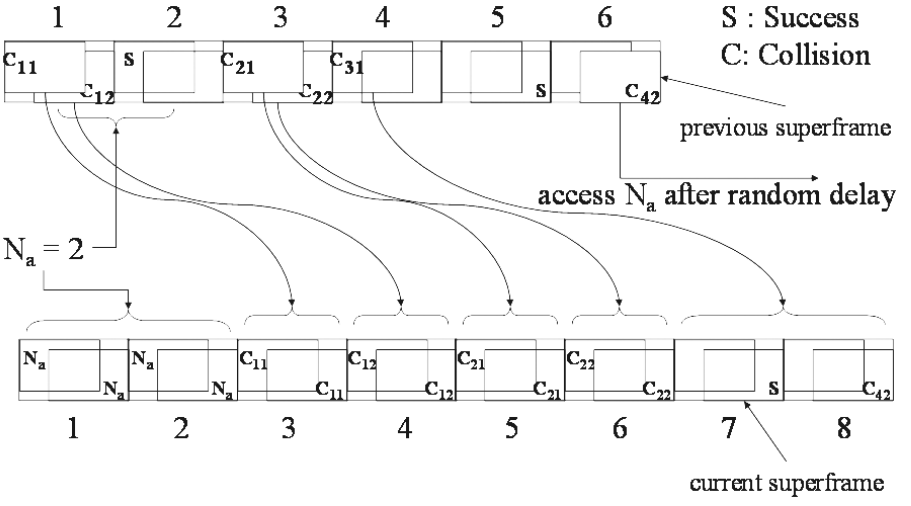


Fig. 2. An example of the staggering algorithm($N_a = 2$ $R_{MAX} = 8$)

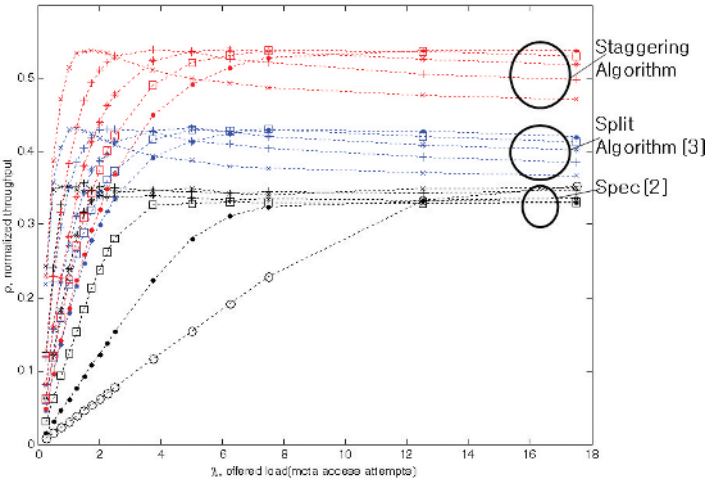


Fig. 3. Offered load versus throughput
(Staggering Alog. VS. Split Algo. VS. Fixed MCTA allocation scheme)

vice verse. However, it can be anticipated easily the maximum throughput of these networks is at most 1/e and high delay and standard deviation of delay because the basic MAC protocols of these networks are slotted ALOHA with back-off. Several lines of staggering and split algorithm result from different N_a. In general, when the MCTA access attempt is low, the higher throughput can be achieved when N_a is low. That is, allocating many MCTAs for new attempts is a

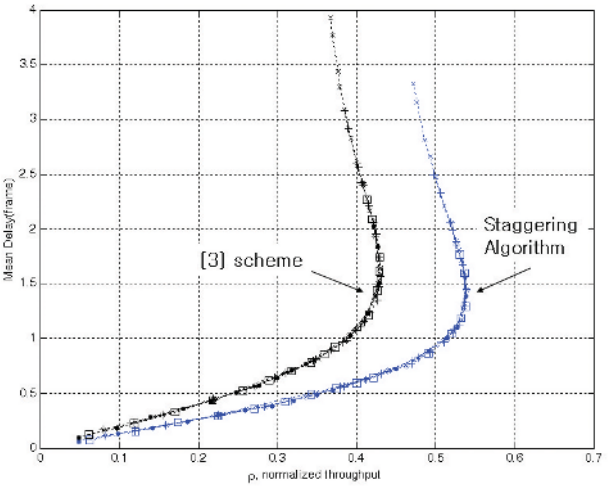


Fig. 4. Throughput versus mean delay (Staggering Algo. VS.[3])

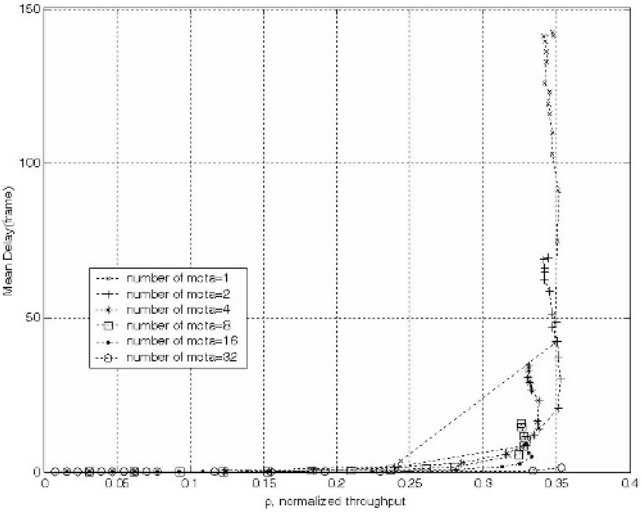


Fig. 5. Throughput versus mean delay (Fixed MCTA allocation scheme)

waste of resources when access attempt is low. When the MCTA access attempt is high, throughput can be improved when N_a is high.

Fig. 4 shows the throughput-delay characteristics of the proposed scheme. For example, the proposed scheme shows the delay within one frame when throughput is 0.4, however, the previous scheme [3] shows the delay beyond one frames. Of course, extreme delay is shown in the fixed MCTA allocation scheme when the number of allocated MCTA is small as shown in Fig. 5. Real-time traffic like streaming video and audio as well as best-effort traffic is anticipated to occupy

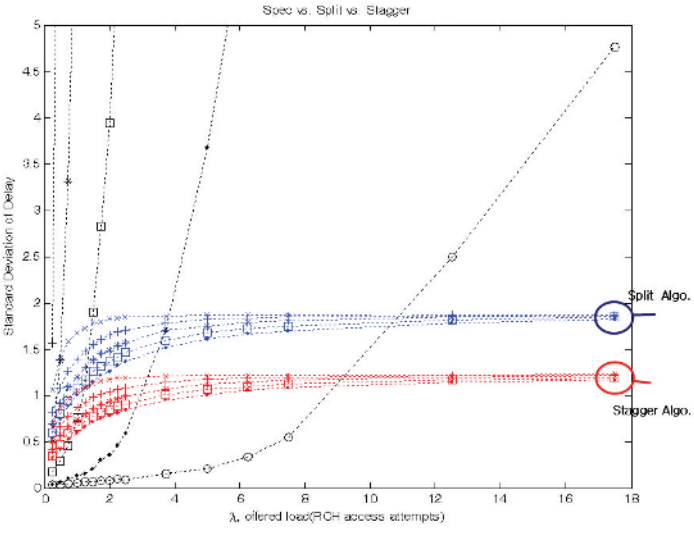


Fig. 6. Offered load versus delay variance

a large portion of WPAN traffic[11]. Therefore, we can expect that the proposed scheme can be useful in guaranteeing QoS of multi-media traffic in WPAN.

Fig. 6 describes the proposed staggering algorithm can guarantee the delay jitter. Standard deviation of delay can be an indicator of sharing the limited resource equally and another importance parameter in the next generation wireless networks. The different lines in the same algorithm are the difference of N_a. Back-off algorithm allows a single or a few winning user to dominate the available bandwidth. Therefore the shared channel can be used unfairly in the back-off algorithm.

5 Conclusion

The slotted aloha scheme with exponential back off algorithm for the IEEE 802.15.3 is simple to implement. Yet, it has the relatively low maximum throughput of 1/e and suffers from large mean and variance of access delay. For the next generation of wireless networks with multimedia services, the delay throughput performance of the slotted aloha scheme with exponential back off needs to be improved. In this paper, we have proposed a new open and association MCTA access and allocation scheme. Through computer simulations, we have shown that the proposed staggering algorithm achieves the maximum throughput as high as 0.54. The result, when compared with previously proposed algorithms [2, 3] gains more than 0.1. The proposed scheme also performs well in terms of delay and delay variance. This can be significant for QoS guarantee for delay sensitive real time multimedia traffic in WPAN.

References

1. IEEE, "802.15 WPAN task group 3(TG3)", http://www.ieee802.org/15/pub/TG3.html
2. IEEE 802.15.3 Working Group, "Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN)." IEEE Computer Society, Sep. 2003
3. Eui-Seok Hwang, et al: "Random Channel Allocation Scheme in HiperLAN/2", LNCS 3331 25 32pp, 2004
4. You-Chang Ko et al: "Collision Reduction Random Access Using m-ary Split Algorithm in Wireless Access Network" LNCS3510, pp.223 233, 2005.
5. Gyung-Ho Hwang, Dong-Ho Cho: "Adaptive Random Channel Allocation Scheme in HIPERLAN/2 ", IEEE COMMUNICATIONS LETTERS, VOL. 6, NO. 1, JAN. 2002, pp. 40-42
6. Kwan-Wu Chin and D. Lowe: "Simulation Study of the IEEE 802.15.3 MAC", Australian Telecommunications and Network Applications Conference (ATNAC), Sydney, Australia, December, 2004
7. Seung H. Rhee, K. Chung, Y. Kim, W. Yoon, and K. S. Chang: "An Application-Aware MAC Scheme for IEEE 802.15.3 High-Rate WPAN", IEEE WCNC, Mar. 2004
8. Byung S. Kim, Y. Fang and Tan F. Wong: "Rate-Adaptive MAC Protocol in High-Rate Personal Area Networks" , IEEE WCNC, Mar. 2004
9. Attila Torok, Lorant Vajda, Kyu J. Youn and Sun-Do June: "Superframe Formation Algorithms in 802.15.3 Networks", IEEE WCNC, Mar. 2004
10. Xin Wang, Yong Ren, Jun Zhao, Zihua Guo and Richard Yao: "Comparison of IEEE 802.16e and IEEE 802.15.3 MAC", IEEE 6th CAS Symp. On Emerging Technologies, Shanghai, China, May, 2004
11. P. Gandolfo and J. Allen: "802.15.3 Overview/Update," The WiMedia Alliance, Oct. 2002

Cumulative-TIM Method for the Sleep Mode in IEEE 802.16e Wireless MAN*

Byungjoo Lee[1], Hyukjoon Lee[1], Seung Hyong Rhee[1],
Jae Kyun Kwon[2], and Jae Young Ahn[2]

[1] Kwangwoon University, Seoul, Korea
parang@kw.ac.kr, {hlee, shrhee}@daisy.kw.ac.kr
[2] Electronics and Telecommunications Research Institute, Daejeon, Korea
{jack, jyahn}@etri.re.kr

Abstract. The IEEE 802.16e WMAN (Wireless Metropolitan Area Networks), which has been designed for fixed or mobile broadband wireless access, is getting into the spotlight as the base technology for the mobile Internet. As most mobile stations are battery-powered in wireless environment, energy efficient protocols are essential for their practical use. Although many systems, including the IEEE 802.16e, adopt the sleep mode, little attention has been paid to the traffic characteristics or status of the mobile station. In this paper, we propose a new scheme called *Cumulative-TIM*, in which BS(base station) does not wake up a mobile station until a sufficient amount of data for the station is stored, thus guaranteeing mobile stations sleep enough amount of time. In this way, mobile stations can use their energy more efficiently than the standard sleep mode: Our simulation result shows that proposed scheme has a better performance compare with the current standard of the WMAN.

1 Introduction

Recently, remarkable advances in communication technologies have provided high-speed and reliable services such as video broadcasting and multimedia streaming. IEEE 802.16 wireless MAN, one of such technologies, has been standardized for fixed broadband wireless systems in 2001[2]. This system uses 10∼66GHz frequency band for transmissions of several mega bytes data files or real time streaming service. Thereafter IEEE 802.16d has been published for support of mobility using 2∼11GHz licensed frequency band, which provides network access to buildings through outside antennas communicating with central station. Additionally, IEEE 802.16e is being developed for wireless broadband system that has high speed and mobility[1]. This standard will fill the gap between fixed wireless local area networks and mobile cellular systems. Many people are making constant efforts for developing the mobile Internet adopting this standard, and it will be one of base technologies for the 4th generation communication system.

* This work has been conducted by the Research Grant of Kwangwoon University in 2005, and in part by Grant No. R01-2005-000-10934-0 from Korea Science and Engineering Foundation in Ministry of Science & Technology.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 502–511, 2006.

In typical wireless systems, mobile devices such as PDA, cellular phones and laptop, are battery-powered for their operation. As the battery capacity is usually constrained via its size, the battery life is so crucial that it is directly related to the life time of the system. If a certain station is drained of its battery, it will not be able to communicate with other stations, and it severely damages the connectivity of the system. In order to increase the energy-efficiency of a communication system, MAC-layer protocols usually adopt an energy-saving mode called power saving mode or sleep mode, in which mobile devices shut off their power and go to the sleep state when there are no data to transmit or receive. Although most wireless communication protocols have this mechanism, few consider each station's traffic characteristics or status.

Many researchers have been paying attentions on the problems of sleep mode, mostly on the modeling and performance analysis of the mode. In [5], the authors investigate the queueing behavior of the sleep mode operation in IEEE 802.16e WMAN, analyzing the power conservation of MSS(Mobile Subscriber Station) in terms of the dropping probability and the mean waiting times in the queue of the BS(Base Station). Assuming Poisson arrival and general service time distributions, they shows the binary exponential backoff scheme has a good performance for increasing the sleep interval. On the other hance, [6] describes the energy consumption in IEEE 802.16e WMAN, by modeling the sleep mode analytically. They evaluate the sleep mode using an analytical model and validate the results by simulations.

As stated above, the performance or the energy-efficiency of the sleep mode can be improved if the BS considers the traffic characteristics and status of the mobile stations. In this paper, we explain how the improvement can be achieved, and propose a new scheme called *Cumulative-TIM*. In our mechanism, BS does not wake up a destination MSS until a sufficient amount of data, bounded to the MSS, is gathered, thus guaranteeing mobile stations sleep enough amount of time. Positive TIM message, sent by BS, wake up the MSS, and MSS determines how often the BS sends the positive TIM message, based on the traffic characteristics or the status of the mobile station. For example, if an MSS is suffer from low energy, it may request the BS to send the message at long intervals. In this way, MSS can use energy more efficiently than the standard sleep mode defined in IEEE 802.16e WMAN draft document.

The remaining part of this paper is organized as follows. Chapter 2 briefly introduces the IEEE 802.16e wireless MAN protocol and its sleep mode. In chapter 3, we propose a new sleep mode using the Cumulative-TIM scheme. Simulation results are provided and discussed in chapter 4, and finally chapter 5 concludes this paper.

2 Preliminaries

2.1 IEEE 802.16e Wireless MAN

The IEEE 802.16e wireless MAN has been designed to provide mobility to the subscriber stations, in addition to their high-speed and wide-area transmission

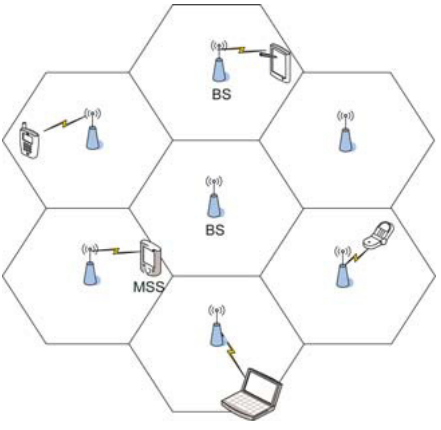


Fig. 1. Multiple cells of the IEEE 802.16e WMAN

capability. The standard, which includes MAC and PHY specifications, supports higher transmission speed than IEEE 802.16 and IEEE 802.16a in lower movement speed. Typically, MSS is able to transmit data at 70Mbps while moving in 60Km/h of speed. The basic unit of the WMAN is a cell which is composed of one BS and multiple MSSs, and supports point-to-multipoint communications[3]. One BS manages the single cell that has 1∼2Km coverage and uses 2∼6GHz licensed frequency band. Fig. 1 shows a simple example of the IEEE 802.16e WMAN: There are a number of hexagonal cells which are managed by the BS located in the center of cells, and have MSSs communicate through their BS. Fig. 2 shows the frame structure which is determined by BS and used for the control and communications in the cell. The beginning of the frame located the preamble that means the frame starting. Next duration of the frame is divided into downlink subframe and uplink subframe. BS refer to DL-MAP(downlink-map) for using the downlink, and MSS consult UL-MAP(uplink-map) to transmit data via uplink. Wireless access control in the uplink direction is done by time division multiple access.

2.2 Sleep Mode in IEEE 802.16e

The IEEE 802.16e Wireless MAN offers a sleep mode for power saving and energy efficient communications. Typical power saving mechanism has *awake* mode and *sleep* mode. In the former, a mobile station is able to transmit and receive data any time. Usually, MSS consumes a lot of energy in awake mode than sleep mode. In the awake mode, even when MSS does not transmit or receive any data, it consumes the battery since the power is always supplied into the communication circuits. This period, where the energy is consumed while no data is transmitted or received, is called an *idle* interval. In the latter, the mode is divided into *sleep* intervals and *listen* intervals. In the sleep interval, MSS stops to supply power into the communication circuits and thus hardly uses its energy resource. As MSS is unable to transmit or receive data during this interval, increasing

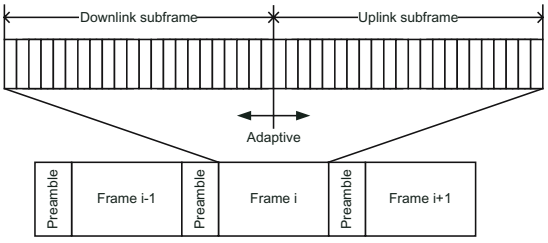


Fig. 2. IEEE 802.16e WMAN frame structure

the sleep interval helps to save the energy resource, while it may degrades the service quality and throughput. According to a schedule, MSS supply energy into the circuits and returns back to the listen interval. Then MSS operates normally and it can transmit and receive data any time. MSS receives a MOB-TRF-IND message from the BS at the beginning of every listening interval. The TIM(Traffic Indication Message) informs MSS of whether data is waiting at BS and it should remain awake to receive the data. Positive-TIM lets MSS be awake during next frame, while negative-TIM allows MSS to go to the sleep interval. Thus, if BS has data bounded to the MSS, it sends the Positive-TIM, otherwise it uses negative-TIM. MSS changes its status according to the events as depicted in Fig. 3.

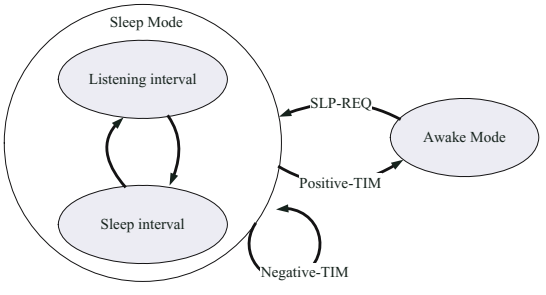


Fig. 3. MSS mode transitions

$$\begin{cases} I_0 = \text{initail-sleep window} \\ \vdots \\ I_k = min(I_{k-1} \cdot 2^{\text{ exponent value}}, \text{ final-sleep window}) \, , k > 0 \end{cases} \tag{1}$$

MSS is required to send MOB-SLP-REQ message to BS before it transits to sleep mode. After BS replies with MOB-SLP-RSP message, MSS may enter the sleep mode. MOB-SLP-REQ message sent by MSS contains a lot of information regarding the sleep mode: initial-sleep window size, final-sleep window size, listening interval duration and final-sleep window exponent value. Sleep

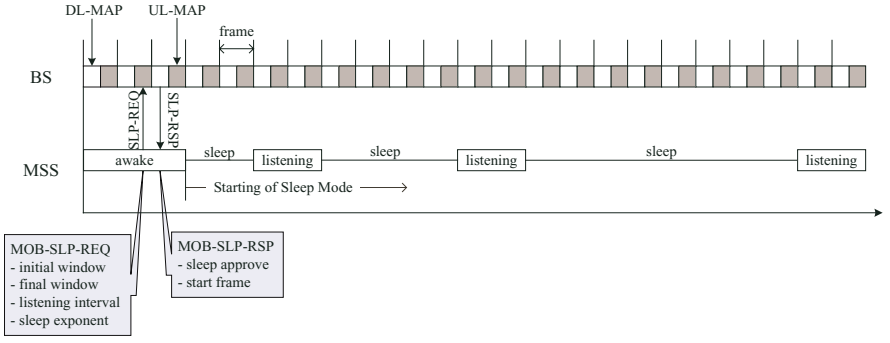


Fig. 4. Sleep mode operation in IEEE 802.16e WMAN

interval duration increases from initial-sleep window to final sleep window according to final-sleep window exponent value. Fig. 4 shows this process where the initial-sleep window is set to 2 frames, the listening interval is 2 frames, and the final-sleep window exponent is 1.

3 Cumulative-TIM Method

In this chapter, we propose *Cumulative-TIM* scheme which provides a better energy efficiency than the standard method of sleep mode in the WMAN. We will describe how the method can improve the energy efficiency, how the scheme works, and what is the differences compared to the legacy sleep mode. In the sleep mode, BS wakes up MSS every time it receives data bounded for the MSS by sending positive-TIM. Thus MSS returns back to normal state regardless of its status or the traffic characteristics. It is obvious that, in some cases, MSS is better to be in a longer sleep interval with additional transmit delay in BS: For example, if the traffic to the MSS is non-real time such as e-mail, MSS need not to wake up immediately and receive the data, and it would be better to be the sleep interval in a longer period of time. Another example is that, if MSS does not have enough battery resource, it may prefer to stay in a longer sleep interval with some transmit delay. Current standard of the sleep mode does not support dynamic adjustment of the sleep interval according to the status and condition of the MSS.

Our proposed Cumulative-TIM allows MSS to represent its tradeoff between energy-saving and delay. MOB-SLP-REQ message from MSS contains *sleep depth*, which is decided by MSS and tells how fast MSS wants to receive the data: If MSS, for example, is short of battery resource, then it adopts a high value of sleep depth that informs BS not to wake up MSS until a sufficient amount of data destined to the MSS is stored. On the other hand, if MSS is receiving real-time traffic, it will use a low value of sleep depth, which results in immediate wake-up and thus prompt transmission of data. BS stores the sleep depth of each MSS and, based on the information, determines when it should send the

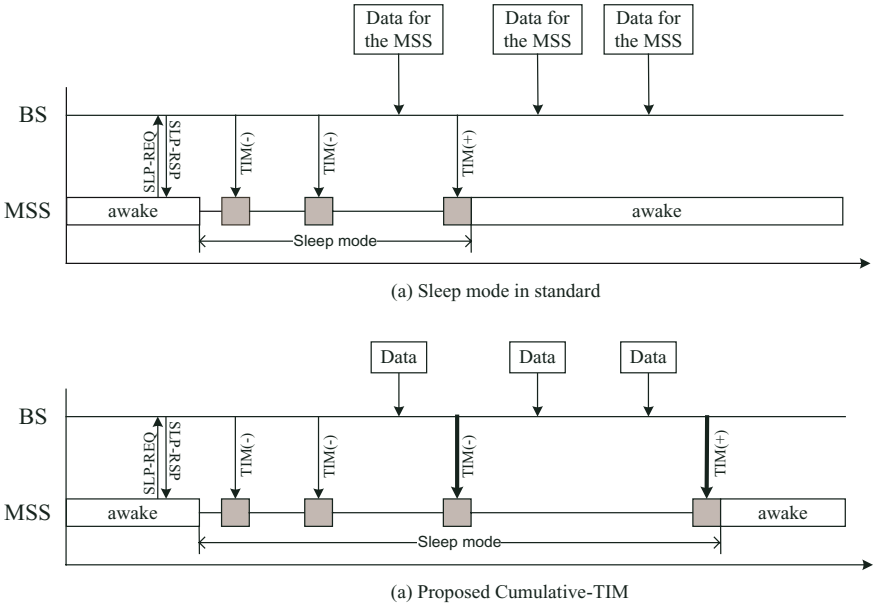


Fig. 5. Comparison of standard sleep mode and proposed sleep mode

positive-TIM: When the number of data packets stored in BS reaches the sleep depth determined by the MSS, BS sends Cumulative-TIM in order to wake up the MSS.

Fig. 5 shows how our proposed method achieves the energy efficiency. In the standard sleep mode, BS sends positive-TIM immediately after it receives data destined to the MSS. This positive-TIM makes the MSS to go into the awake mode. Although this method provides the minimum latency for the communication, it is not efficient in energy-saving: MSS wakes up every time data is destined to it. In the Cumulative-TIM scheme, however, BS sends negative-TIM even if it receives data destined to the MSS before the awake interval. The data is stored in BS's buffer until its amount reaches the sleep depth. When BS sends positive-TIM and MSS wakes up, BS sends all the stored data. Thus, as MSS adopts more larger value of sleep depth, it enjoys longer sleep interval and saves more energy. Fig. 6 depicts MSC(Message Sequence Chart) for the two sleep modes.

Our proposed method has a couple of advantages over the legacy sleep mode. First, it optimize the energy consumption and maximize the life time of battery resource and thus the life time of the mobile station. Extending the battery life may cause some additional latency in data transmission. However, MSS may control the tradeoff between data delay and energy-saving based on the traffic characteristics or status of the station. Second, BS is able to manage the down link more efficiently. Since MSS select their own sleep depth according to their status, resource management for the down link can be achieved in a distributed

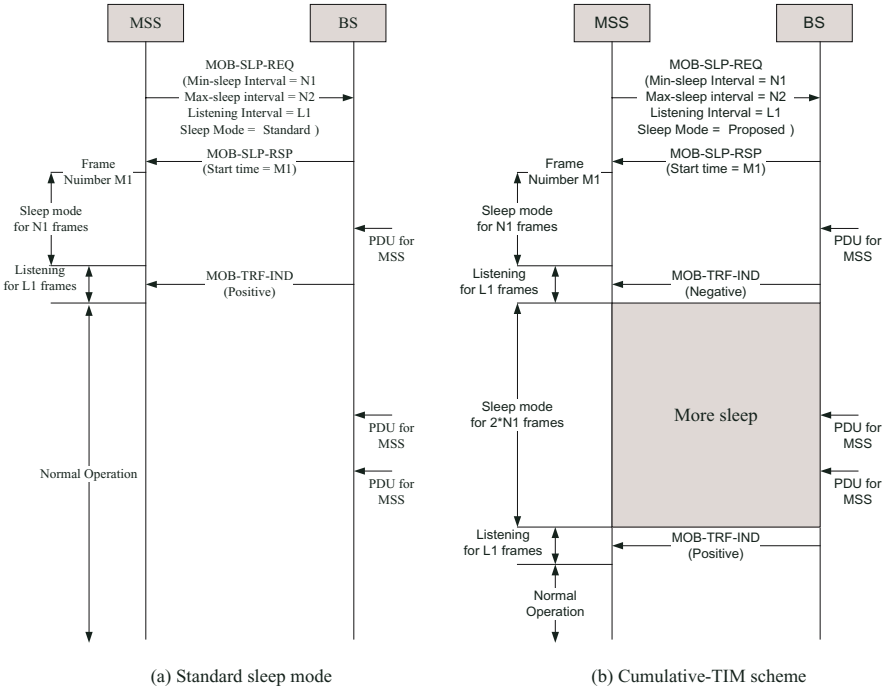


Fig. 6. Message sequence charts for two sleep modes

way: for example, if one MSS is short of energy or receives non-real time traffic, it may use less amount of bandwidth in the down link.

4 Performance Evaluations

4.1 Simulation Environment

We have implemented our proposed sleep mode using the ns-2 network simulator with the CMU wireless extension. We have built the BS that manages frame with the UL-MAP and DL-MAP, and the stations which access the channel based on TDD mechanism using the MAP. In our simulator, single cell consists of one BS and multiple MSS, and one TDD frame consists of equal amount of down link and up link channels. The mean of traffic burst duration is set to 10ms and mean of idle duration is 100ms. For the sleep mode parameters, we use 2 frames of initial-sleep window, 16 frames of final-sleep window, 2 frames of listening-window, and final-window exponent value of 2. The parameters used for the simulation are summarized in Table 1. In addition, we assumed that the subscribe stations are fixed during the simulation and no handoffs are allowed.

Table 1. Simulation parameters

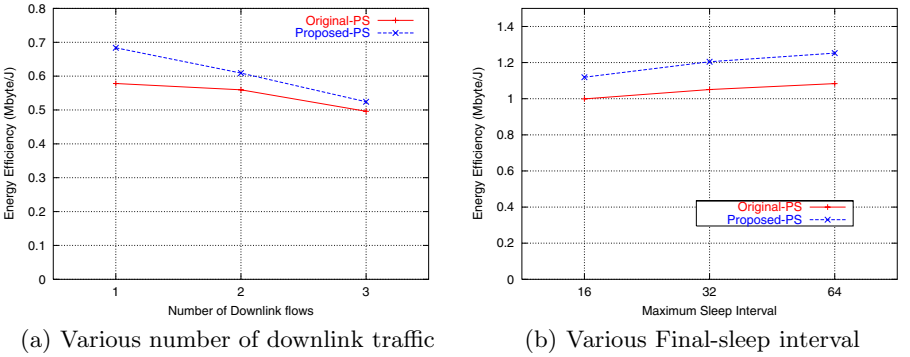
Attribute	Value
Traffic Type	Exponential random
Idle Interval	100ms
Burst Interval	10ms
Initial Energy	10J
TX power consumption	1.6W
RX power consumption	1.2W
Idle power consumption	0.4W
Final-Sleep interval	16 frames
IDLE Timer	0.01 sec

4.2 Results

In this section we evaluate the performance of the proposed Cumulative-TIM scheme by simulations. We adopt *energy efficiency* given in (2) as a metric for the performance evaluation, which is a widely used one in related literature. It shows how the energy resource is consumed efficiently: For example, if the energy efficiency is 1, MSS transmits or receives 1Mbyte of data using 1 joule of energy.

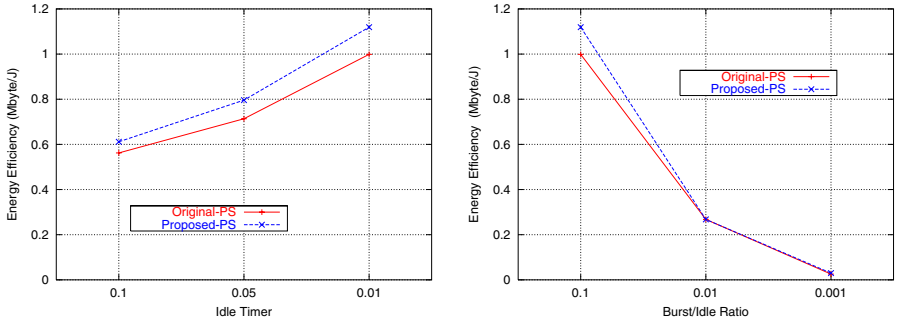
$$\eta = \frac{Aggregate\ Throughput}{Consumed\ Energy}(Mb/sec \cdot J)\,. \tag{2}$$

Fig. 7 clearly shows the effect of the proposed sleep mode. In Fig. 7(a), we measured the energy efficiency according to different values of the number of downlink traffic flows. As the number of flows decreases, cumulative-TIM outperforms the legacy sleep mode and vice versa. Since MSSs in the cell share a common channel resource via the time division duplex, our proposed scheme uses the energy efficiently. High energy efficiency means that MSS use the same energy more efficiently, and thus transmit more data with the energy. The Fig. 7(b)



(a) Various number of downlink traffic (b) Various Final-sleep interval

Fig. 7. Simulation Results



(a) Various Idle timer (b) Various burst/idle ratio

Fig. 8. Simulation Results

shows a result when MMS changes the final-sleep window size as it goes into the sleep mode. Cumulative-TIM shows better performance than the standard sleep mode regardless of the interval values. MSS stays in the sleep mode for longer period of time if the final sleep interval increases; thus, both sleep modes show better energy efficiencies as the window size increases.

In the Fig. 8(a), we simulated the sleep modes using different values of idle timers. The idle timer determines how often MSS enters into the sleep mode: If there is no data in the down link until the timer expires, MSS goes into the sleep mode. When data is sent or received, idle timer is reset and count the time of idle state. In this simulation result, proposed sleep mode saves more energy, when the idle timer increases: i.e., the energy efficiency is better when the timer expires earlier. However, in any values of the timer, cumulative-TIM method shows better performance than the standard sleep mode. According to the Fig. 8(b), the energy efficiency rapidly drops when the burst-idle ratio increases. Burst-idle ratio determines how the traffic is bursty, and bursty traffic decreases the energy efficiency. With extensive simulations, we can conclude that the proposed cumulative-TIM outperforms the standard method in terms of the energy efficiency in any cases.

5 Conclusion

This paper proposed cumulative-TIM method for IEEE 802.16e wireless MAN in order to improve the energy efficiency of the sleep mode. MSS selects sleep depth which determines the tradeoff between energy efficiency and data delay. BS does not wake up a destination MSS until a sufficient amount of data, bounded to the MSS, is gathered, thus guaranteeing mobile stations sleep enough amount of time. Using the method, MSS avoids unnecessary energy consumption and save more energy. We validated the effect of our scheme using extensive simulations, showing the efficiency is improved about 20% in average.

Our future plan includes extending cumulative-TIM such that more practical situation can be considered in the model. For example, MSS may determine its sleep depth based on its remaining battery or the traffic type.

References

1. IEEE 802.16e/D5-2004, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Nov., 2004.
2. IEEE 802.16-2001, "IEEE Standard for Local Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems," Apr. 8, 2002.
3. C. Eklund, R. Marks, K. Stanwood and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," IEEE Communication Magazine, Jun., 2002.
4. S. Ramachadran, C. Bostian and S. Midkiff, "Performance Evaluation of IEEE 802.16 for Broadband Wireless Access," Technical document, 2002.
5. J.Seo, S.Lee, N.Park, H.Lee and C.Cho, "Performance Analysis of Sleep Mode Operation in IEEE 802.16e," *IEEE Vehicular Technology Conference*, Sep., 2004.
6. Y. Xiao, "Energy Saving Mechanism in the IEEE 802.16e Wireless MAN," *IEEE Communication Letters*, Jul., 2005.
7. R. Managhanram, M. Demirhan, "Performance and simulation analysis of 802.15.3 QoS," IEEE 802.15-02/293 contribution, 2002.
8. The CMU Monarch Project, Wireless and mobile extension to ns Snapshot Release 1.1.1., Carnegie Mellon University, 1999.
9. K. Fall and K. Varadhan, "The ns Manual," UC Berkeley, 2001.

An Overload-Resilient Flow Removal Algorithm for M-LWDF Scheduler

Eunhyun Kwon[1], Jaiyong Lee[1], and Kyunghun Jung[2]

[1] Department of Electric and Electronic Engineering,
School of Engineering, Yonsei University,
134 Shinchon-Dong Seodaemun-Gu, Seoul, Korea
{ehkwon, jyl}@nasla.yonsei.ac.kr
[2] Telecommunication R&D Center, Samsung Electronics Co., LTD.,
416 Maetan-3dong, Yeongtong-Gu, Suwon-si, Gyeonggi-do, Korea
kyunghun.jung@samsung.com

Abstract. In the real-time multimedia applications, packet delay should meet stringent Quality of Service (QoS) requirements. Delay Earliest Due Date (EDD) scheduler, originally designed for wireline data networks to operate under a maximum allowed delay, cannot be directly applied to wireless networks, due to the location-dependent errors and time-varying channel conditions. Several modifications of EDD scheduler have been proposed for wireless applications, which typically assume successful admission control, a condition hard to satisfy with wireless networks. In this paper, we propose a removal algorithm for downlink scheduler designed to perform under overloaded situations. Simulation results show that our proposed algorithm outperforms the conventional ones in the QoS guaranteed flows.

1 Introduction

Recently, demands for multimedia applications are rapidly increasing for wireless networks. Delay-related QoS is essential to fulfill the requirement of real-time multimedia applications, such as video conference or voice over IP (VoIP). To support an acceptable level of service, the real-time traffic should be delivered within a certain delay bound. This objective can be pursued by admission control and packet scheduling. Admission control determines whether the system allows a new connection or not, based on the network load and channel condition of the user. Real-time scheduler should deliver the packets of such a nature before their expiration and minimize the number of dropped packets for the reason.

Delay EDD scheduler [1], originally proposed for wireline data networks, processes first the packet of the earliest deadline. When a connection is admitted to the network, the local delay bound should be negotiated. The deadline of each packet is the arrival time plus the local delay bound, which should be computed when the packet arrives in the queue. The packets not served until the deadline should be dropped by the network or by the mobile station. Due to the location-dependent errors and time-varying channel conditions, delay EDD

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 512–520, 2006.

scheduler cannot be directly applied to wireless networks. In [2], an extension of delay EDD scheduler was presented, which introduced the concept of the leading and lagging flows during the calculation of the deadline. This scheduler ensures a fairness bound, and provides graceful degradation in the QoS. Feasible EDD (FEDD) scheduler [3] is one of the scheduling algorithms that provide a guaranteed packet delay upper bound over time-varying wireless channels. This algorithm serves the packets only in good channel conditions, and excludes the users from scheduling if their channel quality deteriorates. For users of good channel quality, FEDD operates similarly to EDD. Proactive EDD (PEDD) scheduler [4] adjusts a packet's deadline dynamically with a predictive manner.

These schedulers assume a binary channel model (good or bad) but M-LWDF scheduler [5][6], proposed for code division multiple access (CDMA) networks, assumes a flat fading channel model. QoS of data users can be defined in different ways. If the user has a real-time application, delays of most packets should be kept below a certain threshold. The QoS requirement of such a user i can be formulated as

$$Pr\{W_i > T_i\} \leq \delta_i \tag{1}$$

where W_i is the packet delay, and T_i and δ_i are the delay bound and the maximum probability of exceeding it, respectively. The M-LWDF scheduling algorithm can be defined by

$$\{i\} = argmax_i\{\gamma_i W_i(t) r_i(t)\} \tag{2}$$

where r_i is the channel rate and W_i is the packet delay for flow i. The choice of γ_i is very important to control the scheduler, depending on the QoS parameters. In [5], shown is the performance of M-LWDF when γ_i equals to $a_i/\bar{r_i}$, $a_i = -(log\delta_i)/T_i$, and $\bar{r_i}$ is the average channel rate for user i. In M-LWDF, a_i gives higher priority to the packets of shorter remaining time, and $\bar{r_i}$ is related to the proportional fairness.

Scheduling algorithms for real-time traffic typically assume proper operation of connection admission control (CAC). However, in the wireless networks, stringent admission control is very hard to achieve due to the channel variations. When a system is congested from over-admission, outage flows that do not satisfy delay requirements increase rapidly, degrading the QoS. In this paper, we propose a removal algorithm that maximizes the number of guaranteed flows in the delay bounded scheduler. The remainder of this paper is organized as follows. The system model and the wireless channel model are introduced in the next section. In Section 3, we describe a removal algorithm for delay bounded scheduler. Numerical examples are presented in Section 4.

2 System Model and Wireless Channel Model

We consider a centralized downlink scheduler at the Base Station (BS), where each flow is assigned its own queue. The channel access scenario is assumed as Time Division Multiple Access (TDMA) or time-slotted CDMA like CDMA/HDR [7].

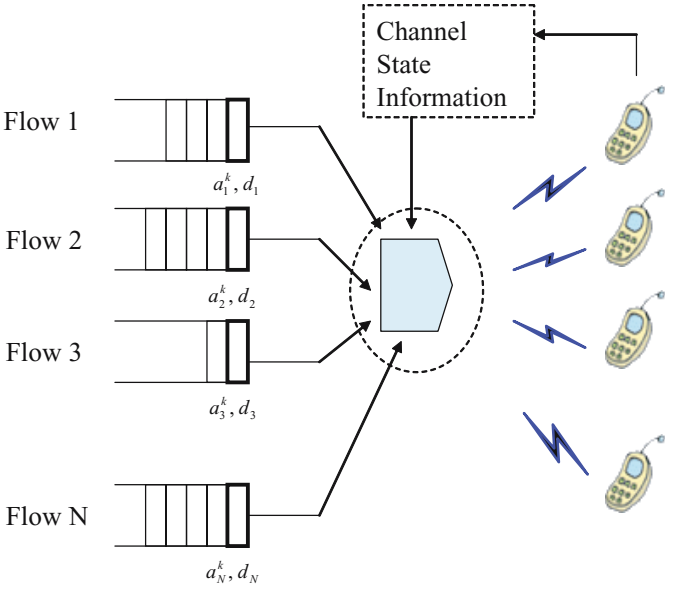


Fig. 1. System Model

The system model is depicted in Fig. 1. a_i^k refers to the arrival time for k^{th} packet in flow i, and d_i designates the delay bound for flow i. We consider the Discrete Time Markov Chain (DTMC) in [6] as our wireless fading channel model. The channel conditions for different users are assumed to be statistically identical and independent. The time-varying channel condition of each user is modeled as a stationary stochastic process, as shown in Fig. 2. The median fading level is denoted as $\bar{R}$, where $R(t)$ is the variable channel rate. Assuming a discrete rate set with three states, $R(t)$ can be either $\bar{R}$ or $\bar{R} \pm 3dB$.

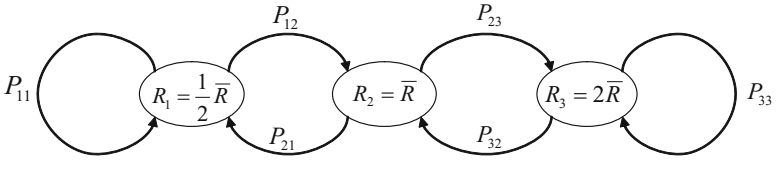


Fig. 2. Wireless channel model

For our DTMC channel model in Fig. 2, we have the transition probability matrix

$$\mathbf{P} = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix}. \tag{3}$$

Summation of each row should be one. Kolmogorov equation $\pi \cdot \mathbf{P} = \pi$ can be obtained, and the stationary distribution $\pi = [\pi_1, \pi_2, \pi_3]$ can be also computed with the normalized condition $\pi \cdot \mathbf{e}^T = 1$.

$$\pi = \mathbf{e} \cdot (\mathbf{P} + \mathbf{E} - \mathbf{I})^{-1} \tag{4}$$

where $\mathbf{e} = [111], \mathbf{E} = \begin{pmatrix} 1\,1\,1 \\ 1\,1\,1 \\ 1\,1\,1 \end{pmatrix}, \mathbf{I} = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{pmatrix}$.

3 Overload Control

3.1 Problem Statements

For real-time multimedia traffic, EDD considers only the deadlines of the packets. This algorithm can process the flows of poor channel conditions. The performance of a delay bounded scheduler is closely related to the system throughput. Lower transmission rate causes more dropped packets whose deadlines expire, increasing the overall outage probability. M-LWDF, which considers the transmission rate, may have a higher throughput than EDD.

For a scheduler to guarantee a certain delay bound, an efficient CAC scheme is typically assumed to be present. However, in the wireless networks, CAC cannot perform perfectly due to the channel variations. With many users of high channel quality, CAC may cause over-admission, increasing the outage flows.

To identify how many flows can be admitted at the CAC process, we suppose that CAC considers the channel state only when the call arrives. Since there can be three kinds of channel state in Fig. 2, a three-dimensional Markov chain can be constructed for CAC system such as the one in Fig. 3. This Markov CAC model can be used to calculate the blocking probability. The number of connection requests for each state is related to the stationary channel distribution, which is calculated in Section 2. To model the CAC system with Markov chain, analysis is carried out under the following assumptions. The connection requests are modeled as independent Poisson processes and channel holding time is assumed to be exponentially distributed.

From the Markov chain in Fig. 3, we generate the station transition matrix and compute the steady state probability. The state space can be denoted as $S = \{(k_3, k_2, k_1) | 0 \le k_1, 0 \le k_2, 0 \le k_3, 0 \le \frac{k_1}{R_1} + \frac{k_2}{R_2} + \frac{k_3}{R_3} \le C\}$, where k_1, k_2, and k_3 represent the number of flows with each channel state, and C is the total capacity. If we assume infinite number of channels in this system, the solution for $p(k_1, k_2, k_3)$ in Fig. 3 can be computed as

$$p(k_1, k_2, k_3) = \frac{\rho_1^{k_1}}{k_1!} e^{-\rho_1} \cdot \frac{\rho_2^{k_2}}{k_2!} e^{-\rho_2} \cdot \frac{\rho_3^{k_3}}{k_3!} e^{-\rho_3} \tag{5}$$

where $\rho_i = l_i / m_i, i = 1, 2, 3$.

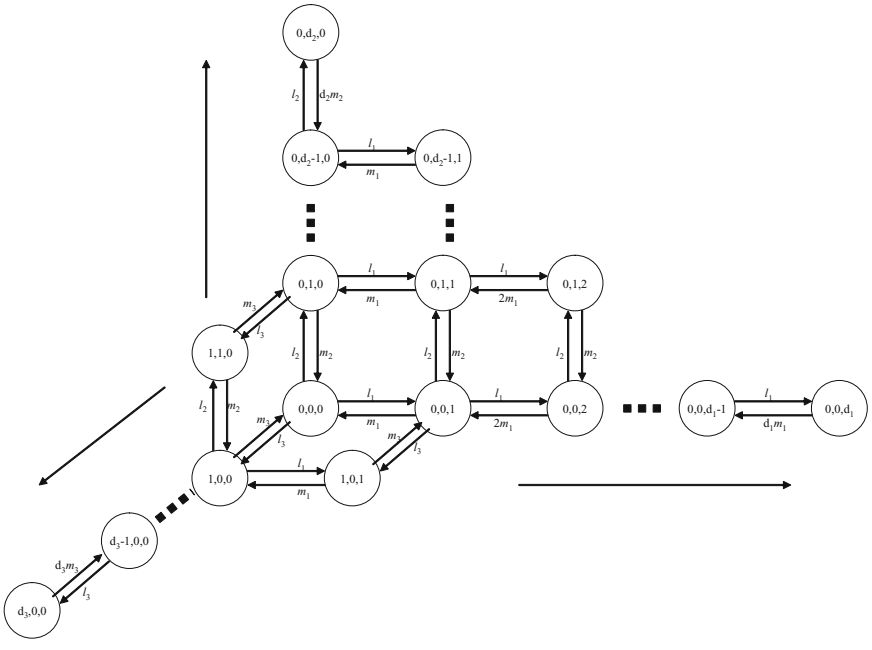


Fig. 3. CAC Markov cain with three dimension

Since the system capacity is limited to C, we have the steady state probability as a truncated result.

$$p'(k_1, k_2, k_3) = \frac{p(k_1, k_2, k_3)}{\sum_{(j_1, j_2, j_3) \in S} p(j_1, j_2, j_3)} \tag{6}$$

To apply the Markov CAC model, we translate a delay requirement into a bandwidth requirement. When the input traffic is assumed as a poisson arrival with λ_i, and the packet length is assumed to be exponentially distributed, we can apply the M/M/1 queueing model [8]. The least service rate μ_i that satisfies (1) can be derived from the distribution of waiting time.

$$Pr\{W_i > T_i\} = exp\{-(\mu_i - \lambda_i)T_i\} \leq \delta_i, \tag{7}$$

$$\mu_i \geq \lambda_i - (log\delta_i)/T_i. \tag{8}$$

For the system stability, the admission criteria can be given as

$$\sum_i \mu_i / R_i \leq 1 \tag{9}$$

where R_i is the transmission rate for user i and μ_i is the variable bit-rate when the multimedia traffic is supplied. Note that R_i can vary with time. μ_i/R_i is

the proportion of the resource, which is allocated to user i. With a time-slotted transmission, this parameter designates a time fraction for user i. When the summation of each user's time fraction exceeds one, the system becomes congested.

Flows failing in the QoS requirements can be regarded as call dropping. To reduce the call dropping from channel variations, the system should not be fully loaded to its capacity at the CAC stage. Even in that case, outage may not occur if the admission control considers the worst case as the lowest transmission rate. However, admission control usually proceeds with current information of the resource and the channel state. Therefore, when the system becomes congested, the delay bounded scheduler may not work properly. Fig. 4 shows that the bandwidth limitation leads to the decrease of Erlang capacity.

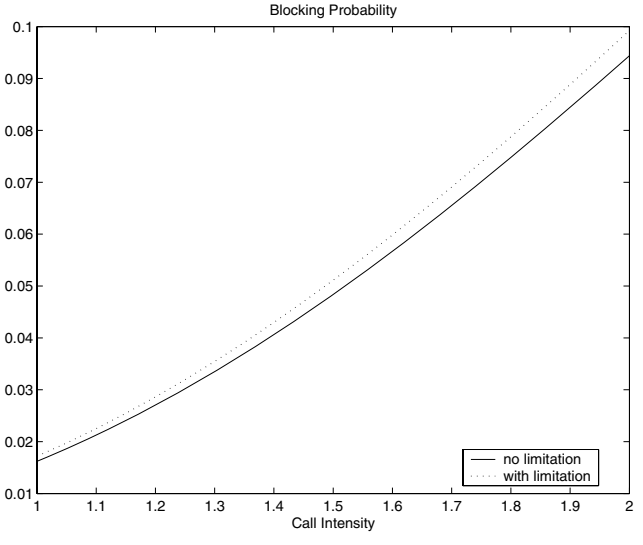


Fig. 4. Blocking Probability

3.2 Proposed Solution

We propose a removal algorithm that minimizes the outage flows for M-LWDF scheduler. In Fig. 5, it is visualized that as the regained resource after a removal is assigned to other flows, the guaranteed flows satisfying the target QoS increase. Our proposed removal rule is as follows. The number of packets in flow i during (t_0, t) is denoted as n_i, then the dropped proportion of packets is

$$DP_i(t) = \frac{1}{n_i} \sum_{k=1}^{n_i} 1_{W_i > T_i} \tag{10}$$

where 1_A is an indicator function, which is 1 when A is true and 0 otherwise. The time window (t_0, t) is the interval during which the number of admitted

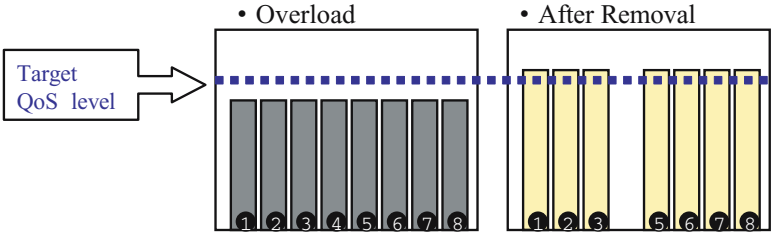


Fig. 5. QoS level after removal

flows are maintained. The iteration will restart when a new call arrives or an existing call departs. Our removal algorithm selects the priority index:

$$i = argmax_i\{DP_i(t) - \delta_i\} \tag{11}$$

$DP_i(t) - \delta_i$ is the outage performance of flow i. If the number of outage flows is zero or one, i.e., $DP_i(t)$ is less than δ_i, the removal will not occur.

Since the flows which experience large numbers of dropped packets require more resource to meet the target QoS level, the removal rule in (7) corresponds to dropping the call that has the largest-outage. With this procedure, we remove the selected connections until all remaining connections satisfy the delay constraints.

4 Numerical Example

To confirm the validity of this algorithm, we run a series of simulations with OPNET [9]. We consider a cell containing 8 users uniformly distributed throughout the cell. We have a set of available rates and the channel conditions vary with time. For each user, a distinct DTMC fading channel model was used, as shown in Fig. 2. The traffic for each user is modeled as a poisson arrival and the length of each packet is exponentially distributed. The scheduling interval is short enough to track rapid channel variations. We assume that there can be only one packet transmission at a time instant.

We compare our scheduler algorithm with FEDD and M-LWDF, as shown in Figs. 6 and 7. We put the limitation on the capacity by the numbers of flows. Fig. 6 shows that our removal algorithm has the least proportion of outage flows among the three schedulers. With a measurement-based CAC and the highest channel quality, the admitted flows can range from zero to eight. When more than four connections are admitted, outage increases due to the time-varying channel conditions. In Fig. 7, the proportion of outage packets rapidly increases in our scheme with arriving packets for the dropped flows. The removal algorithm tends to reduce the number of outage flows rather than the outage probability of each packet. The proposed removal algorithm outperforms the others in the guaranteed flows.

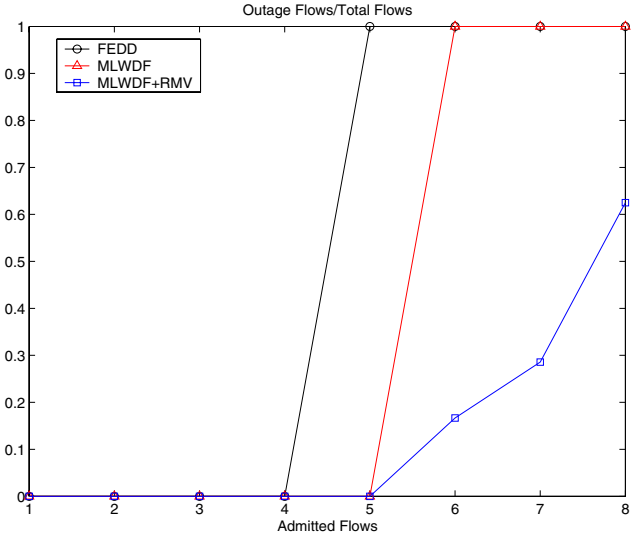


Fig. 6. Performance comparison (outage flows)

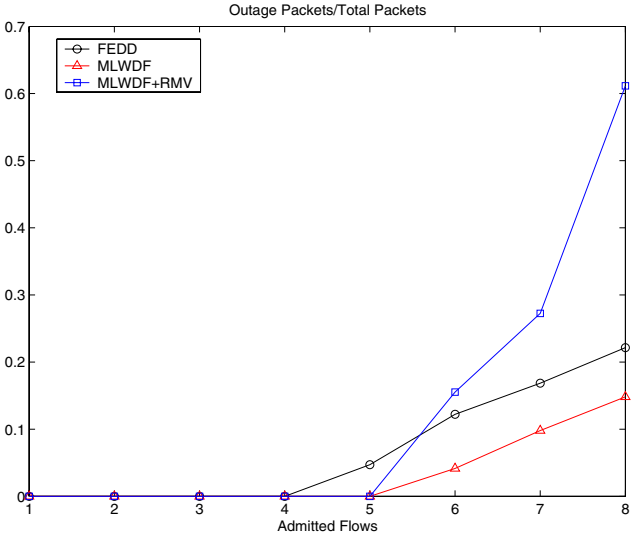


Fig. 7. Performance comparison (outage packets)

5 Conclusion

In this paper, we presented a removal algorithm for M-LWDF, focusing on the reduction of the outage flows under overloaded situations. It was shown that the proposed algorithm excelled in terms of the guaranteed flows. A natural

extension of this work will be the seamless integration of the proposed scheduler and CAC.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication) of Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) (IITA-2005-C1090-0502-0012).

References

1. Domenico Ferrari and Dinesh C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368-379, Apr. 1990.
2. Shiao-Li Tsao, "Extending Earliest-Due-Date Scheduling Algorithms for Wireless Networks with Location-Dependent Errors," in *Proc. IEEE Vehicular Technology Conf. (VTC-Fall 2000)*, Boston, USA, pp.223-228.
3. Sanjai Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *ACM/Baltzer Wireless Networks*, vol. 8 no.1, pp. 13-26, 2002.
4. Peng-Yong Kong and Keng-Hoe Teh, "Performance of Proactive Earliest Due Date Packet Scheduling in Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 53, no.4, pp.1224-1234, July 2004.
5. Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Phil Whiting, "Providing Quality of Service over a Shared Wireless Link," *IEEE Communications Magazine*, pp. 150-154, Feb. 2001.
6. Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, "CDMA Data QoS Scheduling on the Forward Link with Variable Channel Conditions," *Bell Labs Tech. Memo.*, Apr. 2000.
7. Paul Bender, Peter Black, Matthew Grob, Roberto Padovani, Nagabhushana Sindhushayana, and Andrew Viterbi, "CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users," *IEEE Communications Magazine*, pp. 70-77, July 2000.
8. Leonard Kleinrock, *Queueuing Systems, Volume I; Theory*, Wiley Interscience Publication, 1975.
9. OPNET, Available from: <http://www.opnet.com>

Sink Tree-Based Bandwidth Allocation for Scalable QoS Flow Set-Up

James Lembke[1,⋆] and Byung Kyu Choi[2]

[1] IBM, Rocester MN, USA
{jalembke, bkchoi}@mtu.edu
[2] Michigan Technological University
Department of Computer Science
Houghton, MI 49931, USA

Abstract. Although the Differentiated Services architecture supports scalable packet forwarding based on aggregate flows, the detailed procedure of Quality of Service (QoS) flow set-up within this architecture has not been well established. In this paper we explore the possibility of a scalable QoS flow set-up using a sink-tree paradigm. The paradigm constructs a sink tree at each egress edge router using network topology and bandwidth information provided by a QoS extended version of Open Shortest Path First (OSPF). Simulation results are very encouraging in that our methodology requires significantly less communication overhead in setting up QoS flows compared to the traditional per-flow signaling-based methodology while still maintaining high resource utilization.

1 Introduction

Although QoS can be delivered in many ways, two approaches have been recognized as representative solutions: Integrated Services (*IntServ*) [4] and Differentiated Services (*DiffServ*) [3] architectures. While both edge and core routers are supposed to participate in the QoS flow set-up procedure of the IntServ, the *Bandwidth Broker* (BB) model proposed in the DiffServ architecture is supposed to be in charge of all QoS flow set-up activities on behalf of core routers. Core routers in the DiffServ architecture, therefore, could be free of burdensome online computation of QoS routing, resource reservation, admission control, and associated signaling activities depending on the BB's capability. In order to truly support the scalability of the DiffServ architecture, the primary benefit of it, the control plane needs to be scalable as the data plane scalability has been significantly improved by aggregated packet forwarding. In this context, the BB model plays a significant role in providing scalability of the control plane in the DiffServ architecture.

The BB's activities can be classified into two categories: *intra-domain* and *inter-domain*. Investigating intra-domain scalability first may be a good approach because a scalable design of BB for intra-domain is likely to lead to scalable inter-domain activities. This paper addresses the control plane's scalability in a single domain of DiffServ architecture, therefore.

In general, two design options are imaginable: *centralized* and *distributed*. Centralized approaches could easily be a bottleneck of QoS flow set-up activities since a single

⋆ Part of this work was done when this author was in Michigan Tech. University.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 521–531, 2006.

centralized computing entity will be rather easily overwhelmed by a burst of QoS flow set-up activities. Moreover, a centralized BB could be a single point of failure. Distributed approaches, also, have its own disadvantage of information synchronization problem because poorly synchronized network resource information at distributed BBs could cause wrong admission decisions. One of the surest ways to guarantee a minimum overhead of QoS flow set-up activities, in our view, is to let each ingress edge router be able to conduct QoS routing, resource reservation, and admission control without further consulting any core routers on the selected path. We believe that this provides a minimum overhead because QoS flow set-up requests are handled only at the ingress edge routers. On the other hand, treating QoS flow set-up requests only at the ingress router raises another challenge of maintaining the same global view of network resources at all edge routers.

This paper proposes a methodology to the distributed BB design problem. The methodology is based on what we call a *sink tree paradigm*. The paradigm constructs a sink tree at each egress edge router using network topology and bandwidth information provided by a QoS extended version [1] of OSPF. In this paradigm, QoS routing, resource allocation, and admission control are all done at ingress edge routers in a way that QoS flow set-up overhead is minimized.

This paper is organized as follows. In Section 2 we briefly discuss recent work on this research problem. Section 3 describes our model and the sink tree paradigm for a scalable QoS flow set-up. The protocols and algorithms for an implementation of the solution are described in Section 4. The performance of our approach is evaluated in Section 5. Conclusions and future work are given in Section 6.

2 Related Work

Setting up QoS flows has many components such as QoS path selection, resource reservation, admission control, and associated signaling. Scalable flow set up, therefore, requires that each component is scalable. Although each component needs to be investigated for thorough understanding of the scalability problem, we limit our scope, in this paper, to the BB design aspect of all components because the BB is the entity that is supposed to be directly in charge of QoS flow set-up procedures.

Interestingly, a centralized BB model is assumed either implicitly or explicitly by most recent work on BB [7], [8], [9], [10], [13]. In other words, one logical BB per domain is assumed. Many aspects of the centralized BB model have been studied. Efficient admission control algorithms and resource management are proposed in [7]based on the virtual time reference system. Inter-domain signaling overhead is studied in [8]. Some implementations are available, for example, in [9], [13]. While most of these work consider hard resource reservation in the DiffServ architecture, there has been a continued effort for providing an efficient flow set up procedure for the DiffServ architectures within the framework of the IntServ architectures [2]. In this approach, a DiffServ-capable administrative domain is considered as a network element.

There have been some work addressing edge-limited admission control, for example, [14]. The fundamental difference between this class of work and ours is that they rely on online measurement while ours actively re-allocates resources.

Our approach is different from all existing approaches mentioned above. In our approach, the BB is completely distributed to all edge routers. Each edge router acts as an autonomous BB. Consequently, path selection, resource reservation, and admission decision are all done at edge routers without incurring any further communication with core routers.

3 Sink Tree Paradigm for the Edge-Limited Flow Set-Up Model

Model: Our approach is to distribute the BB completely to the boundary of the network. This permits every edge router to act like a BB. Importantly, since each edge router is virtually a BB, any QoS flow set-up request is completely handled at the edge router where the request arrives. By locating the BB at each edge router, we limit QoS flow set-up activities and associated signaling to edge routers. This necessitates that 1) the whole edge-to-edge path is determined at each edge router, 2) resource reservation on the whole path is done at each edge router, and, 3) admission decision is made at each edge router. Finally, we assume that the total amount of bandwidth for QoS flows, B_{QoS}, is fixed in a single domain.

Paradigm: Our solution to distributing the BB is to 1) select only one path for every possible pair of source and destination routers in a given network in a way that all paths having the same destination form a tree, with the destination as the root, 2) allocate resources to each path in an exclusive fashion, effectively partitioning the resources among trees, 3) maintain all information needed for QoS flow set-up at each edge router, and then 4) dynamically re-allocate resources according to traffic demands.

Figure 1 (a) illustrates the *sink tree paradigm*. In this example, a simple logical sink tree is given with seven routers (N1, N2, N3, N4, N5, N6, and N7). Considering the only QoS flows destined to N7, there are four ingress routers (N1, N2, N3, and N4), two core routers (N5 and N6), and one egress router N7.[1] The core routers N5 and N6 do not handle flow set-up requests. 'B' indicates the bandwidth units allocated to the link for the QoS-guaranteed flows. 'B=10' means that Edge[2] (N1, N5) can accommodate 10 QoS flows. For example, Edge (N5, N7) accommodates 30 QoS flows. 'B=30' on Edge (N5, N7) is in fact the sum of 'B=10' and 'B=20' of the descendant edges. Likewise 'B=70' is the sum of 'B=30' and 'B=40' of the descendant edges. Each ingress router, in the paradigm, keeps track of three parameters: *path* to the egress router, *bandwidth* available on the path, and the worst-case *edge-to-edge delay* of the path. When a flow set-up request arrives destined to N7, at N1 for example, the ingress router N1 can immediately make the admission decision. In other words, at each ingress router 1) a unique path for QoS routing is readily available, 2) bandwidth available on the path is readily known, and 3) the worst-case edge-to-edge delay for the path is readily available. without any further on-line computation. As another example, when Edge (N1, N5) has 10 QoS flows and Edge (N2, N5) has 20 QoS flows, a new request at either N1 or N2 can

[1] For example, for flows destined to N1, N1 will be the egress router.

[2] In this graph representation, vertexes are network nodes, and a link connecting two nodes is called *edge*. In this paper, *edge* is distinguished from *link* because we allocate resources to the *edge* of a tree not to the *link* of a network.

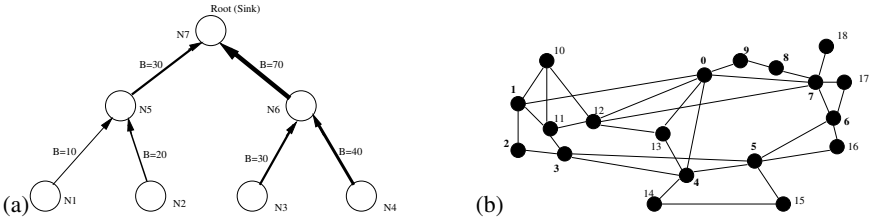


Fig. 1. An illustration of the sink tree (a) A network topology for simulation (b)

safely be rejected. In all cases, as long as flows destined to N7 are considered, N5, N6, and N7 are never involved in the flow set-up activities because the activities are limited to the ingress routers only.

Based on our analysis, we claim that as long as the sink tree paradigm holds, QoS-routing, resource-reservation, and admission-control, become trivial because 1) QoS-routing is now replaced by a simple QoS path table look-up, 2) resource-reservation is already done at off-line resource allocation time, and, 3) admission-control is done by delay table look-up of the path and the available bandwidth checking. In this paradigm, the path selection or QoS routing is done in an explicit source-routing fashion: ingress router maintains a complete list of paths between itself and all egress routers. Every router maintains a QoS routing table that is indexed by a pair of source and destination routers. Given network with a number of N edge routers, the maximum number of entries of the QoS routing table is limited by $N(N-1)$ at any router. As long as such a set of sink trees of bandwidth allocation can be found for a given network, i.e., one sink tree at each egress router, the sink tree naturally provides a methodology for edge-limited scalable QoS flow set-up.

In order for admission control to be replaced by simple delay table look-up at edge routers, all possible values of delay must be pre-calculated for all possible paths from an edge router to any destination. We developed an efficient methodology in which delay is calculated off-line for all predefined levels of bandwidth utilization at each link server on the selected path. Unless otherwise mentioned, the off-line delay computation methodology proposed in [6] is used throughout this paper.

4 Heuristic Approach to the Sink Tree Paradigm

Because finding a set of sink trees is NP-hard [12], we construct sink trees with a set of heuristic algorithms. Once a sink tree is constructed at each egress node the sink tree needs to change at run-time in order to reflect the traffic pattern change into the trees while still maintaining the sink tree paradigm. By 'reflecting' we mean that we re-allocate bandwidth among trees for better resource utilization because at run-time some trees may be congested while others are not. The topology of the tree does not change with re-allocation, however. In our approach, the re-allocation is initiated by the ingress routers. The major source of run-time overhead of the sink tree paradigm, therefore, depends on the re-allocation activities: *re-allocation decision algorithms*, and *re-allocation protocols*. Here, due to the space limit, we present an abstract of the

bandwidth (re)allocation activities. Detailed description of the algorithms and protocols can be found in [11].

Network Resource Monitoring: We make use of QOSPF [1] for network resource monitoring purposes. The link state advertisements in the QOSPF delivers crucial information that is necessary to construct sink trees. This information is topology database and network resource, i.e., *bandwidth*. We assume that all routers run a same version of QOSPF.

Initial-Allocation: For practical purposes, we assume that the initial sink tree construction is activated by a human user on a router. Once a router constructs a set of sink trees, the tree topology information will be disseminated to all routers. All routers will activate the sink trees at a designated time. For initial allocation, our methodology run the single sink tree construction algorithm (Fig. 2 (a)) for each edge router using QOSPF-generated network information. In this algorithm, a network is described by a graph G=(V,E), V = vertices, E = edges. Each router is in one of the three states: 1) *Never seen*: all routers are never seen at the beginning, 2) *Fringe*: fringes are neighbor routers of the sink tree, and 3) *Sink Tree*: eventually all routers become members of a sink tree. Initially, all routers are 'Never-seen' When the algorithm reaches a router for the first time, it becomes a 'Fringe' And then, according to the delay, it becomes eventually a member of 'Sink-tree'. The algorithm terminates when all routers are in 'Sink-tree' state. This is a variation of the MST (Minimum Spanning Tree) algorithm. We use path delay as the link cost that is computed by a methodology proposed in [6].

Bandwidth allocation: We allocate a minimum bandwidth unit to the edges of the trees so that each single path from any ingress node to any egress node has the same bandwidth allocated. This initial allocation continues until either a link is overallocated than the link capacity (or quota for QoS traffic) or the total amount of bandwidth for

```
Input: a graph of network
Output: a sink tree whose sink is v

1. T = {v}
2. Make all neighbors of v fringers
3. While T is not spanning do {
4.      For each fringe u in the order of increasing
        path delay {
5.          Include u into T
6.          Delete u from fringes
7.              For each neighbor w of u do {
8.                  If w is a fringe
9.                          Then make w a fringe with the
                            shorter e2e delay
10.                 Else if w is not in T
11.                         Make w a fringe
12.             }
13.     }
14.     If ther is no such fringe, then return the sink tree
15. }
```

Fig. 2. Single sink tree construction

```
Input: each flow arrival
Output: HELP message out

1. Whenever a flow set−up request comes at an ingress router {
2.      If the available badnwidth is below Threshold A {
3.              If ((current time − last time when HELP sent out) >
                    > HELP_interval) {
4.                      Calculate the amount of resource to be added
5.                      Send out HELP to all ingress routers
6.                      Set timer for PLEDGE
7.              }
8.      }
9. }

10. When the timer expires {
11.     If((HELP_interval + HELP_interval * M) < Upper_limit)
12.             HELP_interval = HELP_interval + HELP_interval * M
13. }

14. Whenever a PLEDGE arrives at the router {
15.     If the corresponding timer is not expired
16.             Reset_timer
17.     Update corresponding PLEDGE list
18.     If (the biggest PLEDGE > minimum of resource to be added) {
19.             Send REQUEST to the biggest  pledger
20.             If((HELP_interval − HELP_interval * F) > 0)
21.                     HELP_interval = HELP_interval − HELP_interval * F
22.     }
23. }

24. When the COMMIT comes {
25.     Add the resource to the path
26. }
```

(a)

```
Input: each HELP arrival
Output: PLEDGE message out

1. Whenever a HELP comes at an ingress router {
2.      If the available bandwidth is above Threshold R {
3.              Reply with PLEDGE
4.      }
5. }

6. When a request comes {
7.      Deduct resource from the path
8.      Reply with COMMIT
9. }
```

(b)

Fig. 3. Algorithm HELP! (a) and PLEDGE (b)

QoS flows, B_{QoS}, is exhausted. After the initial bandwidth allocation, a valid set of the sink trees is constructed. A tree is said valid if it supports the sink tree paradigm.

Re-Allocation: Re-Allocation means that when traffic congestion occurs, the sink tree under congestion tries to add more resources to accommodate more input traffic. Since the total amount of bandwidth for QoS flows is fixed in the model, the bandwidth increase on a sink tree necessitates a corresponding bandwidth decrease from another sink tree. This procedure requires message exchange between routers for the mutual agreement of resource addition at one tree and corresponding deduction at the other tree. We consider that this *communication overhead* is the major source of the overhead of using the sink tree paradigm. We use, in this paper, a simple approach for the resource reallocation negotiation. As can be seen in Fig. 3 (a), whenever an edge router receives a flow set-up request it checks the bandwidth availability. If the bandwidth usage reaches a threshold, it sends resource request message to all other edge routers. Other edge routers either pledge or do not reply depending on their own bandwidth availability as seen in Figure 3 (b). The resource requester selects the first PLEDGE message which pledges enough resource for the requester. And then an acknowledge message and the result will be sent back to the selected pledger and the other edge routers respectively. One concern in this simple negotiation procedure is a possibility of request message explosion when the entire system is overloaded. In order to avoid this situation, we use two paramaters M and F that control the minimum time interval for any two consecutive request messages originated from a same node. A PLEDGE decreases the interval while a timeout increases it (upto an upper limit).

5 Performance Evaluation

In this section, we evaluate the performance of the sink tree paradigm by simulation in NS-2. The performance is measured in terms of *admission probability* and *communication overhead* with and without artificial traffic *congestion*. Figure 1 (b) shows the typical topology of the ISP backbone network in the US. This topology is used throughout this section. Here, all routers act as edge and core routers. This means that the flows arrive at and leave from every router. For simplicity, each link is assumed to have 155 Mbps capacity. We consider two classes of traffic in this study: one QoS class and one non-QoS class. Non-QoS class traffic packets are not starved in favor of QoS class traffic as long as the amount of reserved bandwidth for QoS traffic on each link is less than the link capacity. We assume that all flows in the QoS class have a fixed packet length of 640 bits (RTP, UDP, IP headers and two voice frames), and a flow rate of 32 Kbps. The end-to-end delay requirement of all flows is assumed fixed at 100ms. The admission control behavior is simulated by flow requests and establishments at varying rates with a constant average flow lifetime. Requests for flow establishment form a Poisson process with rate λ, while flow lifetimes are exponentially distributed with an average lifetime of 180 seconds for each flow.

Figure 4 (a) shows the performance of the two resource allocations in a special case of randomly distributed input traffic: Flat and Static-Sinktree. In the *Flat*, bandwidths are evenly allocated on all links while bandwidths are allocated to sink trees in the *Static-Sinktree*. Both of them, however, do not re-allocate resources online. The total amount of bandwidths allocated on the two systems remains the same for fair comparison. In the *Static-Sinktree*, *init_bw_units_per_hop* = 40 means that initially 40 bandwidth units have been allocated to every pair of source and destination nodes. Statistically speaking, because a link can have many sink tree paths, this allocation means that roughly 10% of each link capacity is reserved on average for QoS flows. Throughout this section, one *bandwidth unit* represents the amount of bandwidth needed to support a QoS flow of 32Kbps as defined above. By 'randomly distributed' we mean that the source and destination nodes are randomly chosen for each QoS flow.

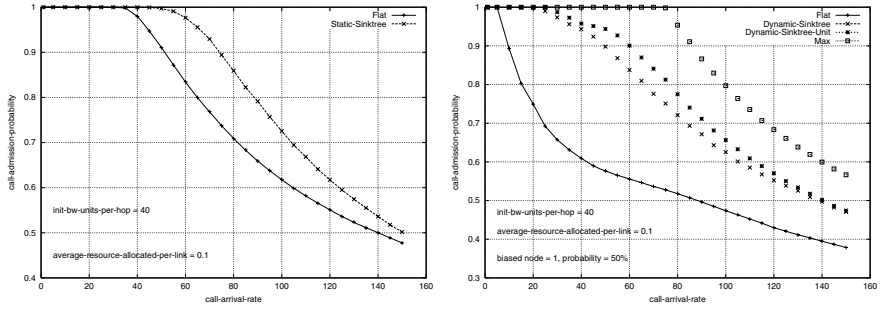


Fig. 4. Call Admission Probabilities with randomly distributed input traffic (without bias: left) and with biased input traffic (right)

A QoS flow request comes with a pair of source and destination nodes. In the Flat, a shortest path between a pair of nodes is always selected while the path is predefined with sink trees in the Static-Sinktree. Interestingly, as seen in the figure, the Static-Sinktree performs better than the Flat even though bandwidths are partitioned. In fact, this is predictable because the bandwidths are blindly allocated at a same rate on all the links in the Flat while more bandwidths are allocated on the paths where more requests are expected to come in the Static-Sinktree.

Following experiments show the performance of the sink tree-based approach under general situations where input requests are not randomly distributed and some paths are congested. In order to generate a congestion of flow set-up requests on a set of specific paths, we bias the probability with which flow set-up requests select a specific destination *node*. Here, we use Node 1 as the artificially biased destination node. A congestion probability 50% means that 50% of all flow set-up requests have Node 1 as their destination[3]. In this experiment, Sink-tree 1[4] receives 50% of flow set-up requests while the other 18 sink trees receive the other 50% of flow set-up requests altogether.

Figure 4 (b) shows the performance of the four systems with an artificial traffic congestion: Flat, Dynamic-Sinktree, Dynamic-Sinktree-Unit, and Max. The *Flat* is the same system as in Figure 4 that does not re-allocate bandwidths online. The *Dynamic-Sinktree* is a sink tree-based system that uses a proactive resource re-allocation as described in the previous section. In this experiment, whenever available bandwidths at an edge node are less than 10% of the initial allocation, a re-allocation of 10% of the initial allocation is tried. When a request arrives at an edge node which has no available bandwidth for the request, the request is rejected without causing any further re-allocation attempt. The *Dynamic-Sinktree-Unit* is different from the *Dynamic-Sinktree* in that it tries a re-allocation in a reactive fashion when an edge router has no bandwidth to honor a request. In addition, it tries to re-allocate only one bandwidth unit for the path requested. Consequently, this violates the sink tree paradigm because a request can cause further control activities in the network core. We consider this as the upper bound of the resource utilization of the sink tree-based system because it re-allocates bandwidths reactively based on requests. The *Max* is a theoretical system that has only one centralized BB in the network. We assume that the BB in the Max system has mighty power to monitor and re-allocate bandwidths online. All control activities including path selection, bandwidth allocation, admission decision, and associated signaling activities are done by the centralized BB. In this system, a request chooses a shortest path from the source to the destination node. The BB maintains the bandwidth pool for the entire network. So, a request is admitted as long as there is bandwidth available in the pool and the bandwidth allocation does not violate the constraint of the link capacity of the selected path. Therefore, the bandwidths are most flexibly shared in this system. We consider this as the upper bound of the resource utilization of any resource allocation. As seen in the figure the Dynamic-Sinktree performs close to the Dynamic-Sinktree-Unit. The small difference between these two is considered as the overhead of the re-allocation algorithm because the re-allocation is done based on a prediction of

[3] According to a previous study on the congestion pattern, 90% of the Internet traffic is destined to 10% of the networks [5].

[4] Sink-tree 2 is the sink tree whose sink node is Node 2.

future demand. The gap between the Dynamic-Sinktree-Unit and the Max is considered the sink tree paradigm-overhead in order to support the edge-limited control activities.

Figure 5 (a) reveals the admission probabilities of Sink-Tree 1, the hot spot of the simulation system. Assuming that the traffic in the hot spot is of the primary interest, this shows more clearly the benefit (the gain of admission probability) of using the sink tree paradigm, an efficient resource re-allocation.

Figure 5 (b) shows the communication overhead of the four systems in a log scale in terms of number of messages exchanged: RSVP-Style, Max, Dynamic-Sinktree-Unit, and Dynamic-Sinktree. The communication overhead of the *Dynamic-Sinktree* is the number of messages (HELP and PLEDGE) generated for resource re-allocation. The parameters for HELP message interval control are $M = 1$ and $F = 0.1$ (Figure 3). The *Dynamic-Sinktree-Unit* generates the same messages when a request comes to an edge which has no available bandwidth for the request. In the *RSVP-Style*, the number of hops from the source node to the destination node for admitted flows was counted in addition to the regular refreshment of every 30 seconds. In the *Max*, the number of hops from the source node to the imaginary centralized BB, Node 1, in this experiment, was counted for all attempted flows. Different locations of the centralized BB produce almost the same results. As can be seen, the Dynamic-Sinktree produces far less messages

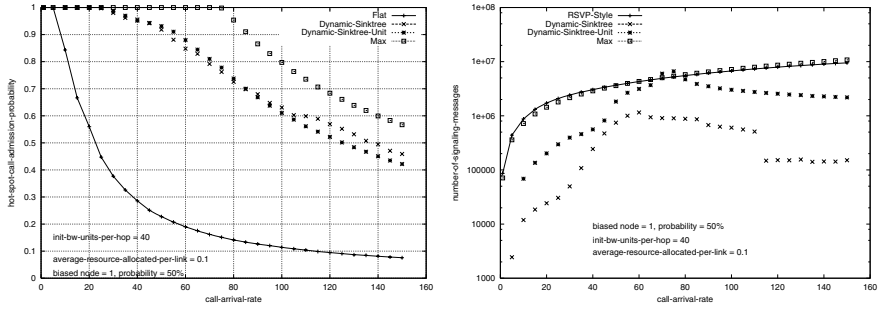


Fig. 5. Call Admission Probabilities of Hot Spot Tree (left) and Signaling Messages (right) of biased input traffic

than either the RSVP-Style or the Max. The Dynamic-Sinktree goes to the peak at λ 60 and remains almost flat for the rest of the experiment. This is because the HELP message generation is suppressed by Algorithm HELP!. The big drop at λ 110 suggests that more of the re-allocation attempts started failing at that point. The Dynamic-Sinktree-Unit produces a lot more messages than the Dynamic-Sinktree because the re-allocation is attempted with only one unit on the requested path. Interestingly, the Max produces almost the same amount of communication overhead as the RSVP-Style. This is because in the Max system any flow request travels from the source to the BB and back to the source. Therefore, as the number of the rejected requests is growing, the communication overhead is linearly increasing in proportion to the volume of the attempts. In the RSVP-Style, although it uses a refresh every 30 seconds, because all nodes participate in the admission decision, any node can reject a request before the request reaches to the

final destination node. In terms of communication overhead, therefore, it is very likely that centralized BB approaches will produce about the same amount of messages as the RSVP-Style. Figure 5 could be misleading as it does not show when the messages are produced. All the three systems except the Dynamic-Sinktree produce messages when a request arrives while the Dynamic-Sinktree does it proactively (see Figure 3). The HELP and PLEDGE messages are produced proactively when the resource consumption reaches to a threshold. As a matter of fact, although our survey is not exhausted, we have not found a centralized BB approach that utilizes bandwidths to the most flexible way as the Max.

6 Conclusions and Future Work

In this paper we explored possibilities of scalable QoS flow set-up by using the sink tree paradigm within the DiffServ architecture. Simulation results are very encouraging in that the paradigm requires far less communication overhead in setting up QoS flows compared to both the traditional per-flow signaling-based methodology and the imaginary centralized BB system while still maintaining high resource utilization by run-time resource re-allocation. Based on these encouraging results we are further investigating the scalable QoS flow set-up problem for the end-to-end multi-domain environment.

References

1. Apostolopoulos, G., Guerin, R., Kamat, S.: Implementation and performance measurements of QoS routing extensions to OSPF. IEEE INFOCOM. (1999) 680-688
2. Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., Felstaine, E.: A Framework for Integrated Services Operation over Differentiated Networks. IETF RFC 2998. (2000)
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Service. IETF RFC 2475. (1998)
4. Braden, R., Clark, D., Shenker, S.: Integrated Services in the Internet Architecture. IETF RFC 1633. (1994)
5. Chen, J., Druschel, P., Subramanian, D.: A New Approach to Routing with Dynamic Metrics. IEEE INFOCOM, (1999) 661-670
6. Choi, B., Xuan, D., Li, C., Bettati, R., Zhao, W.: Utilization-Based Admission Control for Scalable Real-Time Communication. Real-Time Systems, Vol. 24. Kluwer, Is. 2, (2003) 171-202
7. Duan, Z., Zhang, Y. Z.-L., Hou, T., Gao, L.: A core stateless bandwidth broker architecture for scalable support of guaranteed services. IEEE Transactions on Parallel and Distributed Systems, Vol. 15, Is. 2, (2004) 167-182
8. Gunter, M., Braun, T.: Evaluation of bandwidth broker signaling. The 7th IEEE International Conference on Network Protocols (ICNP) (1999) 145-152
9. Hwang, J., S. Chapin, S., Mantar, H., Okumus, I.: An implementation study of a dynamic inter-domain bandwidth management platform in DiffServ networks. IEEE/IFIP NOMS (2004) 321-334
10. Lee, B.-S., Woo, W.-K., Yeo, C.-K., Lim, T., Lim, B.-H., He, Y., Song, J.: Secure communications between bandwidth brokers. ACM SIGOPS Operating Systems Review, Vol. 38, Is. 1, (2004) 43-57

11. Lembke, J.: Simulation of Sink Tree Paradigm for Quality of Service Provisioning in Computer Networks in NS-2. MS Project Report, Department of Computer Science, Michigan Technological University (2005) (http://www.cs.mtu.edu/∼bkchoi/Lembke05.pdf)
12. Papadimitriou, C. H.: The complexity of the capacitated tree problem. Technical Report No. TR-21-76, Department of Computer Science, Cambridge, MA, Harvard University, (1976)
13. Teitelbaum, B., Hares, S., Dunn, L., Nielson, R., Narayan, V., Reichmeyer, F.: Internet2 QBone: building a testbed for differentiated services. IEEE Network Mag., Vol. 13, Is. 5. (1999) 8-16
14. Yuan, P., Schlembach, J., Skoe, A., Knightly, E. W.: Design and implementation of scalable edge-based admission control. Computer Networks, Elsevier, Vol. 37, No. 5. (2001) 507-518

A QoS-Based Adaptive Resource Sharing Protection for Optical Burst Switching Networks

Hyunsu Lim[1], Sang-il Ahn[1], Eun-kyou Kim[1], and Hong-Shik Park[2]

[1] Korea Aerospace Research Institute, Eoeun-dong, Daejon, (305-333), Korea(ROK)
{hyunsu, siahn, ekkim}@kari.re.kr
[2] Information and Communications University, 119, Munjiro, Yuseong-Gu, Daejon, (305-714), Korea(ROK)
hspark@icu.ac.kr

Abstract. Protection and restoration are essential mechanisms for guaranteeing more reliable traffic delivery services. But it is not easy to apply existing mechanisms to optical burst switching (OBS) networks due to its one-way reservation signaling and the statistical burst multiplexing. Thus, to achieve the high transmission performance and reliability simultaneously, unique properties of OBS must be considered in the design of protection scheme. In this paper, we introduce a new 1:1 link-based OBS protection with several control messages. It minimizes burst losses by deflecting bursts until the source edge router arbitrates a working burst path to a backup path when a link failure occurs. Based on this, we propose a genuine dynamic resource sharing (DRS) protection algorithm. It optimizes the number of provisioned protection wavelengths adaptively based on the traffic load as well as the quality of service (QoS) requirements of bursts in near real-time. In addition, DRS can be used as the temporary short-term contention resolution method. The simulation results verify that the proposed schemes improve the network resource sharing and backup link's channel utilization while guaranteeing the targeted protection reliability and QoS requirements of class bursts.

1 Introduction

The rapid evolution of wavelength division multiplexing (WDM) and optical devices technology has increased the implementation possibility for supporting cost-effective integrated services through optical networks [1],[2]. But, due to the high transmission speed, a single link failure during short time may result in large data losses, especially when it occurs in the backbone transmission networks. Thus, the high survivability to withstand and recover types of failures is an essential requirement in all kinds of optical networks. In OBS networks, protection and restoration are more performance critical issues because data bursts are transmitted based on the one-way path reservation of ingress edge routers.

Until now, a few papers have dealt with protection and restoration mechanisms in OBS networks [3],[4]. The authors of [3] validated that their proposed recovery scheme has the high scalability to cover a wavelength failure, a link failure, and a node failure as well. But their mechanism applied traditional circuit-based protection and restoration to OBS networks without considering burst

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 532–541, 2006.

transmission properties. It nearly considered unique property of OBS, the offset time-based reservation which makes the statistical multiplexing possible. In OBS, especially in the common just enough time (JET) scheme, the change of offset time in core routers results in the large and continual burst losses because the offset-time, which is determined based on the conceived an end-to-end path topology and the network load distribution information of edge router, is fixed at the ingress edge routers. Thus, the burst recovery operation in the small district may result in consecutive burst losses in other bursts' working paths and links due to increased reservation contentions.

The authors of [4] proposed the revised version of protection signaling but it did not mention how to manage the OBS's source-based routing. For achieving a more effective protection, the unique channel reservation of OBS by which a control packet is transmitted and reserves a channel by the amount of burst in advance must be considered. The offset time is determined at the ingress node and it can be roughly calculated as $\delta * n + \delta_{QoS}$ where δ, n, and δ_{QoS} are the electric control packet processing time in each intermediate node, the number of hops which burst passes through, and the QoS offset time which is determined by the traffic engineering. Hence if unexpected changes occur, bursts may be discarded eventually until its path is changed although the bursts can be deflected temporarily. Therefore, protection and restoration must be based on the link-protection to minimize burst losses but they must be also cooperated with the path-protection simultaneously. At the same time, for achieving high throughput, the network resources which are reserved as the protection links must be optimized. These can be achieved by the offset-time based statistical multiplexing of bursts in OBS.

In this paper, we propose a new traffic load-based dynamic protection link reservation scheme for OBS networks: DRS (Dynamic Resource Shared) protection. It optimizes the number of protection wavelengths adaptively based on the traffic load and the QoS requirements in near real-time. The proposed DRS scheme improves the network resource sharing as well as the backup link's channel utilization while guaranteeing high protection reliability and QoS requirements. The remainder of paper is as follows. In Sect.2, we introduce a 1:1 link-based protection scheme and control message signaling methods. In Sect.3, we propose the genuine DRS protection algorithm and design issues. We validate performance of 1:1 link-based DRS protection in Sect.4 and Sect.5 concludes the paper with some remarks.

2 1:1 Link-Based Protection Scheme on OBS Networks

OBS is expected as the alternative optical switching technology to efficiently transmit big size bursts with high network throughput. Because the burst transmission is controlled by the source routed path selection and one-way reservation, intermediate nodes do not change the determined path [2]. Thus, a path-protection can not guarantee the high survivability by itself in general non FDL-OBS networks. If the path-protection is applied alone and a link failure

occurs, numerous bursts are dropped until that failure is notified to source edges and bursts are deflected to backup paths. Hence it is inevitable to apply the link-protection in the case of OBS networks. Of course, the 1+1 path- and link-protection can resolve these problems but its link utilization is too low and complex burst-fault monitoring mechanism must be supported. In this paper, we target to achieve optimized resource reservation in the 1:1 link-protection mechanism.

Our proposed 1:1 link protection is based on a periodic signaling with following control messages:

- Failure detection - Liveness message: Exchanges periodically between two adjacent nodes to check the link state (detect faults).
- Restoration Request (Failure notification)- Restoration Request message: Downstream node sends it to the upstream node to request the link recovery procedure.
- Restoration Confirm message: Upstream node sends it to the downstream node to confirm requested restoration.
- Failure Advertisement message - Upstream node notifies the link failure to all edge OBS routers.

Based on these types of messages, recovery operations are done sequentially by the three step handshaking as shown in Fig.1. We assume that a link between nodes is bi-directional with a pair of uni-directional fibers on opposite directions. In this paper, we only consider a uni-directional link failure, single fiber cut. To avoid the optical-electric-optical (O/E/O) conversion processing, signaling messages are transmitted by the control channel group (CCG) of working link, which are exclusively allocated for control packet transmission and all packets are O/E/O conversed at every nodes. The recovery operation is as follows.

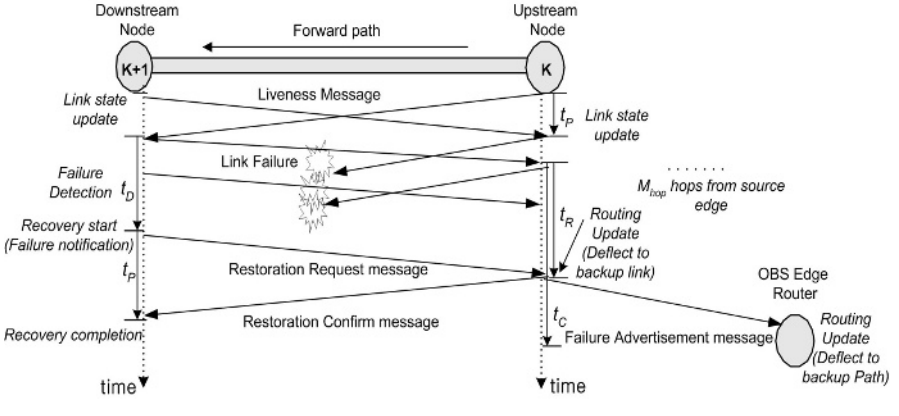


Fig. 1. Logical operation of 1:1 link protection

1. *Monitoring*: In normal operations, the liveness messages are periodically transmitted to each nodes of link in every update interval, t_P, to monitor the link state. We consider the forward direction link failure case, from the upstream node (node_K) to the downstream node node_{K+1}.
2. *Link-Failure Detection*: When a link-failure happens, the liveness message transmitted by node_K is not delivered to node_{K+1}. But node_{K+1} waits for the liveness message during the fault detection time (t_D), which is the N times of liveness message update interval, in order to avoid a wrong restoration due to the temporary link congestion or load fluctuation. After t_D, the downstream node node_{K+1} goes into the link restoration state and transmits a recovery request message to the node_K by the reverse path's CCG, which is still working.
3. *Restoration and Deflection Routing*: When the recovery request message arrives, node_K conceives the link failure and it goes into the restoration state. First of all, it deflects bursts of working link, which is a group of working wavelengths (channels), to the backup link's channels to minimize short-term burst losses. The selection of backup link and the number of backup channels are determined in advance in the manner of 1:1 or m:n scheme. Then node_K confirms the recovery request by sending the restoration response message to node_{K+1}. At the same time, it also notifies a link failure to the every source edge routers. Thus, the logical network topology of edge OBS routers are changed and bursts are transmitted via deflected paths, which are also determined in advance, in order to avoid failed link.

Therefore the effectiveness of restoration and its survivability are directly affected by the fault detection time (t_D), the link restoration time (t_R), and the monitoring interval (t_P). In the network domain, performances of end-to-end packet transmission are determined by the path restoration time (t_C) which requires for edge routers to deflect bursts from working path to the backup path. Thus, when the average traffic arrival rate is λ, the amount of burst losses is derived intuitively as follows.

$$B_{loss} = \lambda \cdot t_R = \lambda((N-1)t_P + 0.5t_P) \geq \lambda \cdot t_{prop} \tag{1}$$

where t_{prop} is the propagation delay of link. These losses are inevitable in the 1:1 link-protection scheme although the backup link is reserved in advance. Generally, the number of hops M_{hop} and t_C affect the performances of restoration because bursts are concentrated on the backup link until the edge node deflects bursts. Generally, t_C is much larger than t_R due to large M_{hop}, and this stands for that it is impossible to manage failures with only the path protection. To reduce short-term burst losses, N and t_P must be minimized. But this increases contentions on CCG frequently or increases the incorrectness of restoration. Thus, there must be trade-off between N and t_P. In this paper, we assume that the optimized values are determined by the network administrator and we do not touch this problem any more because it is not a major scope of research.

It is trivial that bursts under a link-failure are efficiently protected by the proposed 1:1 link-protection although some bursts are dropped. But the network resource utilization decreases proportionally. In the case of general 1:1 protection scheme for optical circuit switching, if there are N wavelengths (channels) in a working link, the upstream node reserves the same number of channels in a backup link. This is a necessary condition for the circuit switching but is not mandatary for OBS networks due to the statistical multiplexing property of OBS. If bursts can be efficiently multiplexed to the backup link by the advance control packet transmission, the number of backup channels (N_{BC}) can be smaller than the number of working link's channels ($N_{WC} = C_{wc}$). In this case, the unused channels of the backup link can be temporarily used as the working link of another path or may be statistically shared as a backup link by other links. Thus the network resource utilization can be improved meaningfully while required restoration reliability and survivability are strictly guaranteed. These are the objectives of our proposed QoS-based adaptive protection channel allocation scheme, DRS (Dynamic Resource Shared) protection.

3 DRS OBS Protection Scheme

The DRS protection determines the number of protection channels of backup link adaptively based on the traffic load and the burst's QoS requirement. It makes efficient network resource sharing possible by optimized statistical multiplexing while the 1:1 link-based protection's survivability is guaranteed. As discussed in [2], burst collision and drop might happen in OBS network. The blocking probability of class burst is determined by the link traffic load (intra-class) and the QoS offset time control method (inter-class). Therefore, if the blocking probability for bursts which are transmitted by the working link can be stably guaranteed, the N_{BC} value for accommodating working link's bursts can vary according to the change of traffic load in working link, ρ_w. It is assumed that ingress edge nodes of OBS networks categorize input data bursts into C classes and bursts are transmitted by the JET scheme with considering QoS offset. To calculate the optimized N_{BC}, upstream node of every link monitors the blocking loss rates of class bursts and the utilization of working link based on the wavelength reservation of control packets. The logical switch architecture for DRS-protection is shown in Fig.2.

The scheduler determines the switching start and stop time of bursts transmitted by wavelengths with using the offset-time information of incoming control packets. In addition, when a link failure happens, it deflects bursts of working link to backup link's restoration channels. The backup link is determined in advance. The link monitor block checks the link utilization and the blocking loss rate of bursts periodically. Finally, the DRS controller determines the optimum number of channels in the backup link, N_{BC}. We define $b_{G,k}$ and $b_{w,k}$ as the guaranteed blocking loss rate and the observed blocking loss rate of working link, respectively, where the traffic load of $class_k$ is $\rho_{w,k}$. These local blocks maintain the observed blocking loss rate metric (B_w) and the guaranteed

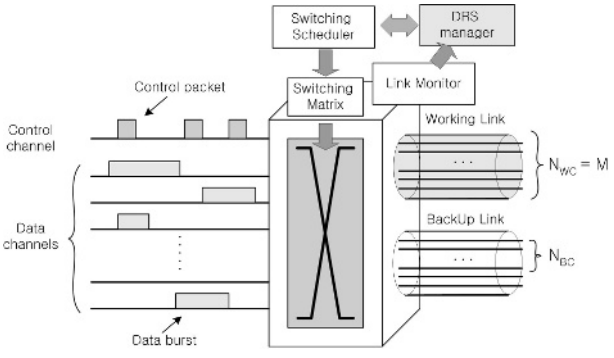


Fig. 2. Logical architecture for DRS based protection

blocking loss rate metric (B_G) of the working link. The link utilization and the target (guaranteed) link utilization of the working link are also defined as $U_w(\rho_w)$ and $U_{G,w}$. These satisfy the following operational requirements.

$$B_w(\rho_w) = \{[b_{w,1}, b_{w,2}, ..., b_{w,C}] \mid 0 \leq b_{w,1}, b_{w,2}...b_{w,C} \leq 1\} \quad (2)$$

$$B_G = \{[b_{G,1}, b_{G,2}, ..., b_{G,C}] \mid 0 \leq b_{G,1}, b_{G,2}...b_{G,C} \leq 1\} \quad (3)$$

$$U_w(\rho_w) = U(\rho_w, N_{WC}), \ \ \rho_w = \rho_{w,1} + \rho_{w,2} + ... + \rho_{w,C} \quad (4)$$

Based on this control information, the upstream node determines the optimum number of N_{BC}s and reserves channels in the backup link. In operations, the monitor block checks $B_w(\rho_w)$ and $U_w(\rho_w)$ in every T (Monitoring interval). We assume that T is same to the Link state update interval of the Liveness message transmission in 1:1 link protection (t_P). In the algorithm description, b_k^s and N_{opt} represent the burst loss rate comparison flag for class k and the optimized number of backup channels, respectively. The DRS algorithm can be firmly expressed as follows:

DRS Algorithm
input: $B_w(\rho_w)$, $U_w(\rho_w)$, and N_{opt}
output: N_{BC}
begin

S1: Compare $U_w(\rho_w)$ with $U_{G,w}$.
- If $(U_w(\rho_w) > U_{G,w})$, set $U_w^s = 0$.
- Else set $U_w^s = 1$.

Then goto S2.

S2: Compare $[b_{w,1}, b_{w,2}, ..., b_{w,C}]$ with $[b_{G,1}, b_{G,2}, ..., b_{G,C}]$.
- If $(b_{G,k} > b_{w,k})$, $0 \leq b_{w,1}, b_{w,2}...b_{w,C} \leq 1$, set $b_k^s = 0$.
- Else, set $b_k^s = 1$.

Then goto S3.

S3: Find N_{BC} with $[b_1^s, b_2^s, ..., b_k^s]$
For $0 < k < C$
- If b_k^s is 1, $0 < k < C$, N_{opt} += 1. goto S4.
- Else if b_k^s is 0 and U_w^s is 0, N_{opt} -= 1. goto S4.

- Then goto S5.

S4: Compute the logical B_w and U_w with N_{opt}.
- If ($N_{opt} < N_{WC}$), goto S1.
- Else goto S5.

S5: Set N_{opt} as the backup link value N_{BC}.
- Goto end.

end

The DRS algorithm searches the optimized number of N_{BC}s recursively until the target blocking loss rates of classes and the link utilization are achieved. Thus, the upstream node allocates the optimized N_{BC} in every T. From the fact that control packets of bursts arrive at the upstream node in advance, this multiplexed protection is generally efficient. Because the number of N_{BC}s is determined adaptively based on the traffic load of working link (ρ_w), provisioned backup channels can support the entire data bursts of working link with the same QoS level in terms of burst loss rate of class although a link failure happens suddenly. In the general load, N_{BC} is generally smaller than the number of working link's channels (M). Thus, part of backup link between $node_k$ and $node_{k+1}$ can be temporarily used as working channels or backup channels by another link which connects $node_j$ with $node_{j+1}$. Therefore, the network resource utilization increases efficiently.

In DRS, the optimized N_{BC} value is easily achieved when ρ_w is low. But, if ρ_w is too high to satisfy the target $b_{w,k}$ of classes, N_{BC} is same to the number of channels in the working link (M). Generally, this problem happens when bursts are concentrated on some links due to the synchronized path overlap of ingress edge nodes. This is a critical problem regardless of the efficiency of protection mechanism. The proposed DRS scheme can be partially applied to resolve this short-term load fluctuation. For achieving a link load balancing, if $U_w(\rho_w)$ and $b_{G,k}$ can not be guaranteed by the working link's M channels, the scheduler of upstream node can deflect the low prioritized class bursts to the reserved backup channels. As shown in [2], the burst loss rate of high priority class bursts is rarely affected by lower class bursts. Thus, if the low priority bursts can be transmitted by the backup link's channels although QoS offset-time decreases, the number of transmission channels increases up to $M + N_{BC}$ and target $U_w(\rho_w)$ and $b_{G,k}$ can be guaranteed with higher probability. Considering that a link failure does not happen frequently, it can be an efficient temporary solution until the source edge routers change the path of bursts with network engineering mechanisms. This short-term load-balancing does not affect the protection survivability under a link-failure because data bursts experience the same burst losses as those of working link.

4 Simulation Results and Performance Evaluation

In this section we validate the efficiency of DRS with simulations. For achieving more effective results, we developed the OBS node simulator by using the OPNET modeler 8.0 and partially introducing C++ for the burst traffic trace. We assume that incoming data traffic is classified into three classes, class1 (the highest priority) for aggregated real time traffic, class2 for non-real time and class3 (the lowest priority) for best effort. Traffic is aggregated into bursts in the class buffers at the ingress edge nodes of OBS network and bursts are transmitted in the manner of Offset-time based JET transmission. It is assumed that the traffic load is uniformly distributed over classes and the ON/OFF model-based Pareto distributed burst source is used [5]. The single data channel speed between nodes is assumed as 2.5 Gbps and the number of data channels of link is 16. It is assumed that no optical buffer block or fiber delay line (FDL) is introduced in the intermediate nodes.

Figure 3.(a) shows the average number of necessary N_{BC}s when the traffic load of working link changes. As can be expected, the number of backup channels for guaranteeing burst loss rates varies according to the change of load. When the load is low, N_{BC} is also small due to the multiplexing of bursts. This means that large amount of backup link channels can be shared by other links. In contrast, N_{BC} of DRS gradually converges to that of the 1:1 link-based protection as the load increases. This is a trivial result and performance of DRS is same as the 1:1 link-based protection under the high load condition. But it is meaningful that the proposed DRS guarantees the improved resource sharing for high network utilization in the general load condition.

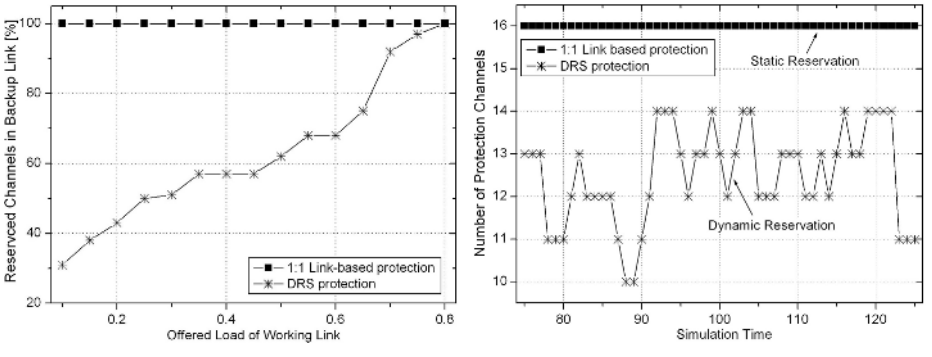


Fig. 3. Burst traffic with Hurst Parameter = 0.8, Average load = 0.4 (a) Average number of Backup Channels (b) The number of Backup Channels

Figure 3.(b) represents the number of provisioned backup channels when the long-term average traffic load is 0.4. As shown, N_{BC} of the DRS algorithm varies adaptively according to the traffic of the working link. The guaranteed burst loss

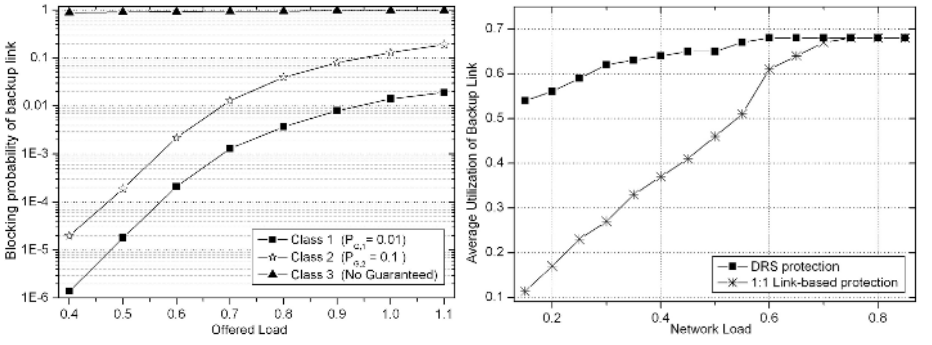


Fig. 4. (a) Block probability of backup link when failure happens. (b) Average resource utilization when failure happens. Burst traffic with Hurst Parameter = 0.8.

rate is assumed as 0.0001 for the class 1 bursts. Under the burst traffic scenario, the real incoming traffic load varies although the average load is low. But N_{BC} is fixed to the number of working link's channels in the general 1:1 link-based protection. Because those statically allocated channels can not be shared by other links, the network resource utilization decreases due to the channel overbooking although numbers of channels per link are required. In contrast, the upstream node under the DRS algorithm reserves only the optimum number of backup channels adaptively based on the current link load and the blocking rates. Thus the unreserved backup channels can be temporarily used for working or backup channels of other links.

The average blocking probability of class bursts under a link failure is shown in Fig. 4.(a). As shown, the DRS algorithm satisfies the required blocking burst loss rate of working link, but the blocking probability of class 3 is somewhat higher than that of working link in every load range. This results from the rapid change of burst traffic load. These defects are inevitable in the DRS algorithm due to dynamic operation, but those might be resolved effectively by the traffic shaping of ingress routers. Figure 4.(b) shows the average channel utilization of backup link when the unreserved channels are fully used for backup channels of other links. As compared, the channel utilization of DRS is higher than that of 1:1 protection in the normal load below 0.6. This high link channel utilization is guaranteed stably by the adaptive backup channel reservation and offset-time based statistical multiplexing. Therefore the wavelengths of network can be provisioned more efficiently according to the load distribution over links while the high resource utilization and throughput are achieved.

5 Conclusion

In this paper, we researched performances of protection and restoration mechanisms in the OBS network. To guarantee the high survivability of protection, we introduced the 1:1 link-based protection scheme and several kinds of control

messages. In addition, we proposed a new adaptive resource sharing protection scheme: DRS. By adaptively reserving backup channels based on the traffic load and QoS of bursts, proposed DRS improves the network resource sharing and the link utilization while it guarantees the QoS requirements of classes in the reserved backup link channels. In further research, the more advanced class-based backup channel reservation will be studied for the absolutely guaranteed QoS services.

Acknowledgement

This work was supported in part by MIC, Korea under the ITRC program supervised by the IITA.

References

1. C, Qiao., M, Yoo.:Optical burst switching (OBS) - A new paradigm for an Optical Internet, Jol. High Speed Network, Vol. 8, 1999, pp.69-84.
2. M, Yoo., C, Qiao., Sudhir, Dixit.:QoS Performance of Optical Burst Switching in IP-Over-WDM Networks. IEEE JSAC, Vol. 18, No. 10, Oct 2000, pp.2062-2071.
3. Seunghun, Oh., et al.:Survivability in the Optical Internet Using the Optical Burst Switch. ETRI Journal, Vol. 24, April, 2002, pp.117-130.
4. Raadim, Bartos., Swapnil Bhata.:Fast Restoration Signaling in Optical Networks. Proceeding of IASTED conference on parallel and distributed computing systems, March 2002.
5. Leland, W., Taggu, M., Willinger, W., Wilson, D.: On the Self-Similar Nature of Ethernet Traffic. IEEE/ACM Trans, Networking, Vol. 2, No. 1, (1994) pp.1-15.

Request Scheduling for Differentiated QoS at Website Gateway

Ching-Ming Tien[1], Shuo-Yen Wen[1], Ying-Dar Lin[1], and Yuan-Cheng Lai[2]

[1] Department of Computer and Information Science, National Chiao Tung University
1001, Ta Hsueh Road, Hsinchu, Taiwan 300
{cmtien, sywen, ydlin}@cis.nctu.edu.tw
[2] Department of Information Management,
National Taiwan University of Science and Technology
43, Section 4, Keelung Road, Taipei, Taiwan 106
laiyc@cs.ntust.edu.tw

Abstract. With the explosive growth of Web traffic, the load on a Web server becomes heavier, leading to the longer user-perceived latency. Website operators would like to employ service differentiation to offer better throughput and shorter user-perceived latency to some specific users. This paper presents an HTTP request scheduling algorithm deployed at the website gateway to enable the Web quality of service without any modification to client or server software. A variation of the deficit round robin scheduling algorithm and a window control mechanism are presented to decide the order and the releasing time of requests, respectively. The order is decided by the response size of the requests and the pre-defined service weights. The ratio of the bandwidth throughput of the service classes is determined by the weights, whereas the releasing time is decided by the service rate of the Web server. The evaluation reveals the scheduling algorithm can provide service differentiation and improve server throughput and user-perceived latency.

1 Introduction

Today more and more users connect to the Internet to surf the World Wide Web. The more accesses to a website, the heavier load will be on the Web server. The busier server leads to the longer user-perceived latency, which means a user will wait for a longer time to download a Web page. Therefore, website operators would like to improve user-perceived latency to keep their customers.

The bottleneck of accessing a Web page could occur on a network or server [1]. The network bottleneck could be resolved by employing network QoS (Quality of Service) mechanisms [2] [3], whereas the server bottleneck could be resolved by clustering servers, caching Web pages, and so on. However, network QoS is hard to be deployed in nowadays Internet infrastructure because all routers have to support and enable network QoS protocols, e.g. RSVP (Resource reSerVation Protocol)[4]. At the server side, the HTTP (HyperText Transfer Protocol) traffic can be controlled at the packet level or the application level. Several recent

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 542–551, 2006.

researches have proposed application-level QoS [5] [6] [7] [8] [9] [10] to provide service differentiation because this approach provides more flexible policies to website operators in traffic control. They made efforts on modifying the system kernel [5] or the daemon program [5] [6] [7] [8] [9] [10] of a Web server to provide Web QoS. However, the shortcoming is those mechanisms are operating system or server daemon dependent.

This research focuses on resolving the server bottleneck because website operators can completely control their servers, but cannot do much on improving the whole network performance. The goals are to improve the server throughput and reduce the user-perceived latency for some specific users; in other words, to provide service differentiation at the server side, and thus allows some users to perceive the shorter latency on downloading Web pages. A QoS website gateway, independent of operating systems and server daemons and transparent to clients and servers, is presented in this paper. HTTP requests incoming to the gateway will be classified and queued into different class queues by the application-level inspection. A variation of DRR (Deficit Round Robin) scheduling [11] and a window control mechanism are presented to decide the order and the releasing time of requests, respectively. In addition, a server probing mechanism is used to seize the characteristics of Web pages, such as URL (Universal Resource Locator) and the response size, to help the request scheduling.

The rest of this paper is organized as follows. Section 2 introduces the architecture of the QoS website gateway and scheduling algorithm. Section 3 describes the evaluation of the scheduling algorithm. Finally, Section 4 gives the conclusion and the future work of this research.

2 QoS Website Gateway Architecture and Request Scheduling Algorithm

Given a Web server and several classes of clients, the goal is to provide service differentiation by HTTP request scheduling on the website gateway. For concentrating on the design of the request scheduling, dynamically-generated Web pages and server clusters are not considered in this research. A dynamically generated page varies its URL and response size, causing that the gateway hard to seize the characteristics of dynamic pages. In server cluster scenarios, the issues of the server load balancing also need to be considered. Therefore, a single Web server with static Web pages is the final scenario discussed in this research.

2.1 QoS Website Gateway Architecture

The architecture of the QoS website gateway is shown in Fig. 1. All HTTP requests originated from clients pass through the gateway, and the gateway schedules them to the Web server according to the QoS policies and the service rate of the Web server. The request classifier first classifies the incoming requests into different service classes by inspecting the content ofIP headers, HTTP

headers, and payloads. The classified requests are then queued into the class queues. The request scheduler decides which request should be fetched next from a class queue and when the request should be released to the Web server according to the QoS policy table and the service rate of the Web server, respectively. For knowing the characteristics of Web pages stored in the Web server, the server prober probes the Web server before the on-line operation of the gateway. In a word, request classification, request scheduling and server probing are the three things the website gateway does for the service differentiation. The more details are discussed as follows.

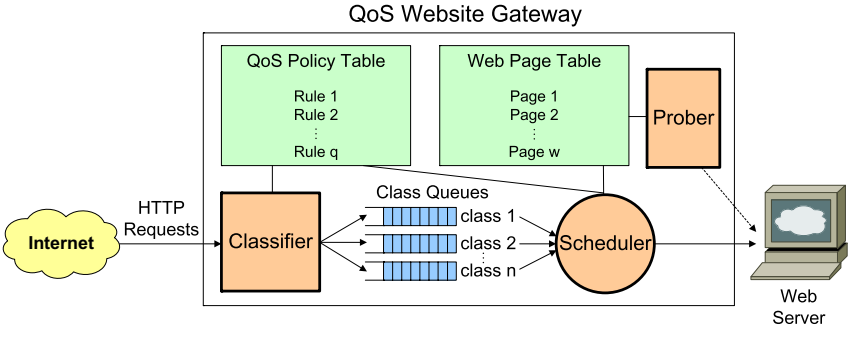


Fig. 1. Architecture of the QoS website gateway

Request Classification. A common classification paradigm is to inspect the IP 5-tuples (source IP address, destination IP address, source port number, destination port number, and protocol type) of a packet header. However, this type of classification is content-blind; that is, the classifier cannot see the information contained in the application layer protocols. The website operator may wish to define more flexible QoS policies based on the application layer protocols such as HTTP for the service differentiation. Therefore, the classifier should be content-aware, that is, it sees the information contained in the protocol headers and payloads.

The purpose of the request classifier is to classify the incoming requests into proper classes based on the QoS policy table. The rules in the policy table can be defined according to the information contained in IP packet headers, HTTP headers and HTTP payloads. HTTP headers generally contain URL, User-Agent, Content-Length, etc., whereas HTTP payloads generally contain cookie names, SSL IDs (Secure Socket Layer Identification), etc. The request classifier compares the information contained in the incoming HTTP requests with the rules in the QoS policy table. If a request matches a specific rule of a service class, it will be put into the corresponding queue and wait for being scheduled.

Request Scheduling. After requests have been classified and queued in the corresponding queues, the request scheduler decides which request should be

fetched next and when the request should be released to the Web server. For the service differentiation, each class queue is assigned a weight, and the server resource is proportionally partitioned according to the weights. The larger weight a class has, the more server resource the requests in that class can utilize. In this research, the server resource is partitioned based on the bandwidth throughput because it explicitly stands for the output rate of an HTTP response. Therefore, the request scheduler schedules requests for partitioning the bandwidth throughput of the server.

The request scheduling algorithm emulates the deficit round robin scheduling to decide which request can be fetched next according to the response size of the requests and the quanta defined in the QoS policy table. On the other hand, the window control mechanism is used to throttle the releasing rate, so as not to overwhelm the processing capacity of the Web server. The operation of the request scheduler and the window control mechanism is shown in Fig. 2.

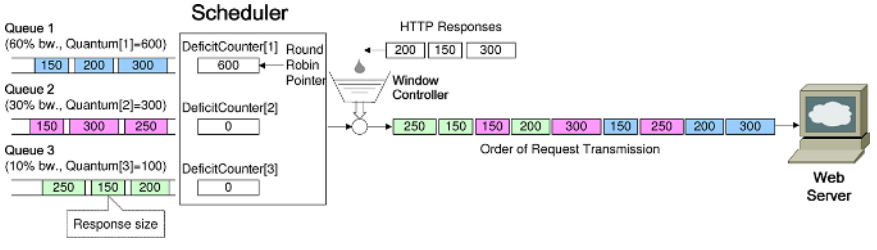


Fig. 2. Request scheduler and window controller

The original DRR scheduling is modified to be capable of handling requests. The queues store HTTP requests instead of IP packets. The numbers in the blocks in the queues represent the response sizes of the queued requests, rather than the packet sizes in the original DRR. Each queue has a deficit counter, which tracks the amount of service credits. The quantum of a queue is added to its corresponding deficit counter in a round robin manner. If the value of a deficit counter is larger than or equal to the response size of the head-of-line request in the corresponding class queue, the scheduler fetches this request; otherwise, this amount is carried over to the next round. In this way, the server resource, i.e. bandwidth throughput, can be proportionally partitioned to each service class according to the QoS policies.

The releasing rate of this modified DRR scheduling should be throttled such that the released requests would not overwhelm the processing capacity of the Web server. For this, a window control mechanism is presented to adjust the releasing rate. The window size stands for the number of the maximum concurrent connections between the gateway and the Web server. When a request is released, the window value is decremented by one. Conversely, when a resulting

response has passed through the gateway, the window value is incremented by one. The scheduler checks the window before releasing a request. If the window value is not zero, the scheduler releases the scheduled requests to the Web server; otherwise, it stops releasing the requests and waits until the window value is not zero. In this way, the processing capacity of the Web server can be utilized well without being overwhelmed. A small number (less than ten) is suggested to be assigned to the window size because a large window size may lead to an over-loaded server.

Server Probing. The request scheduler needs the response sizes of Web pages stored in the Web server when performing the scheduling. The server prober is used to probe the URL and the response size of each Web page on the server before the on-line operation of the gateway. The probed results are recorded in the Web page table and fed to the request scheduler.

For probing the URL and the response size of each Web page on the server, the server prober first retrieves the homepage of the website, parses the homepage to find the embedded objects and the other hyperlinks. The prober recursively scans the Web pages within the same server link by link until all Web pages have been scanned. The probed URL and the response size of each page will be recorded in the Web page table, and they are mainly used for the initial accesses of the Web pages. Because the Web pages and the embedded objects on the server are assumed to be static, each URL and the corresponding response size is one-to-one mapping. The Web page table will be repeatedly updated by the later accesses of the Web pages. By this way, if the content of a Web page is changed in the future, i.e. the page size is changed; the request scheduler can update the Web page table because it knows the latest response size when receiving this page from the Web server.

2.2 Request Scheduling Algorithm

The pseudo code of the scheduling algorithm is shown in Fig. 3. Initially, all deficit counters are set to zero. Upon arrival of a request, the $EnqueuingModule$ invokes $Classify()$ and $Enqueue()$ to classify the request and enqueue it to the corresponding queue, respectively. The $ActiveList$ is used to avoid the overhead of examining empty queues. It maintains a list of indices of the active queues containing at least one request. In the $DequeuingModule$, the $While\ (TRUE)$ loop plays the role of the round robin. The active class queues are processed from the head of the $ActiveList$, say the class i. The scheduling algorithm fetches requests from $Queue_i$ when there is enough service quantum and the $window$ is not zero. The service quantum $DeficitCounter_i + Quantum_i$ determines how many requests can be fetched from the $Queue_i$, that is, the sum of the response sizes of the fetched requests cannot be greater than this service quantum. Before fetching and releasing a request, i.e. invoking $Send(Dequeue(Queue_i))$, the scheduling algorithm checks if the $window$ is not zero. If the $window$ is not zero, the schedulingalgorithm releases the request and decrements the $window$ by one.

Otherwise, the scheduling algorithm will not release any requests in the $Queue_i$. After a resulting response has passed through the gateway, the *window* will be incremented by one.

3 Performance Evaluation

The effect of the service differentiation can be evaluated on both the throughput and user-perceived latency. The aggregated throughput and the user-perceived latency of each service class are measured for comparing the effects between the activation and the deactivation of the request scheduling. The measurement is performed with fixed-sized and mixed-sized Web pages to demonstrate the robustness of our request scheduling algorithm.

3.1 Evaluation Environment

The evaluation environment consists of an Apache Web server[12], a QoS website gateway, and several computers running the WebBench Web performance testing tool [13]. All the the computers are Pentium III 1GHz systems with 256 MBytes main memory and 100 Mbps Ethernet network adaptors. The request scheduling algorithm and the related components are implemented on the NetBSD. A WebBench client issues a new request after it has completely received a response from the server. This means the sending rate of clients depends on the processing rate of the server. In this evaluation, the WebBench clients are divided into three service classes, whose ratio of the quanta is set to 6:3:1.

3.2 Evaluation with Fixed-Size Web Pages

The evaluation with fixed-size Web pages is to observe the effects of the page size, which is changed from 32 bytes to 128K bytes. The resulting throughputs are shown in Fig. 4(a) and 4(b), in which the throughput increases with the page size. The increase of the page size leads to the higher aggregated response size of the requested pages, i.e. throughput. In Fig. 4(a), under the QoS-disabled case, the three service classes get the almost same throughputs because their requests have the same probability of entering the server. Nevertheless, in Fig. 4(b), under the QoS-enabled case, the three service classes get the expected throughputs. The larger weight a service class has, the higher throughput this class gets. In addition, the throughput of the class with the largest weight is improved by up to 176% when the page size is 128K bytes, while that of the class with the smallest weight is penalized by 52%. Furthermore, the average of the total throughput 14.2 Mbps under the QoS-enabled case is higher than 11.7 Mbps under the QoS-disabled case because the request scheduling throttles the releasing request rate to avoid overwhelming the server.

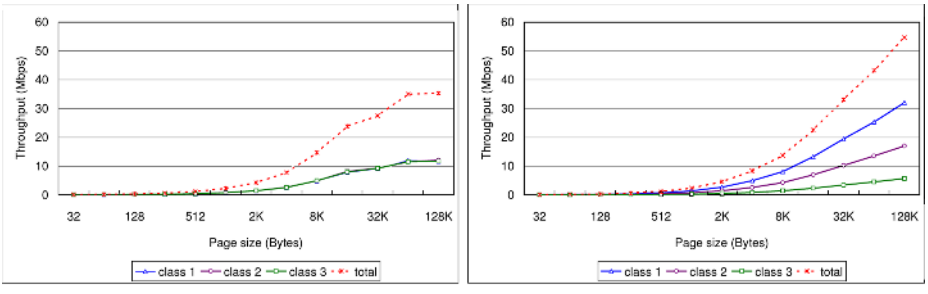
The user-perceived latencies under the two cases are also compared, as shown in Fig. 4(c) and 4(d). The user-perceived latency increases with the page size because the gateway has to process more packets for each response. The three service classes get the same user-perceived latency when the QoS is disabled,

```
Incoming Request Part
Initialization:
  For (i = 0; i < NumofClasses; i = i + 1)
    DeficitCounter_i = 0;
Enqueuing Module: on arrival of request req
  i = Classify(req);
  If (ExistsInActiveList(i) == FALSE) then
    InsertTailActiveList(i); /* add queue i to active list */
    DeficitCounter_i = 0;
  Enqueue(i, req); /* enqueue request req to queue i */
Dequeuing Module:
  While (TRUE) do
    If (ActiveList is not empty) then
      i = RemoveHeadActiveList();
      DeficitCounter_i = DeficitCounter_i + Quantum_i;
      While ((DeficitCounter_i > 0) and (Queue_i is not empty)) do
        req = Head(Queue_i); /* get request req from queue i */
        ResponseSize = GetSize(req);
        If (ResponseSize <= DeficitCounter_i) then
          If (window ≠ 0) then
            Send(Dequeue(Queue_i));
            DeficitCounter_i = DeficitCounter_i − ResponseSize;
            window = window − 1;
          Else /* return to the original condition */
            InsertHeadActiveList(i);
            DeficitCounter_i = DeficitCounter_i − Quantum_i;
            return(); /* exit this module */
        Else
          break; /* skip the while loop */
      If (Queue_i is empty) then
        DeficitCounter_i = 0;
      Else
        InsertTailActiveList(i);
    Else
      return(); /* exit this module */
Outgoing Response Part
Enqueuing Module: on arrival of response rsp
  Enqueue(rsp);
Dequeuing Module:
  While (TRUE) do
    If (Queue is not empty) then
      Send(Dequeue(Queue));
      window = window + 1;
```

Fig. 3. Request scheduling algorithm

whereas they perceive different latencies when the QoS is enabled. The larger weight a service class has, the shorter latency this class obtains under the QoS-enabled case. In addition, the user-perceived latency of the class with the largest

weight is improved by up to 69% when the page size is 128K bytes, while that of the class with the smallest weight is penalized by 75%. Note that the average of the user-perceived latency 351 ms under the QoS-enabled case is shorter than 440 ms under the QoS-disabled case. This proves the presented scheduling algorithm eliminates the server bottleneck and improves the total throughput.



(a) QoS-disabled throughput.

(b) QoS-enabled throughput.

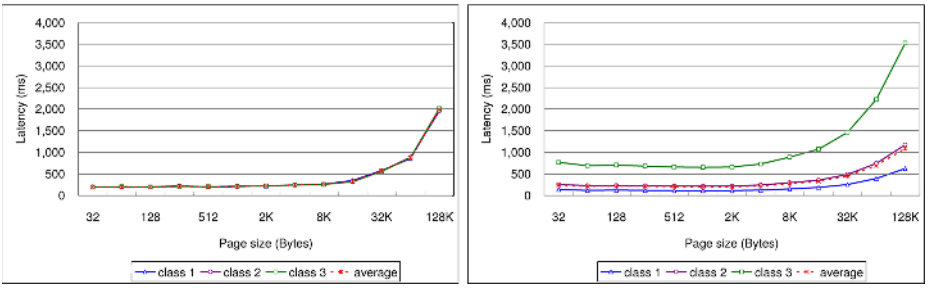


Fig. 4. Throughput and user-perceived latency under various fixed-size Web pages

3.3 Evaluation with Mixed-Size Web Pages

In order to evaluate the QoS website gateway in a more realistic environment, the mixed-size Web pages are employed on the server. The page sizes have a log-normal distribution [14], whose probability density function is shown as follows. σ (standard deviation for the natural logarithm of the data) and μ (mean for the natural logarithm of the data) are set to 9.357 and 1.318, respectively.

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2/2\sigma^2}$$

The throughput of each class is shown in Fig. 5(a). The ratio of the throughputs under the QoS-enabled case is still as expected, close to 6:3:1, demonstrating the request scheduling algorithm works well even in a more realistic environment. The user-perceived latency is shown in Fig. 5(b). The observations on the evaluation with mixed-size Web pages are similar to that with fixed-size Web pages.

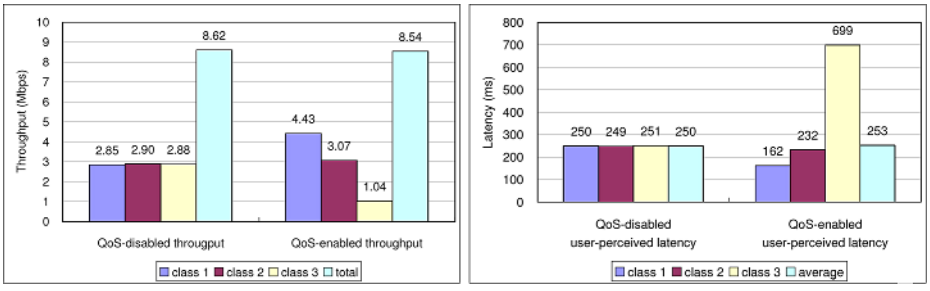


Fig. 5. Throughput and user-perceived latency under mixed-size Web pages

4 Conclusion

Service differentiation is a way for website operators to provide better throughput and shorter user-perceived latency to some specific users. This paper presents a request scheduling algorithm deployed at a website gateway to enable the Web qoS without any modification to client or server software. The QoS website gateway consists of a request classifier, a request scheduler, and a server prober. The content-aware request classifier classifies and queues incoming HTTP requests into corresponding class queues according to the pre-defined QoS policies. The request scheduler and the window control mechanism decide which request should be fetched next and when the request should be released to the Web server. The server prober scans URLs, gets the corresponding response size of the Web pages on the server, and feeds the probed results to the request scheduler for helping the scheduling. The QoS website gateway is evaluated in the scenarios of fixed-size Web pages and mixed-size Web pages to demonstrate the robustness of the request scheduling algorithm. The results show the effectiveness of service differentiation on the throughput and user-perceived latency. The total server throughput is also improved.

There are mainly two directions for future works. The Web pages on a Web server can be statically or dynamically generated. The URL and the response size of a static page are easy to be seized and use for the scheduling. A dynamically generated Web page varies its URL and response size. When dealing with dynamic Web pages, the QoS website gateway has to be equipped with a more sophisticated server prober to correctly estimate the response size. Furthermore, enabling service differentiation at the QoS website gateway for a server cluster is also a considerable work. In this case, the QoS website gateway has to schedule the requests for the service differentiation and balance the server load simultaneously.

References

1. Mills, P. Loosley, C.:A performance Analysis of 40 e-Business Web Sites, CMG Journal of Computer Resource Management, Issue 102 (2001)
2. Braden, R., Clark, D., Shenker, S.: Integrated Services in the Internet Architecture: an Overview, IETF RFC 1633, www.rfc-editor.org/rfc/rfc1633.txt (1994)

3. Blake, S., Black, D., et al.: An Architecture for Differentiated Services, IETF RFC 2475, www.rfc-editor.org/rfc/rfc2475.txt (1998)
4. Braden, R. Ed., Zhang, L., et al.: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, IETF RFC 2205, www.rfc-editor.org/rfc/rfc2205.txt (1997)
5. Almeida, J., Dabu, M., Manikutty, A., Cao, P.: Providing Differentiated Levels of Service in Web Content Hosting, Proccedings of the 1998 Workshop on Internet Server Performance, (1998)
6. Pandey, R., Barnes, J. F., Olsson, R.: Supporting Quality of Service in HTTP Servers, Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing (1998) 247–256
7. Eggert, L., Heidemann, J.: Application-Level Differentiated Services for Web Servers, World Wide Web Journal, Vol. 2, No. 3 (1999) 133–142
8. Bhatti, N., Friedrich, R.: Web Server Support for Tiered Services, IEEE Network, Vol. 13, Issue 5 (1999) 64–71
9. Vasiliou, N., Lutfiyya, H.: Providing a Differentiated Quality of Service in a World Wide Web Server, ACM SIGMETRICS Performance Evaluation Review, Vol. 28, Issue 2 (2000) 22–28
10. Chen, X., Mohapatra, P.: Performance Evaluation of Service Differentiating Internet Servers, IEEE Transaction on Computers, Vol. 51, Issue 11 (2002) 1368–1375
11. Shreedhar, M., Varghese, G.: Efficient Fair Queuing Using Deficit Round-Robin, IEEE/ACM Transaction on Networking, Vol. 4, Issue 3 (1996) 75–385
12. The Apache HTTP Server Project, http://httpd.apache.org/
13. WebBench, http://www.veritest.com/benchmarks/webbench/
14. Barford, P., Crovella, M.: Generating Representative Web Workloads for Network and Server Performance Evaluation, ACM SIGMETRICS Performance Evaluation Review, Vol. 26, Issue 1 (1998) 151–160

A Tunnel-Based QoS Management Framework for Delivering Broadband Internet on Trains

Frederic Van Quickenborne, Filip De Greve,
Filip De Turck, Ingrid Moerman, and Piet Demeester

Department of Information Technology (INTEC)
Ghent University - IBBT
Gaston Crommenlaan 8, bus 201, B-9050 Gent, Belgium
Tel.: +32 9 33 14974, Fax: +32 9 33 14899
{frederic.vanquickenborne, filip.degreve}@intec.ugent.be

Abstract. Current satellite, GPRS and GSM systems show different shortcomings to provide Broadband Internet access to trains. In this paper, we motivate that an Ethernet based aggregation network in combination with WiFi and WiMAX antennas is the best approach for realizing Broadband Internet access in trains. The focus is on the management system for the Ethernet aggregation network and more specifically on the implementation of the module for tunnel switching trigger management. The components of the management system are presented and different tunnel switching strategies supported by the framework are compared in terms of minimal, average and maximal packet loss.

Keywords: Ethernet, Aggregation network, GPS, Packet loss.

1 Introduction

1.1 Motivation

Providing Broadband Internet access to railway passengers is an interesting challenge. With the current emerging trials and early commercial releases it is only a matter of time before best-effort Internet on the train will become a reality. The satellite-based communication systems were the first solutions on the market but they lack uplink connectivity from train to satellite. Recently the first commercial bi-directional satellite communication system (4 Mbit/s down and 2 Mbit/s up) has been realized that can offer high speed Internet to high speed trains [1], for instance on the Thalys. However as our own experiments on the Thalys trial Paris-Brussels show (see Table 1) the high satellite latencies make real-time communication impossible. This is due to the fact that the signal has to travel four times the distance Earth-satellite (twice for query and twice for answer). Currently, also WiFi is used as a technology to deliver Broadband Internet to trains, as Japan Telecom has announced [2] that, in conjunction with Hokkaido Railway Company (JR Hokkaido), they have successfully demonstrated stable wireless Broadband Internet connections on the train using WiFi hotspots placed along the railways. The experiment showed that the high-speed

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 552–561, 2006.

Table 1. Main results from the Thalys Paris-Brussels experiment

Experiment	Result
Average experienced down-link bandwidth	approx. 1 Mbit/s
Average ICMP round-trip time	616 milliseconds
Average Voice over IP latency	4 seconds

wireless LAN system is capable of offering high-quality Internet connections on a train traveling at speeds of 120 km/h or faster. The network quality is sufficient for IP telephony and video streaming services. But also an alternative system architecture based on the WiMAX pre-IEEE 802.16e standard (also called the Mobile WiMAX standard or WiBro) is gaining interest: on-roof antennas with WiMAX base-stations located near the railroad track provide a bi-directional Broadband connection of 32 Mbit/s with seamless handoff in a high-speed environment [3]. In April 2005, the first Broadband WiMAX service on trains in the UK got operational [4]. However, besides good results with WiFi and good mobility support in the Mobile WiMAX standard, still the aggregation network which is responsible for the transport of data traffic from the fast moving users to the service providers' networks, has to be configured in-time. This transport of data traffic between the service providers' domains and the WiFi and WiMAX antennas is done by means of tunnels in the aggregation network. The challenge is to design telecom networks in such a way that high bandwidth services which require a high level of Quality of Service – such as multimedia content delivery, video phoning and on-line gaming – can be provided. These tunnels are needed in order to guarantee this QoS. The focus in this paper is on the management of tunnel set-up and tear-down triggers in an Ethernet aggregation network for both WiFi and WiMAX technologies. Table 2 compares both technologies with respect to the bandwidth, the coverage and the mobility support.

1.2 Ethernet Aggregation Network

This choice for Ethernet in the aggregation network is motivated by the fact that telecom operators tend mainly for economical reasons towards networks consisting of standard QoS-aware Ethernet switches. Ethernet networks [5] use a spanning tree protocol to maintain a loop free active topology. The legacy IEEE 802.1D Spanning Tree Protocol (STP) and the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP), both use only N-1 links in a network of N nodes. This limits the amount of links that can be used in these networks but with the introduction of the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) the

Table 2. Comparison of WiFi versus WiMAX technologies

Technology	Bandwidth	Coverage	Mobility
WiFi	54 Mbit/s	200 meters	Bad
Fixed WiMAX	75 Mbit/s	4.5 to 7.5 km (maximum of 45 km)	Bad
Mobile WiMAX	15 MBit/s	1.5 to 4.5 km	Good

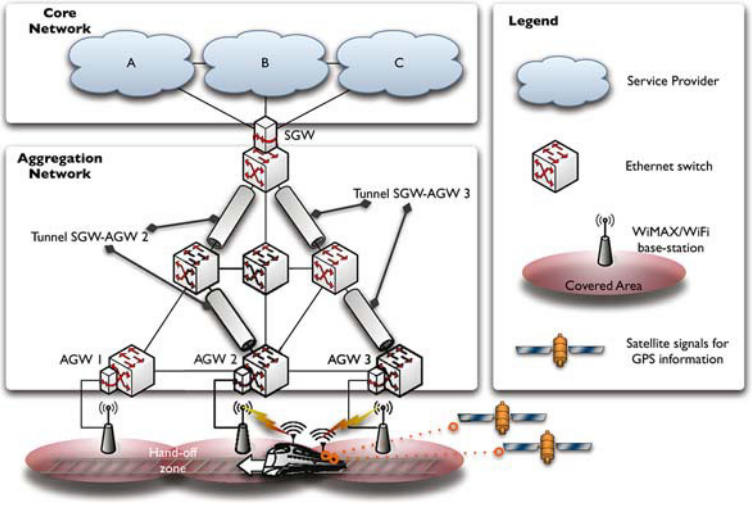


Fig. 1. Schematic representation of considered network architecture, which consists of a core part, an aggregation part and networks on the trains

bandwidth efficiency can be improved by maintaining multiple trees instead of a single tree. Most commercial switches are IEEE 802.1q & p compliant: i.e., they support the Virtual LAN (VLAN) technology and are QoS-aware (based on priority scheduling). VLANs provide a way of separating the physical topology in different logical networks and can be used to define end-to-end tunnels in the network. The configuration of VLANs can be performed automatically by means of the standardized GVRP (GARP VLAN Registration Protocol) or can be done in a management-based way by contacting every network device separately. In summary, the ease of use and the auto-configuration of standard Ethernet, in combination with the recent advances in QoS support are probably Ethernet's strongest features.

The remainder of the paper is structured as follows: Section 2 presents the considered network architecture, whereas Section 3 details the implemented system for tunnel switching trigger management. The management system and the design of the framework are addressed in Section 4. Section 5 gives the evaluation results of tunnel-based QoS management in the framework. Finally, the main conclusions are summed up in Section 6.

2 Considered Architecture

2.1 Aggregation Network

Due to the limitations of latency for fast moving users, we do not consider satellite technology, but opt for WiFi and WiMAX-based solutions. The FAMOUS (= FAst MOving USers) network architecture has already been published by the authors [6] and is depicted in Figure 1. As can be seen in this Figure, WiMAX or

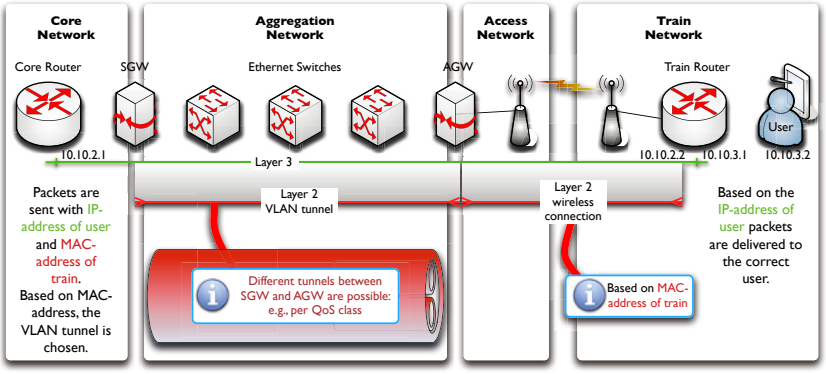


Fig. 2. Layer 3 and Layer 2 connections through the network

WiFi base-stations are positioned along the railroad track and every base-station is connected to an AGW (Access GateWay) which gives access to the aggregation network part. Note that in the aggregation network part traffic demands from separate users are not considered but groups of users are aggregated together. By aggregating the requests the system remains scalable for aggregation networks of realistic size. The traffic of each group of moving users is multiplexed in the AGWs into a VLAN tunnel. The aggregation network is responsible for the transport of the data traffic, by means of high bandwidth tunnels moving at high speed, to the service provider (SP) domain. The connection between the SPs and the aggregation network is realized by Service Gateways (SGWs).

2.2 Tunnel Establishment

While commuters are moving along the railroad trajectory, their attachment point to the aggregation network will hop from one AGW to another. In order to preserve the connection between the train and the core network, tunnels must move with the trains. Due to the moving tunnel concept, seamless connectivity is not assured. However, service guarantees can be assured by making on-time resource reservations in the aggregation network. This prevents high congestion levels which are inherently harmful for the network performance during tunnel switching. There can be multiple connections per train, dependent on the number of antennas on the roof of the train. In this paper we assume that every train has multiple antennas but that it is using a single associated tunnel in the aggregation network.

An overview of the overall network configuration is depicted in Figure 2. Between the core router (placed in the service provider domain) and the train router (placed on every train) an IP-connection is established. IP-packets going from the core router to the train router, use the IP-address of the user as destination IP-address and are encapsulated in Ethernet packets with the MAC-address of the train antenna as destination address. In the other direction, packets are always sent to the core router except from the packets for other users in the same train. The Layer 3 connection consists of two Layer 2 tunnels. The first tunnel

(as seen from the core router towards the train routers) is a VLAN tunnel. The VLANs are fixed end-to-end tunnels, automatically installed by the management system. At their due time, tunnel registrations are done for every connection, but only when the connection will be effectively using the VLAN tunnel. When the train is no longer connected to the AGW and tunnel registrations are no longer required, they are released. In this way the system will always guarantee that the current and the next hop tunnel will be able to maintain the service level and that useless reservations are prevented. Based on the MAC-address of the packets (and thus based on the `train_id`) the packets are mapped on the correct tunnel. This tunnel goes from the SGW towards the AGW where the train is connected to, via different Ethernet switches. Because the SGW, the Ethernet switches and AGWs are all 802.1p-aware, packets with higher priority will be handled first, and so QoS is assured. At the destination AGW of the tunnel, all packets are transmitted over the air to the connected train(s). Only the train with the correct MAC-address will receive the packets. Finally, the packets are delivered to the correct user, based on the IP-address of the packet. This paper details the automatic set-up and tear-down of the VLAN tunnels, based on the location information transmitted by the trains.

3 Management System

3.1 Overview

This section focuses on the specific management framework for tunnel switching trigger, taking into account the exact positions of the trains. A diagram of a management system for aggregation networks to support fast mobile users has already been presented in [7] and [8]. Figure 3 shows the management system. It consists of two parts: an off-line part and an online part. In the off-line part, the initial tunnel path calculation is based on the results of the network dimensioning process. The implemented network dimensioning module takes multiple input parameters into account, as shown in the figure. The required network capacity and the initial tunnel paths are calculated and the results are stored in the database. This database is used by the online part to set-up and tear-down the correct tunnels. In the online part, triggering the trains delivers information of the train's positions. The component that gets this information, looks up the relevant information in the tables based on the received location information of the trains. From the data of the database, the decision is made if a new tunnel must be activated and/or if an existing one must be deactivated. The monitoring component collects information on the usage of the tunnels. Based on this information, the GUI is kept up-to-date and displays the most recent status of the network devices. The central component uses all the provided information to update the tables, hence the tables contain most recent information.

3.2 Framework Design Considerations

In order to measure the performance of the different approaches, a framework has been developed. The framework is depicted in Figure 4. All the network devices

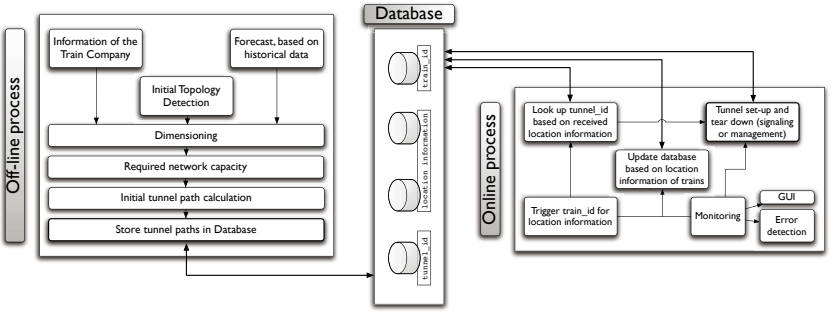


Fig. 3. Schematic overview of the management system, consisting of two parts: an off-line part and an online part. The off-line process is done before any train is moving on the rail tracks. Its main purpose is the calculation of the needed tunnels. The different components of the off-line part are published by the authors in [10]. The second process is the online one. The main purpose of this part is to activate the needed tunnels at their due time and deactivate them when they are not needed anymore.

are emulated using the Click modular router toolkit [9]. The Click modular router toolkit is a software architecture for building flexible and configurable routers. We implemented a Layer 2 data plane and a Layer 3 control plane in Click for the different network devices, with some additional features:

Layer 2 data plane - To achieve a fully enabled Ethernet switched network, all devices are implemented as Ethernet switches with VLAN support and with 802.1p-support by providing two queues per port. All the network devices are also GVRP-aware. The trains are connected as depicted in the Figure: they have physical links to all AGWs they pass by. Off course, not all the links will be active at all time. This is discussed in a next paragraph. The Spanning Tree Protocol is running, but only the management tunnel (VLAN tunnel 1) is aware of the Spanning Tree. Other VLAN tunnels define their own Spanning Tree, as proposed in the Per VLAN Spanning Tree [11] solution.
Layer 3 control plane - This control plane is used to communicate between the different network devices for management actions. We implemented two control mechanisms: (i.) the management and (ii.) the GVRP approach. The first one sets up control connections from the SGW with each of the network devices via CORBA, in order to perform the needed VLAN tunnel actions. GVRP, on the other hand, needs trigger messages to both end-stations of the tunnel. The further signaling messages from these end-stations to the rest of the network use the management tunnel. More information can be found in [12].
Additional features - As mentioned before, some extras are also defined: (i.) Links between trains and AGWs: there is always just one link active between each train and the AGWs, so all the other links of the same train are inactive. This is maintained by each train separately. (ii.) Location information: this information is controlled by each train separately and sent to the SGW, each time the SGW asks for this information. CORBA is used for these queries. (iii.) Database in

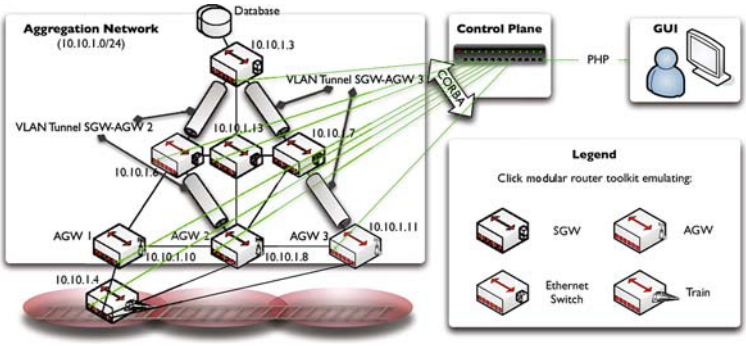


Fig. 4. Framework with the different components. It consists of two planes: the control plane and the data plane. The mentioned IP-addresses are used for the control plane. In the control plane, CORBA and PHP are used as protocols for Layer 3 communication between the network devices. The data plane only has Layer 2-functionality. The VLAN tunnels are situated in the data plane and are used for the data traffic.

SGW: the database is locally stored in the SGW and thus very fast. For each train, a pointer is kept, indicating its last known position.

It is important to mention that this framework only measures packet loss and latency as a consequence of the tunnel set-up and tear-down actions, based on location information of the trains. No packets are lost and no delay is introduced due to the wireless connection, as this has already been studied by other research groups.

4 Evaluation Tunnel Set-Up and Tear-Down

As stated before, we use location information of the trains to set-up the needed tunnels and to tear-down the ones that are not needed any more. GPS information, received by the trains from the satellites, can be used for very accurate train positioning. As disadvantages of GPS, we mention the purchase and installation of the GPS infrastructure on trains, the network usage by the broadcast messages with the GPS location information inside and the malfunctioning of GPS infrastructure in train-tunnels. The latter disadvantage can be dealt with by predicting the train position [8]. Remark that many trains are already equipped with GPS infrastructure. A flow chart of the overall process is depicted in Figure 5. The queries on the trains are sent via the control plane. An alternative approach is to use the management tunnels. Each SGW-AGW pair has such a tunnel, and because the needed bandwidth is low (only low bandwidth management information must be transmitted), these tunnels are left active continuously. On the Figure the times are given that are needed for the different actions (the used hard- and software is also mentioned on the Figure). The first step, the query for the train position, takes about 1 ms. A parameter sets the time between two queries, we call this parameter the `location update timer`. Because the

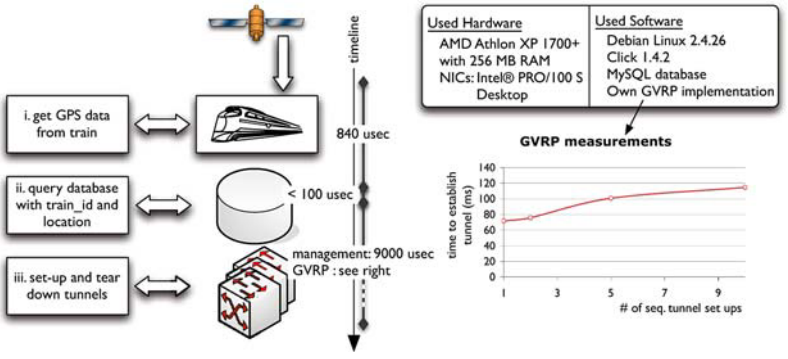


Fig. 5. Flow chart of the process, consisting of three steps: a query for the position of the train, a query for the VLAN `tunnel_id`s and the set-up and tear-down actions

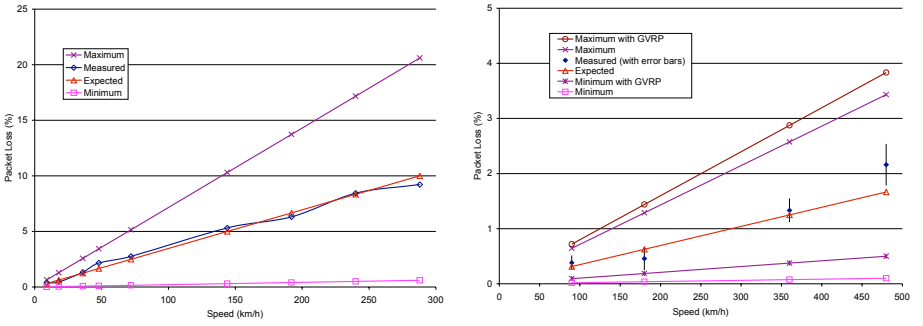
database is stored locally and because we make use of a pointer indicating the last known position of the train, the look up time is very fast. For the last phase, we studied two possibilities: the management approach and the GVRP approach. Because the management approach can connect to all network devices almost simultaneously, it is much faster then the GVRP approach. On the other hand, the GVRP method only needs two manual control messages, is independent of the length of the tunnel, and GVRP is resilient, because it automatically recovers from errors. More information on GVRP and the time measurements are published in [12]. The total time for the management method takes about 10 ms and is independent of the tunnel length. For the GVRP approach, the set-up time is dependent on the tunnel length, but for the tunnels that we consider in this paper (we take tunnels with one or two intermediate hops), it takes up to 100 ms.

5 Performance Measurements

For the performance measurements, the used test flows are generated by means of the Smartbits traffic generator. One port of the Smartbits device is connected to the train and one port is connected to the highest network device, where all the tunnels end. Traffic is sent from users in the train to the SP, because this gives a good value for packet loss due to activation of the tunnels triggered by the positions of the train. If packet loss is measured for the reverse direction, also packets are lost due to the bad traffic engineering done by the SGW, namely because the SGW has to decide where the packets must be sent to.

5.1 Measured Packet Loss

Figure 6(a) and Figure 6(b) depict the measured packet loss values for different speeds of the train, if WiFi resp. WiMAX base stations are placed along the



(a) WiFi technology. (b) WiMAX technology.

Fig. 6. Packet loss with respect to the speed of the train

railways. The values for the WiMAX based solution are significantly lower, because less tunnel switches are needed for the same speed, as the covered area is almost 10 times larger.

5.2 Expected Packet Loss

Due to the assumed uniform distribution of the tunnel switch times within the interval between any two location updates, the minimum and maximum packet loss values can be calculated as shown in equations (1) and (2). The expected packet loss (in %) is given by $\frac{\text{min+max}}{2}$. As mentioned before the time to set-up tunnels is 10 ms for the management approach. The time between two antennas is dependent on the used technology and the speed of the train. In the graphs, the packet loss results are depicted for the `location update timer` equal to 500 ms, for the WiFi and WiMAX technology, respectively.

$$\text{min (in \%)} = \frac{\text{total tunnel setup time}}{\text{time between two antennas}} \tag{1}$$

$$\text{max (in \%)} = \frac{\text{total tunnel setup time } + \text{ location update timer}}{\text{time between two antennas}} \tag{2}$$

5.3 Management Versus GVRP

In Figure 6(b) the minimal and maximal packet loss values for both the management and GVRP approach (total tunnel setup time is 70 ms in this case) are depicted, based on the equations given in the previous subsection. We can state that the difference between both maximum values are negligible, in comparison with the minimum values. Indeed, the minimum packet loss for a train, going at 300 km/h is less than 0.075 % for the management approach and is approximately 5 times more (0.35 %) for the GVRP approach. This small difference has big consequences for video quality.

6 Conclusion

It has been proven that an Ethernet based aggregation network in combination with WiFi and WiMAX antennas is the best approach for delivering Broadband Internet on trains, due to the tunnel reservations in the aggregation network. The components responsible for the tunnel set-up and tear-down have been presented. Finally, different tunnel switching strategies have been compared on our framework. It has been proven that WiMAX is better than WiFi in terms of packet loss due to the tunnel switching mechanism in the aggregation network. However packets loss is only avoidable if multiple tunnels are simultaneously reserved in the network, leading to a less optimal network usage.

Acknowledgment

Research funded by PhD grant for Frederic Van Quickenborne (IWT-Vlaanderen).

References

1. Broadband internet access on train, web site. *http://www.21net.com*.
2. JCNNetwork. Japan telecom succeeds in broadband internet access experiment for trains. *http://www.japancorp.net/Article.Asp?Art_ID=8788*.
3. Wi-lan launches libra mobilis, press communication. *http://www.wi-lan.com/news/press/20041019.htm*.
4. P. Judge. 100 mph wimax hits the rails to brighton. *http://www.techworld.com/mobility/features/index.cfm?FeatureID=1351*, 2005.
5. IEEE 802.1. Standards for local and metropolitan area networks.
6. F. De Greve, F. Van Quickenborne, and et al. Famous: A network architecture for delivering multimedia services to fast moving users. *Wireless Personal Communications Journal*, 33(3-4):281–304, June 2005.
7. F. Van Quickenborne, F. De Greve, F. De Turck, and P. Demeester. On the management of aggregation networks with rapidly moving traffic demands. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2005.
8. F. Van Quickenborne, F. De Greve, F. De Turck, I. Moerman, and P. Demeester. Management of aggregation networks for broadband internet access in fast moving trains. In *Mobility Aware Technologies and Applications*, volume 3447 of *Lecture Notes in Computer Science*, pages 273–283, 2005.
9. F. Van Quickenborne, F. De Greve, F. De Turck, I. Moerman, B. Dhoedt, and P. Demeester. Optimization models for designing aggregation networks to support fast moving users. In Gabriele Kotsis and Otto Spaniol, editors, *EuroNGI Workshop*, volume 3427 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 2004.
10. R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek. The click modular router project. *http://www.pdos.lcs.mit.edu/click/*.
11. Cisco Systems. Per-vlan spanning tree (pvst) maintains a spanning tree instance for each vlan configured in the network. *http://www.cisco.com/en/US/tech/tk389/tk621/tk846/tech_protocol_home.html*.
12. F. Van Quickenborne, F. De Greve, P. Van Heuven, F. De Turck, B. Vermeulen, S. Van den Berghe, I. Moerman, and P. Demeester. Tunnel set-up mechanisms in ethernet networks for fast moving users. *NETWORKS*, 2004.

A Resource Management Mechanism for Hose Model Based VPN QoS Provisioning*

Haesun Byun[1], Hyeonje Woo[1], Kyoungmin Kim[1], and Meejeong Lee[2]

[1] Department of Computer Science and Engineering,
Ewha Womans University, Korea
{ladybhs, hjwoo, kmk}@ewhain.net,
[2] lmj@ewha.ac.kr

Abstract. Among the resource management mechanisms for the hose based Virtual Private Network(VPN) Quality of Service(QoS), VPN-specific state provisioning allows the service provider to obtain highest resource multiplexing gains. However, users of a VPN may experience unfair usage of resources among themselves since the reserved resources of a VPN are shared by the VPN users in a similar way that the traditional LAN bandwidth is shared by the attached hosts. In this paper, we propose a resource reservation protocol and a traffic service mechanism, which not only enable dynamic and automatic resource reservation according to the VPN-specific state provisioning algorithm, but also enforce the fair usage of reserved resources among the users of a VPN in case of congestion.

1 Introduction

Virtual Private Networks(VPNs) are likely to be used by customers as replacement for networks constructed using private lines, and therefore Quality of Service(QoS), together with the security, is an intrinsic part of a VPN service. Traditionally, VPN QoS requests are done in a way that traffic demand is specified for each site pair and resources are reserved for point-to-point pipes (usually called a *customer pipe*) between these VPN endpoints. Duffield et al. proposed another VPN QoS service model called *hose model*[1]. A hose can be considered as a link from a user site to the network. In the hose model, instead of the complete traffic matrix, the total amount of traffic that a user site injects into and receives from the network through a hose is specified, making the specification of QoS requests a lot simpler compared to the traditional customer pipe-based specifications. In addition to this, there are several advantages to the hose model from a customer perspective. It provides flexibility by allowing data to and from a given hose endpoint to be distributed arbitrarily over other endpoints. Customers can also obtain statistical multiplexing gains since hose rate is usually less than the aggregate rate required for a set of customer pipes.

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment), and by the ITEP(Korea Institute of Industrial Technology Evalution and Planning).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 562–571, 2006.

From a provider's perspective, though, it presents a more challenging resource management problem due to the need to meet the Service Level Agreements(SLAs) with a very week specification of the traffic matrix. Feasibility of using hose model in practice calls for a bandwidth efficient resource provisioning mechanism. To cope with the challenges, mechanisms exploiting the statistical multiplexing, dynamic measuring and resizing or routing algorithms minimizing the capacity requirements are studied extensively[1],[2],[3],[4],[5].

Resource provisioning for the hoses can be implemented in several ways[1]. The differences of the alternatives are mainly with respect to the level of resource sharing. The simplest one is to rely on default point-to-point shortest path routing and not making use of any hose-specific state information for resource sharing: network resources to accommodate worst case traffic split, i.e., the traffic from each hose can be directed entirely to just one other endpoint, are reserved on the default path between each end point pair(usually called a *provider pipe*). Resource sharing is not accounted for in this approach. A provider can make use of the hose-specific state parameters in order to achieve resource sharing: source trees, rooted at ingress points of a VPN and spanning all of the egress points of the VPN (hereinafter, the terminology hose tree is used for this source tree), can be formed, and the provider can leverage the knowledge of hose parameters to determine the amount of resources to be reserved on the hose tree. Resource sharing within the hose is achieved in this approach. Finally, by taking into account the VPN-specific state parameters, further capacity reduction can be obtained. The set of hoses of a VPN constitutes a graph, and the entire hose parameters of the VPN, i.e., the VPN-specific state parameters, are taken into account to determine the amount of resources to be reserved on the links of this graph. Note the reserved resources on the graph are shared by all of the hoses for the VPN. In the hose/VPN-specific state provisioning, employing explicit routing to maximize the number of links on which resource sharing is achieved and thereby minimize the capacity requirement is possible. Various algorithms and analysis for the problem of finding a tree with the optimal cost resource provisioning have been studied in [2],[3],[4],[5],[6].

Yet another issue on the VPN QoS provisioning is related to applying the above provisioning mechanisms in a real network. In order to deploy the mechanisms in practice, a resource reservation protocol is necessary for dynamic and automatic provisioning of networks. To our best knowledge, however, this problem has not been knuckled down to yet. There exist protocols such as RSVP-TE and P2MP RSVP-TE, proposed for the resource reservation in the MPLS networks in general[7],[8]. However, both of them are not appropriate for VPN-specific state provisioning by several reasons. RSVP-TE is a protocol to set up a Point-to-Point Traffic Engineered Label Switched Paths(P2P TE LSPs), and it allows resource sharing by the LSPs with the identical egress points only, whereas VPN-specific state provisioning requires the capability to allow the sharing of resources among the LSPs with different ingress and/or egress points. P2MP RSVP-TE is an extension of RSVP-TE to set up Point-to-Multipoint(P2MP) TE LSPs for multicast transmissions, and it enables resource sharing by the

LSPs with different ingress and/or egress points. However, it does not have label assignment mechanism for unicast transmission, which is required by VPN unicast transmissions. In addition, the mechanism to compute the amount of resources to be reserved on a link according to the VPN-specific state provisioning algorithm is not defined. Therefore, a new or an extension of the existing resource reservation protocol needs to be defined for dynamic and automatic resource provisioning based on VPN-specific state.

Furthermore, while the capacity requirement of VPN-specific state provisioning is obviously the minimum among the three resource provisioning mechanisms of the hose model, unfair usage of capacity among the users of VPN may occur since the way that reserved resources are shared by the VPN users is similar to the way that the traditional LAN bandwidth is shared by the attached hosts. Therefore, a mechanism enforcing a fair usage of the reserved resources is necessary in case of congestion. In this paper, we propose a resource reservation protocol and a traffic service mechanism to implement dynamic and automatic resource provisioning based on VPN-specific state and to provide a fair usage of reserved resources to the VPN users. The rest of this paper is organized as follows. In section 2, the proposed mechanism is explained in detail. Simulation results exhibiting the effect of the proposed mechanism are presented in section 3. Finally, conclusions of our study are given in section 4.

2 A Resource Reservation Protocol and a Traffic Service Mechanism for VPN-Specific State Provisioning

Instead of introducing a whole new mechanism, we propose a mechanism that is based on the P2MP RSVP-TE, which is already proposed to IETF for standardization. The proposed mechanism elaborates a set of modifications to the P2MP RSVP TE, leveraging the resource sharing capability of P2MP RSVP-TE. It provides the fair usage of VPN resources as well as a dynamic and automatic resource provisioning of a VPN according to the VPN-specific state. Basically, the way that RSVP messages are transmitted between the sources and the destinations is very similar to what is done in the P2MP RSVP-TE. Therefore, we focus on the aspects that are specific to the proposed mechanism in explaining the operation of proposed mechanism. Specifically, the extensions to the RSVP message formats and the structures of Path State Blocks(PSBs) and Reservation State Blocks(RSBs), and the modifications to the processing of RSVP messages are defined. A service mechanism of VPN traffic to enforce the fair usage of reserved VPN resources is also specified.

2.1 Extensions to the P2MP RSVP-TE

Fig. 1 and 2 show the formats of Path and Resv messages respectively for the proposed mechanism. They are basically similar to the formats of P2MP RSVP-TE Path and Resv messages except for the several extensions and renaming of objects. In Fig. 1 and 2, those changes and extensions are identified with shading.

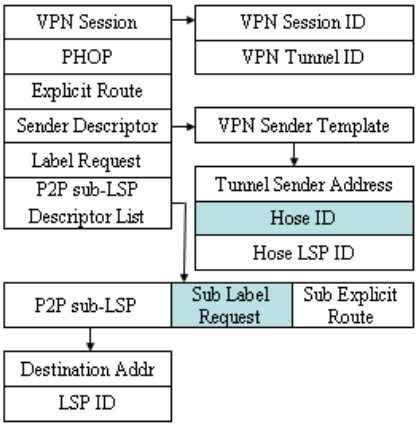


Fig. 1. Format of a Path message

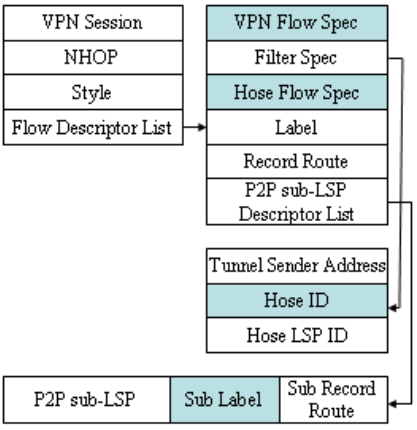


Fig. 2. Format of a Resv message

The *VPN Session* object is a renaming of the *P2MP Session* object in P2MP RSVP-TE. It is composed of the *VPN Session ID* and the *VPN Tunnel ID*, through which the P2P sub-LSPs belonging to a same VPN associate themselves with one another. Fig. 3 illustrates the relationship between the VPN tunnel and its hose trees and P2P sub-LSPs. For the *VPN Session ID*, a multicast group IP address, which needs to be distributed to the Provider Edge(PE) devices by the administrator, is supposed to be used.

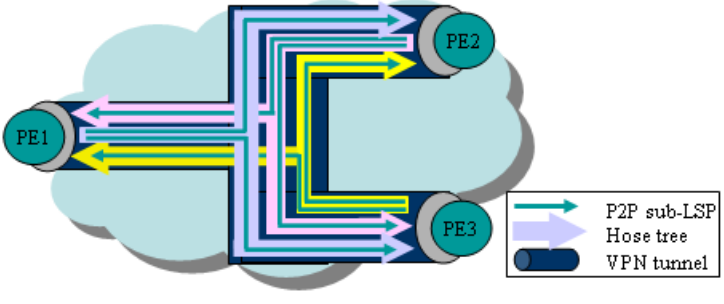


Fig. 3. Relationship between the VPN tunnel and its hose trees and P2P sub-LSPs

Extensions are made mainly due to the differences of the VPN tunnel of the proposed mechanism and the P2MP tunnel of the P2MP RSVP-TE. The first extension is an addition of the *Hose ID* field in the *VPN Sender Template* object of a Path message and the *Filter Spec* object of a Resv message. The *Hose ID* field identifies the traffic source of a VPN tunnel. While a P2MP tunnel of P2MP RSVP-TE always has a unique traffic source, multiple traffic sources may exist for the VPN tunnel set up by the proposed mechanism, and hence requires the explicit specification of the traffic source. Note if multiple user sites belonging

to a same VPN are attached to a single PE, a VPN tunnel, which starts from the PE, has multiple traffic sources corresponding to those user sites.

The second extension is the addition of *Sub Label Request* object and *Sub label* object to the *P2P sub-LSP Descriptor* object of Path and Resv messages respectively. Unlike the P2MP tunnel of P2MP RSVP-TE, where multicast is the only transmission style, data packets of a VPN tunnel may need to be transmitted in unicast. Therefore, each P2P sub-LSP requires separate label assignment.

The last extension is related to the fair usage of VPN resources. Since VPN-specific state provisioning allows resource sharing among all the hose trees, Shared Explicit(SE) style reservation is made[7]. Even though the SE style itself requires a single *Flow Spec* (i.e., a single specification of reserved resources) for multiple *Filter Specs* (i.e., sources of the traffic), the proposed mechanism inserts a new object called *Hose Flow Spec* for each *Filter Spec* in order to inform the routers of the size of the hose bought at each traffic source site so that the router may enforce the fair usage of VPN resources according to that size if congestion happens.

Each router maintains PSBs for the Path messages. A PSB is created per [*VPN Session*, *VPN Sender Template*] at the arrival interface of the Path message. Fig. 4 shows the structure of a PSB. An ingress PE generates Path messages for the hoses attached to itself. A Path message may contain one or more *P2P sub-LSP Descriptors* corresponding to the destinations of the hose. If an intermediate router receives a Path message, it first checks whether a matching PSB exists at the message arriving interface. Matching PSB is defined as the PSB with the [*VPN Session*, *VPN Sender Template*] of the received Path message. If the matching PSB exists, it is updated and refreshed. Otherwise, a new PSB is created for the Path message.

VPN Session	VPN Sender Tamplate	Sender Tspec	PHOP	In Intf	P2P sub-LSP	ERO	Label	Out Intf	Expiration Time
					P2P sub-LSP	ERO	Label	Out Intf	Expiration Time

Fig. 4. Structure of a PSB

VPN Session	Resv Intf	Style	VPN Flow Spec	Filter Spec	NHOP	P2P sub-LSP	RRO	Label	Expiration Time
						P2P sub-LSP	RRO	Label	Expiration Time
						P2P sub-LSP	RRO	Label	Expiration Time
				Filter Spec	NHOP	P2P sub-LSP	RRO	Label	Expiration Time
						P2P sub-LSP	RRO	Label	Expiration Time
						P2P sub-LSP	RRO	Label	Expiration Time

Fig. 5. Structure of a RSB

When an egress PE receives a Path message, corresponding Resv message is generated and sent back to the source of the Path message. In order to have the

hose trees of a VPN share the resources, reservation style of the Resv message need to be set to SE. If an intermediate router receives a Resv message, it first validates the legitimacy of the received Resv message by checking whether there is a *corresponding PSB*. The corresponding PSB of a Resv message is defined as the PSB with [*VPN Session*, *VPN Sender Template*, *P2P sub-LSP* entries, *Out Intf*] that matches [*VPN Session*, *Filter Spec*, *P2P sub-LSP Descriptor List*, the arriving interface] of the Resv message.

If the Resv message is a valid one, the active RSB is then looked for. An RSB is maintained for each VPN at the arriving interface of a Resv message. Fig. 5 shows the structure of a RSB. The active RSB is the RSB maintained at the Resv message arriving interface with the [*VPN Session*] of the Resv message. If the active RSB exists, it is updated and refreshed with the information in the Resv message. Otherwise, a new RSB is created for the Resv message.

Specifically, the bandwidth value of *Hose Flow Spec* object, denoted by B_{Hose}, and the bandwidth value of *VPN Flow Spec* object, denoted by B_{VPN}, in the active RSB are set as follows respectively:

$$\begin{cases} \text{If the router is an egress PE,} & B_{Hose} = \min\left(P_{CorrPSB}, \sum\limits_{s \in S} H_s\right), \\ \text{otherwise,} & B_{Hose} = M_{Hose} \end{cases}$$

where $P_{CorrPSB}$ denotes the bandwidth value of *Sender Tspec* object in the corresponding PSB, S is the set of user sites attached to the PE, H_s is the size of hose from the PE to the user site s, and M_{Hose} denotes the bandwidth value of *Hose Flow Spec* object in the Resv message. On the other hand,

$$\begin{cases} \text{If the router is an egress PE,} & B_{VPN} = \min\left(\sum\limits_{k \in K} P_k, \sum\limits_{s \in S} H_s\right), \\ \text{otherwise,} & B_{VPN} = M_{VPN} \end{cases}$$

where K is the set of PSBs with [*VPN Session*, *In Intf*] of the corresponding PSB, P_k denotes the bandwidth value of *Sender Tspec* in the PSB k, and M_{VPN} denotes the bandwidth value of *VPN Flow Spec* object in the Resv message.

After the update or creation of a RSB, the Resv message is transmitted to the next hop toward the traffic source. Reflecting the VPN-specific state provisioning algorithm[1], the bandwidth values of *VPN Flow Spec* and *Hose Flow Spec* objects, denoted by M_{VPN} and M_{Hose} respectively, in the Resv message to be transmitted, are computed as follows:

$$\begin{cases} M_{Hose} = \min\left(P_{CorrPSB}, \sum\limits_{h \in H} B^h_{Hose}\right) \\ M_{VPN} = \min\left(\sum\limits_{k \in K} P_k, \sum\limits_{v \in V} B^v_{VPN}\right) \end{cases},$$

where H is the set of RSBs with [*VPN Session*, *Filter Spec*, *NHOP*] of active RSB, B^h_{Hose} denotes the bandwidth value of *Hose Flow Spec* object in RSB h, V is the set of RSBs with [*VPN Session*, *NHOP*], and B^v_{VPN} denotes the bandwidth value of *VPN Flow Spec* object in RSB v.

2.2 Fair Usage of VPN Resources

Before introducing our fair usage enforcing mechanism, let us define the notion of fair share of VPN resources for a certain user site or a hose. Let's assume that P2P sub-LSPs of N different hoses pass a certain link, and the bandwidth assignment on that link to a hose i according to the hose-specific state provisioning is B^i_{Hose} , and to the VPN according to the VPN-specific state provisioning is B_{VPN} respectively. The fair share of hose i, denoted by B^i_f is defined when VPN-specific state provisioning is used, and it is defined as follows:

$$B^i_f = \left(\frac{B^i_{Hose}}{\sum_{k=1}^{N} B^k_{Hose}} \right) B_{VPN} \tag{1}$$

Fair usage enforcing mechanism is implemented in two folds. First of all, the hoses of a VPN sharing an outgoing interface queue are guaranteed to take at least $Q_f = \frac{Q}{N}$ spaces in the queue, where Q is the capacity of the queue and N is the number of hoses sharing the queue. In order to implement this, a packet counter is maintained for each hose. Let q_h denote the number of packets in the queue belonging to hose h. When a packet from hose h arrives at the queue, if the queue length q, which is equal to $\sum_{k=1}^{N} q_k$, is less than Q, it is accepted. If $(q \geq Q)$ and $(q_h < Q)$, then a hose h' with $q_{h'} > Q_f$ is selected and the last packet of hose h' is dropped to make space for the arriving packet. In order to avoid selecting the same hose repeatedly, the search goes in round robin fashion starting from the next hose of the previously selected one. If $(q \geq Q)$ and $(q_h \geq Q_f)$, then the arriving packet is dropped.

In Intf	In Label	Hose ID	Out Intf	Out Label
IF1	L1	Hose 1	IF2	L2
IF1	L3	Hose 2	IF2	L4

Fig. 6. Label switching table

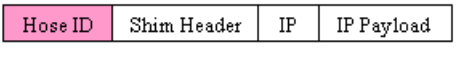


Fig. 7. Dummy header to identify a hose

The second tire of fair usage enforcing mechanism is to provide a logical Weighted Fair Queuing(WFQ) service to the hoses sharing an outgoing queue. In order to do this, it is necessary to identify the corresponding hose of each packet in the outgoing queue. To this end, we extend the label switching table with a *Hose ID* field as shown in Fig. 6. When the label switching table is looked up, the corresponding *Hose ID* is also obtained, and it is attached to the head of a packet as shown in Fig. 7. Fig. 8 illustrates the 2 level WFQ at an outgoing interface. The outer level service provides WFQ service to the different queues, corresponding to different VPN sessions/flows, according to the bandwidth allocation determined by the VPN-specific state. Whereas the inner level service provides a logical WFQ service to the different hoses sharing a single

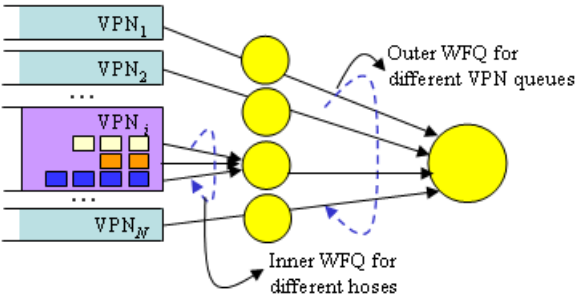


Fig. 8. Two level WFQ

queue according to the fair share of bandwidth allocated to the hoses. The *Hose ID* is detached when the packet leaves the outgoing queue.

3 Performance Evaluation

The performance of proposed mechanism is studied through a set of simulation experiments. The simulation is implemented with the Opnet Modeler 11.0. The plain VPN-specific state provisioning and the VPN-specific state provisioning with the fair usage enforcement mechanism are compared. Hereinafter, they are called *plain-VPN* and *fair-VPN* respectively. Fig. 9 shows the simulation network model. There are three hoses in the network: *H1* and *H2* are ingress hoses, and *H3* is an egress hose. The size of hoses is 10 *Mbps* each. The amount of bandwidth to be reserved on each link according to the VPN-specific state provisioning is indicated on each link. 4*Mbps* and 4∼10*Mbps* CBR traffic flows are generated by site 1 and site 2 respectively and all of them are destined to site 3. Each LSR has a queue with the capacity of 1000 packets of 512*byte* size. Congestion may happen at the router *P* when the amount of traffic from site 1 and 2 toward site 3 exceeds the reserved capacity at *P*. Even though it is a small network, the effect of fair-VPN can be clearly manifested, and without loss of generality, it could be considered as a part of a large network, where the impact of proposed mechanism is specifically rendered.

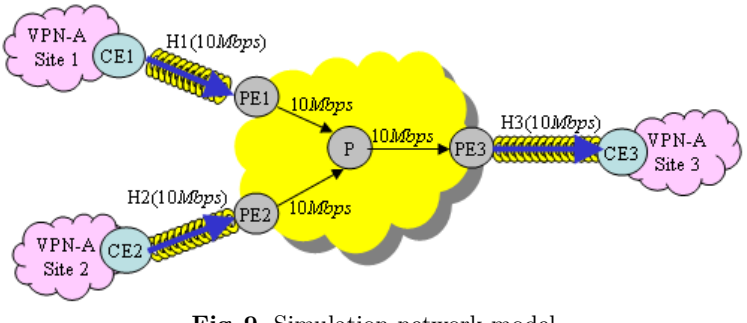


Fig. 9. Simulation network model

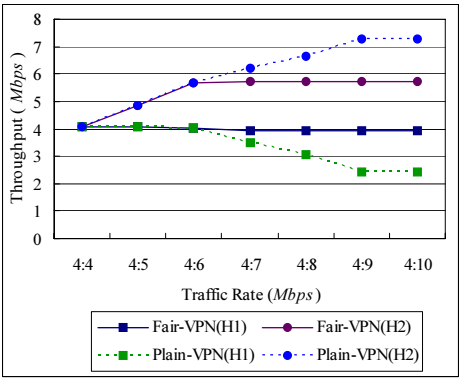


Fig. 10. Throughput

Fig. 10 shows the per hose throughput in *Mbps*. In all of the graphs in this section, the x axis is marked with a:b, and a and b represents the rate of traffic injected by the site 1 and site 2 respectively. When congestion does not occur, i.e., the sum of traffic rate from site 1 and site 2 is less than the reserved bandwidth, the throughput of *H1* and *H2* are almost the same as the rate of user traffic injected through each hose respectively. Congestion is experienced at P when the injected traffic from *H2* is larger than $6Mbps$. If congestion occurs, the throughput of both *H1* and *H2*, in plain-VPN, becomes lower than the rate of traffic generated from site 1 and site 2 respectively. On the other hand, in fair-VPN, due to the proposed fair usage enforcement mechanism, the throughput of *H1*, whose traffic rate does not exceed its fair share at the congested link, is kept to the rate of traffic generated by the user site even when congestion occurs, while *H2*, whose traffic rate exceeds the fair share, suffers lowered throughput. Note if a VPN is provisioned with VPN-specific state provisioning, congestion occurs in a VPN only when the traffic injected by one or more sites toward a certain destination exceeds the capacity of egress hose at the destination. In this

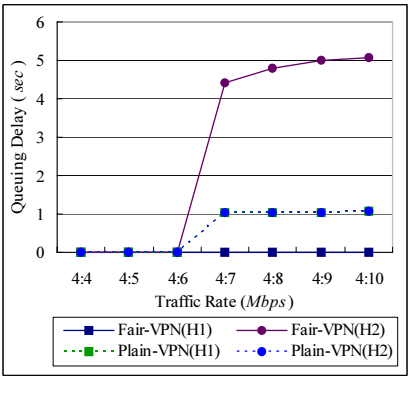


Fig. 11. Queuing delay at router P

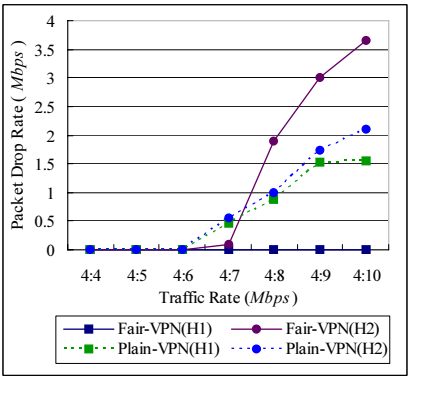


Fig. 12. Packet drop rate at router P

case it is fair not to sacrifice a user site whose traffic rate is lower than its fair share on a congested link if there is no other specific priority policy is specified.

Fig. 11 and 12 show the queuing delay and the packet drop rate respectively at the router P, where the congestion may occur. When the amount of traffic from *H1* and *H2* does not exceed the reserved bandwidth at the link between P and *PE3*, traffic from both *H1* and *H2* experience almost no queuing or a packet drop. If congestion occurs, i.e., the traffic rate of *H2* is higher than $6Mbps$, the packet drop rate as well as the queuing delay of user traffic from both hoses increase in plain-VPN, and the level of performance degradation that the hoses experience is similar. However, in fair-VPN, *H1*, whose traffic rate does not exceed its fair share at the congested link, does not suffer any performance degradation, and only the traffic from *H2* suffers the congestion.

4 Conclusions

In this paper, we propose a resource reservation protocol and a traffic service mechanism to implement dynamic and automatic resource provisioning based on VPN-specific state and to provide a fair usage of reserved resources to the VPN users. Through simulation experiments the effectiveness of proposed mechanism is presented. We have also implemented Opnet process models for the entire resource reservation protocol suit that is proposed in this paper, and a more through and general simulation experiments, with a practical network model such as Korea Telecom VPN network and with various realistic VPN user traffic models, are currently under way.

References

1. N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, J. E. Van der Merwe: Resource Management With Hoses: Point-to-Cloud Services for Virtual Private Networks, IEEE/ACM Transactions on Networking, Vol.10, No.5, October 2002
2. A. Kumar, R. Rastogi, A. Silberschatz, B. Yener: Algorithms for Provisioning Virtual Private Networks in the Hose Model, IEEE/ACM Transactions on Networking, Vol.10, No.4, August 2002
3. Gustavo de Veciana, Sangkyu Park, Aimin sang and Steven Weber: Routing and Provisioning VPNs based on Hose Traffic Models and/or Constraints, Conference on Communication Control and Computing, 2002
4. Thomas Erlebach, Maurice Ruegg: Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing, INFOCOM, Vol.4, March 2004
5. Yu-Liang Liu, Yeali S.Sun, Meng Chang Chen: MTRA: An On-Line Hose-Model VPN Provisioning Algorithm, Technical Report, 2004
6. Alpar Juttner, Istvan Szabo and Aron Szentesi: On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks, IEEE INFOCOM 2003
7. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow: RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC3209, December 2001
8. R. Aggarwal, D. Papadimitriou, S. Yasukawa: Extensions to RSVP-TE for Point to Multipoint TE LSPs, draft-ietf-mpls-rsvp-te-p2mp-03.txt

Analysis of Multimedia Streaming Service over Server-Based Many-to-Many Overlay Multicast

Youngjun Kim, Kwanghoon Kim, Moonsoo Kang, and Jeonghoon Mo

Wireless Internet and Networks Lab.
Information and Communications University
119 Munjiro, Yuseong-gu, Daejeon, Korea
{yjkim412, hoon0217, kkamo, jhmo}@icu.ac.kr

Abstract. Among many overlay multicast algorithms, there is one that uses powerful and dedicated servers as multicast routers in the application level. The multicast tree is constructed on the servers, which is referred to as server-based overlay multicast (SOM). Clients who wish to join just access one of the servers. The SOM is adequate for real time multimedia applications, such as video conferencing. In this paper, we want to find the answer to the following question: "What is the best form of network topology that has minimum cost and maintains a minimum quality of service (QoS) of multimedia application?" To answer this question, we made analytical models of SOM with three different topologies–linear, star and hybrid types that combine the linear and star topologies–and compared their performances. We also tried to answer the maximum number of clients that can be served by a group of servers for a given QoS such as tolerable delay. We validated the analysis with an extensive simulation.

1 Introduction

Overlay multicast has gained much attention due to its easy deployment into networks to support multicast service. To use the conventional IP multicast, every router on the network should be replaced, resulting in high cost. Even though the IP multicast has been seriously studied, it is not commercially used for this reason. Despite the fact that overlay multicast shows suboptimal performance compared to IP multicast, it easily implements the functions for multicast without any change of networks because packets for multicast are replicated and forwarded to their receivers on the level of application [1,2,3,4,5]. Many researchers have studied the overlay multicast to find a more efficient multicast mesh or tree construction algorithm.

We classify the algorithms into two categories. The first category is that every user in a multicast group may be a member of the multicast tree. In other words, a user may be a terminal node or an intermediate node in the multicast tree [3,4,5]. The second category consists of powerful and dedicated servers that act as a multicast router at the application level. The overlay multicast tree is constructed in advance among them. To join a multicast session, a user

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 572–581, 2006.

accesses one of the servers [6]. The first category is more ideal and flexible. However, the resources of each user–such as computing power or memory–are not the same. Therefore the tree construction considering resources will be more complex and difficult. The second category is less flexible, but the algorithm is simpler and faster. To support large scale multicast and especially real time multimedia stream, the second one seems to be a more attractive choice in the authors' opinion. In this paper, we refer to the second as SOM. To construct such an overlay network, a network designer will have to deal with issues such as what form of network topology can best guarantee minimum cost and maintain the minimum quality of service(QoS) of multimedia application. To deal with these issues, we made analytical models of SOM with one basic topology and three different topologies–linear, star and hybrid types that combine the aforementioned two topologies–and compare their performances. We provide the maximum number of clients that can be served by a server and a group of servers for a given QoS, such as tolerable delay. We validate the analysis with an extensive simulation.

The remainder of this paper is organized as follows. Section 2 presents a survey on some related papers about video conferencing and overlay multicast. In Section 3, we introduce four queue models according to several topologies. In Section 4, we show the results of comparing the simulation with that of modeling. Finally, we offer our conclusion in Section 5.

2 Related Work

A video conference application is a type of real-time multimedia application and requires a certain level of QoS(Quality of Service) [7,8]. Contents of video conference application are transmitted via the network as packets at regular intervals and must be received without significant loss. Therefore, these packets must have a small delay variation to prevent effective loss [7]. A late loss due to delay jitter, caused by variable end-to-end delay, may impede interactive communication of video conferencing [7]. To achieve acceptable video conferencing quality, the end-to-end delay, especially the queuing delay (Q-Delay), is always smaller then the tolerable delay. Video conferencing uses standard video image size and audio and video signals share the bandwidth. Therefore, we consider the standard size of a QCIF video frame as 176 by 144 pixels, or 25,344 pixels [8]. Eight bits are typically required to encode a pixel. That is, the total size of a video frame is 202,752 bits. Motion picture films are shown at 24 frames per second. A commonly accepted lower bound of video signal consists of 15 frames per second. Fifteen video frames occupy 4,041,280 bits. Video frames can also be compressed before sending and uncompressed when they are received. A compression ratio of 10:1, which degrades the image slightly, means that one second (15 frames) of compressed video can fit into 404,128 bits [8]. In this paper, a tolerable delay of the video conference application is defined as the amount of time that users are willing to wait before giving up on communication. Delay of 150ms/packet is the limit for video application to stay uninterrupted [7]. We consider the tolerable delay of video conferencing as an important point to achieve acceptable video

conferencing quality. Thus, the delay must be smaller than the tolerable delay of the video conference application.

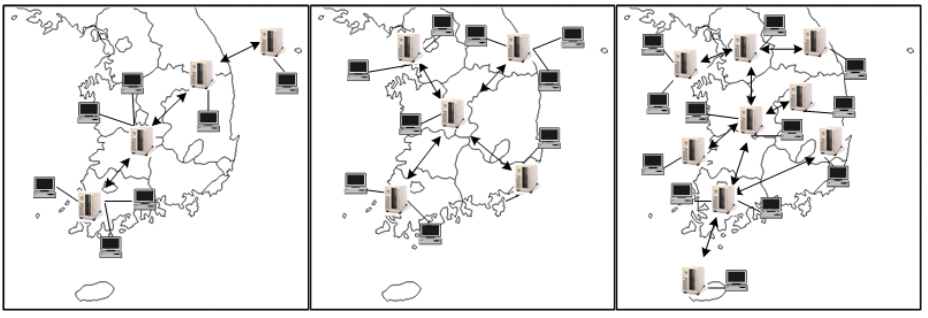
The overlay multicast network is to handle routing, group management, and overlay spanning tree construction for data delivery to optimize overall network delay. There are many projects related to overlay multicast: Narada [1], ALMI [2], YOID [3], NICE [4], Overcast [5], OMNI [6], and so on. Narada, designed for small-to-medium sized overlay networks, builds a richer mesh and then the spanning trees. That is, it uses a two step process to build and refine the source-specific multicast tree to increase its performance [1]. ALMI was proposed to support a large number of groups, each with a small number of participants. It uses a centralized algorithm to improve the multicast tree and enable better reliability and reduced overhead [2]. NICE is a cooperative framework to scale multi-party applications. Its protocol uses a distributed algorithm by which nodes are self-organized into a layered topology with nodes organized at each layer into clusters [4]. Overcast provides a scalable and reliable single-source multicast using a simple protocol for building efficient data distribution trees that adapt to changing network conditions. It mainly uses large-scale software distribution to users [5]. The Overlay Multicast Network Infrastructure (OMNI) is a two-layer approach to overlay multicast. We consider the OMNI as a model of overlay multicast network infrastructure in our research. The lower layer consists of a set of service nodes that are distributed throughout a network infrastructure. The lower layer provides data distribution services to any host, such as clients or participants in the overlay multicast session, that are connected to an OMNI node, like a server, over a directed spanning tree rooted at the source OMNI node. An end-host subscribes with an OMNI node to receive multicast data. The OMNI nodes organize themselves into an overlay network which supports the multicast data delivery.

3 Modeling

A tolerable delay of video conferencing is defined as the amount of time users are willing to wait before giving up video conferencing. We consider the tolerable delay of video conferencing as an important point to achieve acceptable quality of service. Generally, processing, transmission, propagation and Q-Delay are the four kinds of delay used. However, we concentrate on Q-Delay for server-based networks because it is the most important factor in quality of service for video conferencing [7]. Actually, other delays greatly depend on the performance of the hardware or physical aspects of the network, which is hard to control. Therefore, we consider that the Q-Delay must be smaller than the tolerable delay in video conferencing. We deduce the delay equations of each packet for several topologies of the server-based overlay network and validate analysis with simulation. Fig. 1 shows examples of three kinds of networks, such as linear, star, and hybrid networks.

3.1 2-Server Model(1:1 Server Model)

For the purpose of researching the relationship between delay and the maximum number of clients according to the structure of servers, we first assume that the



(a) Linear network topology (b) Star network topology (c) Hybrid network topology

Fig. 1. Three kinds of network topologies

number of servers is 2. From a 2-Server model, we extend our research to the general architecture of the server network. Fig. 2 (a) shows a queuing model diagram of two servers, server 1 and server 2. First of all, we assume that the number of clients connecting to server 1 in Fig. 2 (a) is m1, and that the data rate of each client is λ_1. Since each client uses the same application, it is reasonable that the data rate of each client is identical. Then, the lower part of server 1 that transmits data to server 2 can be modeled by $M/M/1$ queuing model as in Fig. 2 (a). The upper part of server 1 that transmits its data to its own client can also be modeled by another queuing model, $M^{[x]}/M/1$ Queues.

In this queuing model, we assume that the critical delay that determines the whole system's performance is the Q-Delay through two queues (the combination of In-Queue and Out-Queue of two servers). First, we explain the Q-Delay in general $M^{[x]}/M/1$ queue. In general $M^{[x]}/M/1$ queue that has k identical queues, input rate λ and service rate μ, has average Q-Delay T as in [9].

$$T = (k+1)/2(\mu - \lambda k) \tag{1}$$

Our 2-Server model has 4 possible Q-Delays, the combination of 4 queue delays. These delays are defined below.

$$T_{11} = Q - Delay\{(Bottom\,of\ Server1) + (Top\ of\ Server1)\} \tag{2}$$

$$T_{12} = Q - Delay\{(Bottom\ of\ Server1) + (Bottom\ of\ Server2)\} \tag{3}$$

$$T_{22} = Q - Delay\{(Top\ of\ Server2) + (Bottom\,of\ Server2)\} \tag{4}$$

$$T_{21} = Q - Delay\{(Top\ of\ Server2) + (Top\ of\ Server1)\} \tag{5}$$

"Bottom and Top of Server's Q-Delay" represent the queue delay of above and below queue of Fig. 2 (a). The above 4 possible delays should be less than the application delay tolerance $T(a)$. Using this condition, the following 4 constraints

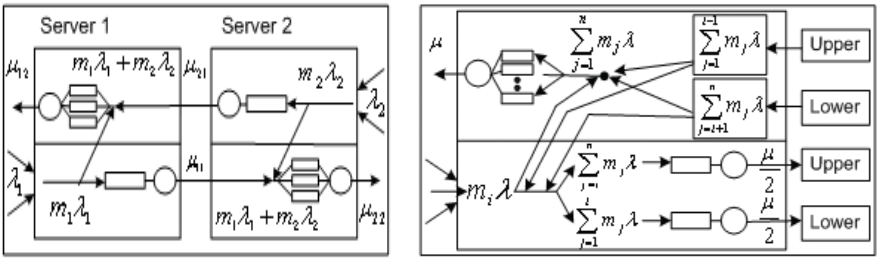
$$T_{11} = \frac{m_1 + 1}{2\{\mu - (m_1 + m_2)m_1\lambda\}} < T(a) \tag{6}$$

$$T_{12} = \frac{1}{(\mu - m_1\lambda)} + \frac{m_2 + 1}{2\{\mu - (m_1 + m_2)m_2\lambda\}} < T(a) \tag{7}$$

$$T_{22} = \frac{m_2 + 1}{2\{\mu - (m_1 + m_2)m_2\lambda\}} < T(a) \tag{8}$$

$$T_{21} = \frac{1}{(\mu - m_2\lambda)} + \frac{m_1 + 1}{2\{\mu - (m_1 + m_2)m_1\lambda\}} < T(a) \tag{9}$$

are derived. As we assume that each server has the same bandwidth and each client uses the same application, it is reasonable that $\mu_{11} = \mu_{12} = \mu_{21} = \mu_{22} = \mu$and $\lambda_1 = \lambda_2 = \lambda$. To generalize the 2-Server model in Fig. 2 (a), we investigate three network models, such as linear, star, and hybrid network models.



(a) 2-Server model (b) Linear network model

Fig. 2. 2-Server and Linear Network Models

3.2 Linear Network Model

The idea of the 2-Server model can be generalized to an arbitrary network topology based on several servers. For the purpose of simplicity, we studied the model of the linear network depicted in Fig. 2 (b). We call the server whose position is above the ith server "Upper of *ith* server" and the opposite server "Lower of that." In this structure, the queue model of the ith server is depicted in Fig. 2 (b). Assuming the total number of servers is n, the queue model in Fig. 2 (b) can be divided into two parts. One part transmits receiving packets to its clients and the other part transmits the receiving packets to its upper and lower, as shown Fig. 2 (b). The top queue can be modeled by the $M^{[x]}/M/1$ queue like the 2-Server model, and the bottom queue can be modeled by the $M/M/1$ queue.

For the sake of the calculation of delay each packet experiences, we define three kinds of delay.

$$W_{i,1} = (Q - Delay\ going\ to\ Upper\ of\ ith\ Server),\ W_{i,1} = \frac{1}{\frac{\mu}{2} - \Sigma_{j=1}^{i} m_j \lambda} \quad (10)$$

$$W_{i,2} = (Q - Delay\ going\ to\ Lower\ of\ ith\ Server),\ W_{i,2} = \frac{1}{\frac{\mu}{2} - \Sigma_{j=i}^{n} m_j \lambda} \quad (11)$$

$$W_{i,3} = (Q - Delay\ going\ to\ its\ Clients),\ W_{i,3} = \frac{1}{2(\mu - (\Sigma_{j=1}^{n} m_j \lambda) m_i)} \quad (12)$$

Each delay can be calculated using the delay definition of the $M/M/1$ queue and the $M^{[x]}/M/1$ queue. Using the above delay equations, we can define the total Q-Delay of packets that are transmitted by an *ith* server and received by a *jth* server. $T_{(i,j)}$is Q-Delay from ith server to clients of jth server.

$$T_{(i,j)} = \Sigma_{k=0}^{j-i} W_{(i+k,1)} + W_{(j,3)} \quad if\ \ i < j, \quad T_{(i,j)} = W_{(j,3)} \quad if\ \ i = j,$$

$$T_{(i,j)} = \Sigma_{k=0}^{i-j} W_{(i-k,2)} + W_{(j,3)} \quad if\ \ i > j,\ \ T_{(i,j)} < T(a)\ \ \forall_{(i,j)} \quad (13)$$

Like the 2-Server model, the total Q-Delay cannot exceed the delay tolerance of specific application. The number of above constraints is n^2. As the number of servers increases, one cannot solve the feasible solutions in polynomial time. But if we delete some redundant equations, we can reduce the above constraints to 2n. The reduced constraints sets are listed below.

$$\Sigma_{i=1}^{n} \frac{1}{\frac{\mu}{2} - \Sigma_{j=1}^{i} m_j \lambda} + \frac{m_k + 1}{2(\mu - m_k \lambda \Sigma_{i=1}^{n} m_i)} < T(a) \quad \forall_{k=1,2,\ldots,n} \quad (14)$$

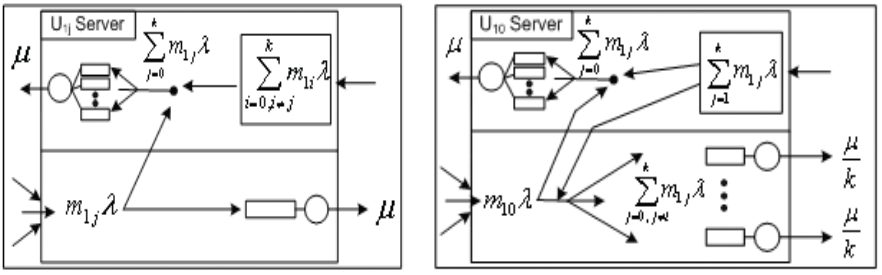
$$\Sigma_{i=1}^{n} \frac{1}{\frac{\mu}{2} - \Sigma_{j=1}^{i} m_{(n+1-j)} \lambda} + \frac{m_k + 1}{2(\mu - m_k \lambda \Sigma_{i=1}^{n} m_i)} < T(a) \quad \forall_{k=1,2,\ldots,n} \quad (15)$$

One can easily predict that the feasible region in the above constraints will be similar to that of the 2-Server model except that the dimension is n. By using the mathematical tool to find a feasible solution, the max $(\Sigma m_{i,i=1,2,\ldots,n})$ is achieved when $m_1 = m_2 = \ldots m_n$is satisfied. However, according to the increase of the server, the average delay will increase, so the number of clients per server will decrease.

3.3 Star Network Model

In this section, to generalize the 2-Server model, we investigate the linear network model. But for completeness, we need to study another fundamental structure: the star network. Fig. 3 shows the models of the star network.

The general star network structure is very complicated to analyze, so we restrict our target to the simple star network that consists of one central server and



(a) Queuing model of a side server (b) Queuing model of a central server

Fig. 3. Star Network Models (Queuing Model of $U_{1,j(j>0)}$and U_{10} Server)

many side servers which connect to the central server by one hop. It is necessary to divide the queue model into two parts, the central server (called U_{10}) and side servers (called $U_{1j,(j=1,2,\ldots,n)}$). Fig. 3 shows the U_{10} and $U_{1j,(j=1,2,\ldots,n)}$models. Similar to the linear network model, we define the building blocks of Q-Delay below. Delay can be expressed using the delay of the conventional $M/M/1$ and $M^{[x]}/M/1$ queue model.

$$W_{10,0} : Q - Delay(U_{10} \rightarrow its\ clients),\ W_{10,0} = \frac{m_{10}+1}{2(\mu - (\Sigma_{i=0}^{k} m_{1i}\lambda)m_{10})} \quad (16)$$

$$W_{10,j} : Q - Delay(U_{10} \rightarrow U_{1j(0<j\leq k)}),\ W_{10,j} = \frac{1}{(\frac{\mu}{k} - \Sigma_{i=0,i\neq j}^{k} m_{1j}\lambda)} \quad (17)$$

$$W_{1j,0} : Q - Delay(U_{1j(0<j\leq k))} \rightarrow its\ clients),\ W_{1j,0} = \frac{m_{1j}+1}{2(\mu - (\Sigma_{i=0}^{k} m_{1i}\lambda)m_{1j})} \quad (18)$$

$$W_{1j,1} : Q - Delay(U_{1j(0<j\leq k))} \rightarrow U_{10}),\ W_{1j,1} = \frac{1}{(\mu - m_{1j}\lambda)} \quad (19)$$

Using the above equations, we can define the total Q-Delay of the packet that is transmitted by $U_{1i,(i=1,2,\ldots,n)}$-Server and received by $U_{1j,(j=1,2,\ldots,n)}$-Server as:

$$T_{i,j} = W_{1i,1} + W_{10,j} + W_{1j,0},\ T_{i,j} < T(a)\ \ \forall_{(i,j)\in\{1,2,\ldots,k\}} \quad (20)$$

Similar to the linear network, the reduced constraints is:

$$W_{10,j} + W_{1j,0} < T(a)\ \ \forall_{(i,j)\in\{1,2,\ldots,k\}} \quad (21)$$

The feasible region of these constraints is very similar to the feasible region of the linear network model.

3.4 Hybrid Network Model

We have studied the general model, the linear and star network, but the real network is different. We give an example of modeling a more general network

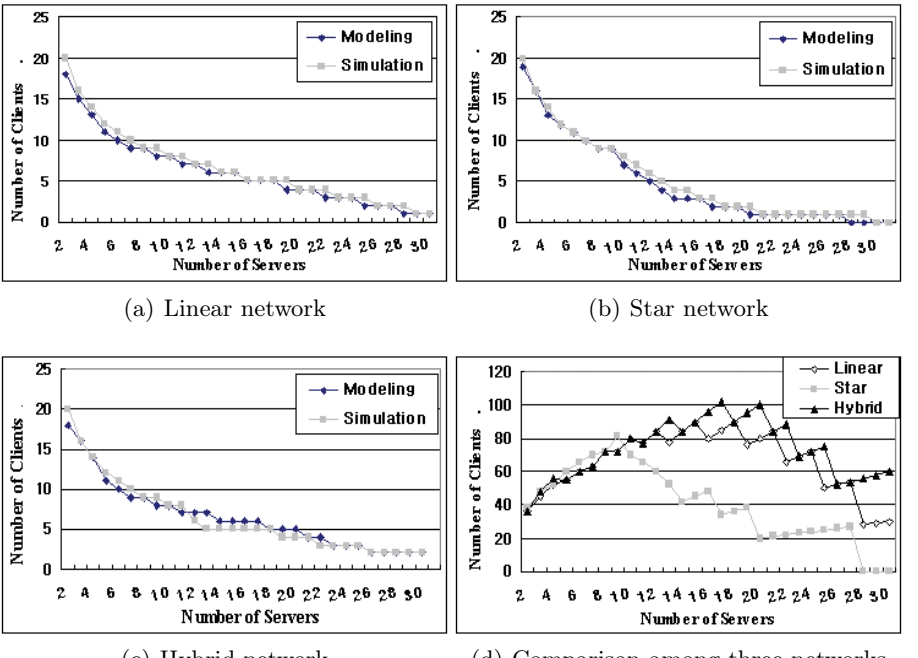
case. The simplest network model is the 1st order hybrid network, which consists of several central servers and side servers. Each central server is connected as in the linear network shown in Fig. 1 (c). The queue model of the above 1st order hybrid network can be easily constructed using the queue model of the linear and simple star network. Therefore, we omit the concrete description but for convenience we show the delay constraints of each packet. The delay between central servers is always smaller than the delay between side servers and the central server. And also, the Q-Delay is less than T(a). Therefore, it is sufficient that we only concern ourselves with delay of each side server. Then, the Q-Delay between the j_1 side server connected to i_1 central server and j_2 side server connected to i_2 central server is :

$$T_{\{(i_1,j_1),(i_2,j_2)\}}$$

$$= \frac{1}{(\mu - m_{i_1 j_1}\lambda)} + \frac{m_{i_2,j_2}+1}{2\{\mu - (\Sigma_i \Sigma_j m_{ij}) m_{i_2 j_2}\lambda\}} + \Sigma_{i=1}^{\lceil \frac{n}{k} \rceil} \frac{1}{(\frac{\mu}{k+2} - \Sigma_{l=1}^{i}\Sigma_{j=1}^{k} m_{ij}\lambda)} \quad (22)$$

4 Simulation

In the previous section, we showed analytical models of SOM with three different topologies: the linear, star and hybrid types. In order to validate the analysis, we



(a) Linear network (b) Star network

(c) Hybrid network (d) Comparison among three networks

Fig. 4. Comparison between Modeling and Simulation Results

simulate the models according to the three different topologies. Fig. 4 (a), (b) and (c) show comparisons between the results of modeling and simulation in the three different topologies. Each line shows the average number of clients served by one server as the number of servers varies. From Fig. 4 (d), the three lines show the total number of clients of the three different topologies as the number of servers varies. In the comparisons, we can see that performance of the 1^{st} order hybrid network is similar to that of other networks below 10 servers. However, the hybrid network shows much better performance than the other networks over 11 servers. As a result, the hybrid network shows better performance than the other networks as a whole.

5 Conclusions

Under the assumption that multimedia applications such as video conferencing are served with many-to-many overlay multicast on server-based networks, we present an analytical model, the Q-Delay model. It is an important point to achieve acceptable quality of service. For simplicity without losing generality, we classify the forms of overlay networks into three different topologies: the linear, star and hybrid type that combines the linear and star types, and compared the performances of these topologies. For each topology, we derived the equations of Q-Delay experienced by packets in many-to-many overlay multicast. This provides a concrete relationship between the number of servers and the maximum number of allowable clients per server.

We conclude that the linear overlay network topology does not scale well with an allowable tolerable delay. In the star network, the center of the network is easily burdened with heavy traffic. As a result, the 1^{st} order hybrid network shows better performance than other networks with tolerable delay varied. Through extensive simulation, we validated the analysis.

References

1. Chu, Y.H., Rao, S.G., Zhang, H.: A case for end system multicast. In: SIGMETRICS. (2000) 1–12
2. Pendarakis, D.E., Shi, S., Verma, D.C., Waldvogel, M.: ALMI: An application level multicast infrastructure. In: USITS. (2001) 49–60
3. Francis, P.: Yoid: Extending the internet multicast architecture. Technical report (2000) unrefereed report.
4. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. In: SIGCOMM. (2002) 205–217
5. Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, M.F., Jr, J.O.: Overcast: Reliable multicasting with an overlay network. In: OSDI. (2000) 197–212
6. Banerjee, S., Kommareddy, C., Kar, K., Bhattacharjee, S., Khuller, S.: Construction of an efficient overlay multicast infrastructure for real-time applications. In: INFOCOM. (2003)

7. Varadarajan, S.: (Multimedia: Application Layer, The lecture of the computer networking)
8. ITC: Videoconferencing, bandwidth. Technical report, (University of Virginia, Information Technology and Communication)
9. Gross, D., Harris, C.M.: Fundamentals of Queuing Theory. 3rd edn. (John Wiley & Sons)

Performance Evaluation and Comparison of Two Random Walk Models in the PCS Network

Jang Hyun Baek[1], Jae Young Seo[1,⋆], and Kyung Hee Kim[2]

[1] Dept. of Industrial and Information Systems Eng., Chonbuk Natl. University, Korea
{jbaek, jaeyoung}@chonbuk.ac.kr
[2] Dongbu Information Technology, Seoul, Korea
khkim@dongbu.com

Abstract. In this paper, we evaluate the performance of the MBR (movement–based registration) scheme using a modified one-dimensional random walk model and a two–dimensional random walk model to compare the accuracy of two random walk models. The difference between these two models is identified and the performance difference in a mathematical approach is presented to support it. Analytical results are provided to demonstrate that the modified one–dimensional random walk always overestimates the performance of the MBR and that the two–dimensional random walk model should be used to obtain the exact performance of the MBR, and other registration schemes.

1 Introduction

Owing to the rapid growth of PCS (personal communications systems) users, location management has become a key function in wireless personal communication networks. In location management, there are two major processes: terminal paging and location registration. When an incoming call arrives, the network searches the destined mobile terminal (MT) by sending a signal through the air. This process is called terminal paging. MT's information, such as location, status, and other characteristics is stored in a location database. Records of this database are updated whenever the MT performs a location update or response in the terminal paging. This database update is called location registration.

The investigation of the movement of MT is a critical problem in location management. In movement–based registration (MBR), the system performs location registration whenever the number of cell boundary crossings reaches the given threshold value.

Akyildiz [1] analyzed MBR using the simple one–dimensional random walk model. In this model, the innermost ring (ring–0) consists of only one cell. Ring–0 is surrounded by ring–1, which in turn is surrounded by ring–2, and so on. For similar purposes, the modified one–dimensional (1–D) random walk model is suggested [2]. The complex two–dimensional (2–D) random walk is reduced to a simple 1–D random walk with one barrier state contained in the different

⋆ Corresponding Author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 582–589, 2006.

definitions of ring–i. In [2], ring–i is defined as the cells that surround the ring–(i-1) cells with a reachable place.

Conversely, a 2–D random walk model for hexagonal cell configuration in a cellular network is studied [3]. This paper classifies the states, according to the proposed algorithm, reduces the number of states, and decreases the computation time.

In this paper, we evaluate the performance of the MBR scheme using a modified 1–D random walk model [2] and a 2–D random walk model [3] to compare the accuracy of two random walk models. Exact performance of the MBR with the 2–D random walk is presented. Emphasis is placed on how the performance of MBR changes as adapting the 2–D random walk model, and which factor influences the origin of differences between the two random walk models.

2 Previous Results

In this paper, we consider the MBR when MT performs the location update when the number of movement is equal to a predestined value d. This value is called the location update *threshold*. When there is an incoming call, the network pages the cells within the distance of d from the cell last registered.

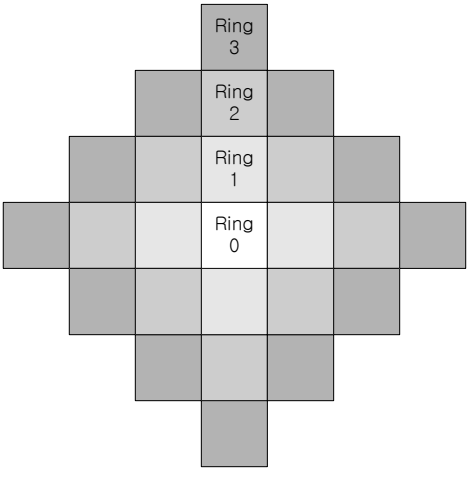


Fig. 1. Location area and rings (d=4) for the modified 1–D random walk model

MBR is analyzed with the 1–D random walk in [1]. In this paper, the square cell of PCS network is assumed, and the ring–i is defined as the cells surrounding the ring–(i-1). Because the shape of the ring is like a square, in the worst cases, the MT can not move from a cell in ring–(i-1) to corner cells of ring–i.

However, ring–i in the modified 1–D random walk model [2] is defined as the cells that surround the ring–(i-1) cells with a reachable location. Due to the different viewpoints of the definition of the ring, the number of cells in the ring changes, as well as the configuration of the ring. Figure 1 shows the mesh cell configuration in the modified 1-D random walk model [2].

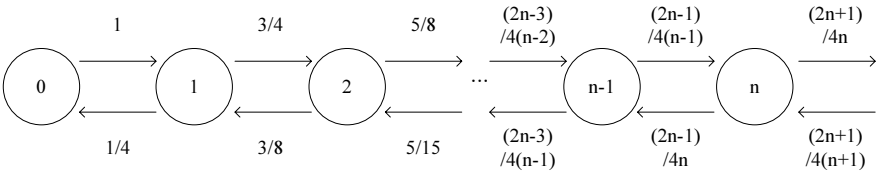


Fig. 2. State diagram with absorbing state for the modified 1–D random walk model

The state diagram of the 1–D random walk is shown in figure 2. In the figure, an MT is in state–i, when it is currently staying in ring–i. Note that after K moves, the MT can, at most, move to ring–K. Thus, we can modify the 1–D random walk model such that state 0 to K–1 are the transient states and state K is the absorbing state.

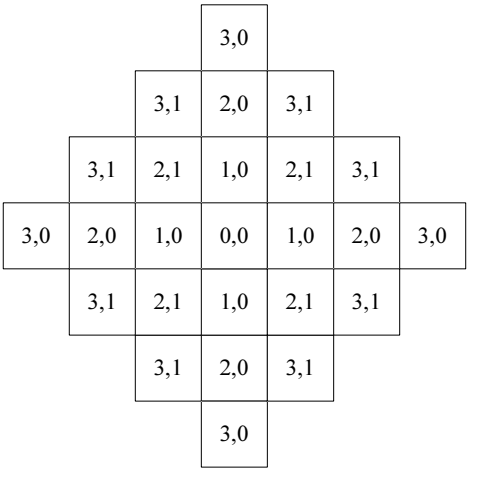


Fig. 3. Location area and rings (d=4) for the 2–D random walk model

According to the cell type classification [3], the cells of a ring in the 2–D random walk model may exist in different states, while the cells of a ring in the 1–D random walk model have the same state. The state is defined in [3] as follows *Two cells A and B are of the same state if the multiset of the state for A's neighbors is the same as that for B's neighbor.* In this algorithm, if two cells have the same relative position on different pieces, then they are grouped together and assigned to the same state. It is intuitive for the cells with different neighbors to have different types. Based on the type of classification for a hexagonal cell configuration [3], 4–division lines make the cluster into 8 symmetric pieces in a mesh cell configuration.

Figure 4 shows the state diagram for a 4–subarea cluster, as shown in figure 3, which have a corresponding (7×7) transition matrix P, where the indices represent the type (0,0), (1,0), (2,0), (2,1), (3,0), (3,1), (4,0) in the given order.

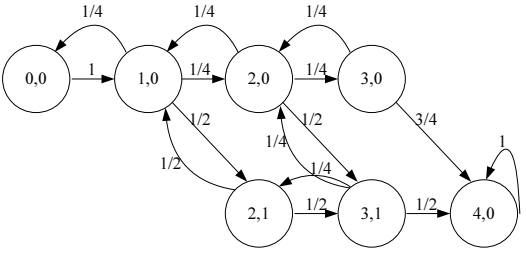


Fig. 4. State diagram(d=5) for the 2–D random walk model

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/2 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 1/4 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/4 & 0 & 0 & 0 & 3/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3 Selective Paging and Performance Evaluation

3.1 Selective Paging

When an incoming call arrives, the network system starts the paging process in order to find the exact position of the called MT. The system should be capable of locating the destined MT within η paging times where η is the maximum paging delay. For instance, if η=3, the network partitions the location area into 3 subareas and then pages each subarea one after another until the called MT is found.

Although, the line–paging [7] and the sectional–paging scheme [8] are proposed, the ring-paging scheme has almost been used in various location registration studies where the paging technique is required.

In our study, the shortest–distance–first (SDF) partitioning scheme [4] is considered. Under this scheme, the location areas are partitioned into l=min[η, d] subareas. Subarea j is denoted as A_j, where $0 \leq j < l$. Each subarea has one or more rings. subarea A_j contains ring s_j to e_j where s_j and e_j are the indices of the first and the last rings in subarea A_j. The value for s_j, e_j are denoted as

$$\begin{aligned} s_j &= \begin{cases} 0 & for\, j = 0 \\ \left\lfloor \frac{dj}{\eta} \right\rfloor & otherwise \end{cases} \\ e_j &= \left\lfloor \frac{d(j+1)}{\eta} \right\rfloor - 1 \end{aligned} \tag{1}$$

In order to locate the called MT, the network simultaneously polls all cells in subarea A_0. If the MT is found in subarea A_0 the terminal paging process is complete. Otherwise, the network polls the cells in subarea A_1 and so on.

3.2 Performance Evaluation

In order to obtain the paging cost and the registration cost, the probability $\alpha(K)$ that there are K movements between two incoming call arrivals [1] is as

$$\alpha(K) = \begin{cases} 1 - \frac{1}{\theta}[1 - f_m^*(\lambda_c)], & if\ K = 0 \\ \frac{1}{\theta}[1 - f_m^*(\lambda_c)]^2[f_m^*(\lambda_c)]^{K-1}, & if\ K > 0 \end{cases} \tag{2}$$

where $\frac{1}{\theta} = \frac{\lambda_m}{\lambda_c}$ and $f_m^*(\lambda_c)$ are the mobility–to–call ratio (MCR) and the Laplace–Stieltjes transform of the cell residence time, respectively.

The cost of performing a location update and for polling a cell is U and V, respectively. The expected location registration cost per call arrival [1], denoted by C_u, is expressed as:

$$C_u = U \sum_{i=1}^{\infty} i \sum_{j=id}^{(i+1)d-1} \alpha(j) \tag{3}$$

From the Chapman–Kolmogorov equation, the n–step transition matrix P^n is given as $P^n = P^{n-1} \times P,\ \ n \geq 1$. Let an element of $P_{i,j}$ in the modified 1–D random walk model, be the probability that a MT in ring–i moves to state-j after n cell boundary crossings, and let $\beta(k, K)$ be the probability that after K movements, the distance between the current and the initial position is k. Thus, the probability $\beta(k, K)$ is obtained as $\beta(k, K) = P_{0,k}^K$.

However, for the 2–D random walk model, let $P_{(x,y),(x',y')}^n$ in P^n be the probability that a MS at a cell of type (x, y) reaches a cell of type (x', y') after n steps. The probability that the MS from center (0, 0) moves to a cell of type (x, y) after K movements is

$$\beta((x, y), K) = P_{(0,0),(x,y)}^K \tag{4}$$

The number of cells in ring–i is $g(i) = 4i$, $i = 1, 2, 3, ...$ and the probability in the modified 1–D random walk model that the MT is located in state–i cell when a call arrives, π_i, is

$$\pi_i = \sum_{k=0}^{\infty} \alpha(k)\beta(i, k \bmod d) \tag{5}$$

On the other hand, the probability in the 2–D random walk that the MT is located in state (i, j) when a call is generated, $\pi_{(i,j)}$, is

$$\pi_{(i,j)} = \sum_{k=0}^{\infty} \alpha(k)\beta((i, j), k \bmod d) \tag{6}$$

The expected terminal paging cost per call arrival for the modified 1–D and the 2–D random walk model, denoted by C_ν^{1-D} and C_ν^{2-D}, are expressed respectively:

$$\begin{aligned} C_\nu^{1-D} &= V \sum_{k=0}^{l-1} \Big(\sum_{r_i \in A_k} \pi_i \Big) \Big(\sum_{j=0}^{k} \sum_{r_i \in A_j} g(i) \Big) \\ C_\nu^{2-D} &= V \sum_{k=0}^{l-1} \Big(\sum_{r_i \in A_k} \pi_{(i,j)} \Big) \Big(\sum_{j=0}^{k} \sum_{r_i \in A_j} g(i) \Big) \end{aligned} \tag{7}$$

In the next section, the numerical results reveal that the expected terminal paging cost per call arrival for the modified 1–D and for the 2–D random walk model, C_ν^{1-D} and C_ν^{2-D}, are unequal.

4 Numerical Results

For illustrative purposes, the mesh cell configuration of the PCS network is assumed. The cell residence time and the incoming call arrivals are assumed to follow an exponential distribution with λ_m and a Poisson process with λ_c, respectively. It is also assumed that U=10 and V=1 as in [1], [2], [6].

The location registration cost is the function of $\alpha(K)$. Under a certain MCR, both the 1–D and the 2–D random walk models always produce the same location registration cost; therefore the location registration cost can be discarded as a relevant factor in comparison of in our strategies in the study. Figure 5 shows the paging cost for varying values of threshold, d. In the figure, we can see that paging cost of the 2–D random walk model is always less than that of the 1–D random walk model. Additionally, we can see that the gap between the two models increases as the threshold value increases, and as the MCR increases. This indicates that the 1–D random walk model overestimates the performance

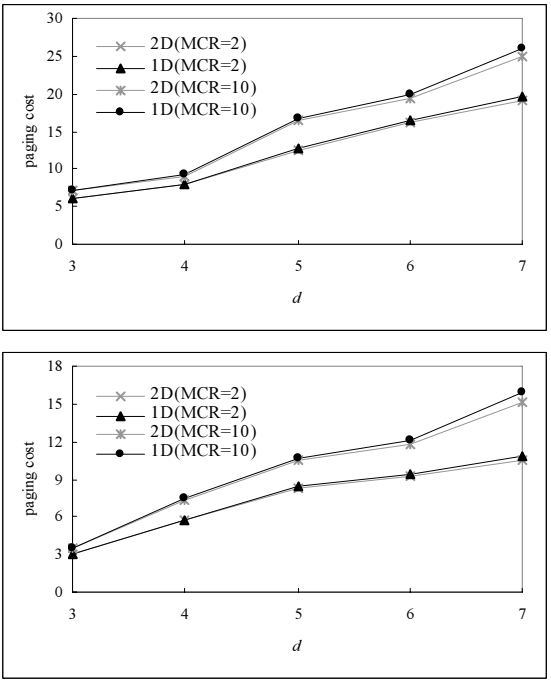


Fig. 5. Paging cost with various threshold value in maximum paging delay (η=2, 3)

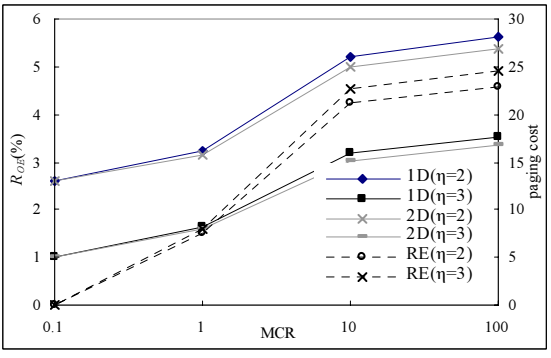


Fig. 6. Paging cost and R_{OE} changes with various MCR values, when d=7

of the MBR, and that the 2-D random walk model should be used to obtain the exact performance of the MBR and other registration schemes.

Figure 6 illustrates the paging cost changes for different MCR values, given that d=7, and that the overestimation ratio is defined as:

$$R_{OE}(\%) = 100 \times \frac{C_{\nu}^{1-D} - C_{\nu}^{2-D}}{C_{\nu}^{1-D}} \tag{8}$$

As the MCR increases, the difference between the two paging costs, which can be measured as R_{OE}, becomes larger. However, when the MCR is relatively small, both random walk models produce a small and similar paging cost, with the same paging delay. In the small MCR, many calls are designated to a MT with lower mobility. As the number of movements is small, the selective paging makes both paging costs small and similar.

Therefore, both paging costs have shown no difference, between one another. Moreover, under the same MCR, as the paging delay value increases, the R_{OE} also increases. When the paging delay is large, the network system partitions the paging area into large subareas and pages step by step. This large step paging reduces the paging cost. This figure clearly demonstrates that the performance of the MBR that uses the modified 1–D random walk is overestimated.

5 Conclusions

The original 2-D random walk model has many states, which make it difficult to calculate the performance evaluation of a system. In the previous study, it is assumed that the complex 2–D random walk can be reduced to a simple 1–D random walk with one barrier state. In this study, we evaluated the performance of the MBR scheme using two random walk models to compare the accuracy of the two random walk models. Analytical results are provided to demonstrate that the modified one–dimensional random walk always overestimates the performance of

the MBR and that the two–dimensional random walk model should be used to obtain the exact performance of the MBR and other registration schemes.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (R05–2004–000–11569–0).

References

[1] Akyildiz I.F., Ho, J.S.M., Lin Y.B.: Movement–based location update and selective paging for PCS networks. IEEE/ACM Trans. Networking, **Vol. 4** (1996) 629–638
[2] Baek, J.H., Ryu, B.H.:An improved movement–based registration in personal communication system networks. IEICE Trans. Comm., **E83-B(7)** (2000) 1509–1516
[3] Tseng, Y.C.,Hung W.N.: An improved cell type classification for random walk modeling in cellular networks. IEEE Comm. Letters, **Vol. 5. No. 8** (2001) 337–339
[4] Ho, J.S.M., Akyildiz, I.F.: Mobile user location update and paging under delay constraints. ACM-Baltzer J. Wireless Networks, **Vol. 1. No. 4** (1995) 413–425
[5] Rose, C., Yates, R.: Paging cost minimization under delay constrains ACM–Baltzer J. Wireless Networks, **Vol. 1. No. 4** (1995) 211–220
[6] Chung, Y.W., Sung, D.K., Aghavami, A.H.: Effect of uncertainty of the position of mobile terminals on the paging cost of an improved movement–based registration scheme. IEEE Trans. Comm., **Vol. E86-B. No. 2** (2003) 859–861
[7] Baek, J.H., Seo, J.Y.: Modeling and performance evaluation of unidirection-based registration with line–paging for cellular networks, International journal of industrial engineering–theory applications and practice, **Vol. 10. No. 4** (2003) 519–524
[8] Tung, T., Jamalipour, A.: A novel sectional paging strategy for location tracking in cellular networks, IEEE Comm. Letters, **Vol. 8. No. 1** (2004) 24–26

Time-Out Bloom Filter: A New Sampling Method for Recording More Flows

Shijin Kong[1], Tao He[2], Xiaoxin Shao[1], Changqing An[2], and Xing Li[2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, P.R. China, 100084
ksj00@mails.tsinghua.edu.cn, sxx03@mails.tsinghua.edu.cn
[2] China Education and Research Network (CERNET), Beijing, P.R., China, 100084
hetao@cernet.edu.cn, acq@tsinghua.edu.cn, xing@cernet.edu.cn

Abstract. Packet sampling is widely deployed to generate flow records on high speed links. However, random sampling in which 1 in *N* packets is chosen suffers from omitting majority of flows, most of which are short flows (within *N* packets). Although usage-based applications work well by sampling long flows and neglecting short ones, there are many other applications which depend on nearly per-flow information. In this paper, a novel sampling method is proposed to remedy the flow loss flaw. We use a Time-out Bloom Filter to alleviate the sampling bias towards long flows. Compared with random sampling, short flows have a much greater probability to be sampled while long flows are always sampled, but with much fewer sampled packets. Experimental results show that, with the same sampling rate, our solution records several times more short flows than random sampling. Particularly, up to 99% original flows can be retrieved. Besides, we also propose an adaptive TBF system in fast SRAM to perform online sampling.

1 Introduction

With the increasing network traffic, most measurement systems employ packet sampling to reduce resource consumption. First, there is not much resource remained for data collection on routers and switches because of their heavy workload. Forming flow statistics on a sampled substream of the original traffic reduces memory consumption and frequency of flow lookups. Second, transmitting sampled data to collectors, which is common for many applications, can greatly save collection bandwidth as well as processing and storage cost at collectors. The burden on routers and switches is alleviated at the same time since the volume of sampled data is small.

However, random sampling in which 1 in *N* packets is chosen causes great loss of flow information and it is difficult to recover. A flow is defined as a set of packets with a same key which consists of some fields in packet header. If any packet of a flow is sampled, we call this flow is sampled. In random sampling, a short flow which has fewer than *N* packets is easily *lost* if none of its packets is sampled, and a long flow has a much greater probability to be sampled. The bias towards longer flows causes majority of short flows lost, and thus brings a great total flow loss since most flows in traffic are short flows (e.g. 82.3% in the trace for our experiments).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 590 – 599, 2006.

No sampling method has been designed to meliorate the *flow loss flaw* in random sampling. Many usage-based applications work well by sampling long flows and neglecting short ones [13] because of the heavy-tailed distribution of flow lengths, that is, most traffic is carried by a small proportion of long flows [1]. However, there are still a lot of applications which depend on other detailed per-flow information instead of flow lengths. Here are some examples.

***Attacks Detection*:** information of short flows is very important to detect network intrusions such as port scanning and SYN flooding. These attacks usually consist of numerous short flows with only several packets. It is hard to discover these attacks unless nearly per-flow state is maintained [16], including flow key and correct count of SYN/FIN flags. Moreover, identifying a victim requires enough flow records.

***Traffic Identification*:** in particular, P2P traffic can be effectively identified by using connection patterns [2] instead of checking payloads of packets. This method counts <IP, Port> pairs retrieved from the flow information. Any obvious loss of flow records will cause fallacious counting results and hurt identification accuracy.

***Network Deployment Characterizing*:** the diversity of flow records reflects the spatial distribution of flows in the network. Workloads of network devices (e.g. the size of route tables) can be balanced according to the distribution, which helps to deploy and manage network more efficiently.

In order to meet those requirements, we present a novel packet sampling method in this paper to meliorate the flow loss flaw. Little attention has been paid to exact values of flow lengths as they are not useful in those applications. In our solution, the number of total sampled flows is increased by recording more short flows which are lost in random sampling. Packets are selected by a data structure called Time-out Bloom Filter (TBF). In TBF, some packets can have a great probability to be sampled, and others are definitely discarded. For short flows, considerable proportions of their packets have such a probability. But for long flows, the proportions are very small. Thus, our solution has a smaller sampling bias towards long flows than random sampling. Experimental results show that our method can sample several times more short flows than random sampling with same *sampling rate* (the ratio of sampled packets number to original packets number). Up to 99% of total original flows can be retrieved from sampled data while random sampling only records 37%.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 proposes our sampling method and section 4 makes the comparison with random sampling. Results on experimental traces are presented in Section 5 and section 6 proposes an adaptive TBF system. Finally, section 7 concludes the whole paper.

2 Related Work

In this section, we review previous work on flow-related sampling, and *hash-based* applications which are similar to ours.

Classical uniform sampling methods like random sampling are reviewed in Duffield's paper [3]. Flow records on randomly sampled packets are commonly generated by Cisco NetFlow [4] with configurable sampling period *N*. In [5], an adaptive NetFlow with dynamic sampling rate is devised, and one of its primary contributions is to

give accurate numbers of non-TCP flows. Sampling methods for long flows such as smart sampling [6, 7] can give an accurate estimation of total usage for each flow size. Although per-flow detailed information can not be told by these methods, research has been done to estimate the distribution of flow lengths [8, 9].

Hash-based sampling methods have also been applied for several purposes. Trajectory sampling [10, 11] puts particular flow keys in hash tables. Later packets with those keys are selected at each node to detect spatial distribution of flows. Space-Code Filter [12] uses a group of Bloom Filters with different resolutions to estimate flow length of any given key. Multistage Filter [13] selects packets based on several hash functions to identify large flows. And Partial Completion Filters in [14] propose a scalable solution to detect network attacks.

3 Our Sampling Method

In this section, we introduce Time-out Bloom Filter for packets sampling. TBF is derived from standard Bloom Filter [15]. A BF is a hash table with m bits, denoted as $b[0]$, $b[1]$, …, $b[m\text{-}1]$, each of which can be 0 or 1. There are d independent hash functions, $h_1(x)$, $h_2(x)$, …, $h_d(x)$, attached, and each of them maps a given key to one of the m bits. To insert a key c into the table, all the d bits $b[h_1(c)]$, $b[h_2(c)]$, …, $b[h_d(c)]$ are set to 1. Initially there are n keys in the table. Later, the table is used to check whether a given key c' has been inserted. If $b[h_1(c')]$, $b[h_2(c')]$, …, $b[h_d(c')]$ are all set to 1, c' is recognized as one of the n initial keys, otherwise it is not. On average, BF has a complexity of O(1) for querying a given key, which is much faster than traditional hash method with complexity O(m). However, since different keys can have same values calculated by hash functions, each bit can be set by several keys. If all the d bits of a non-initial key have already been set to 1 by initial keys, it will be mistaken as one of them. This mistake is called a *false positive* error.

Similarly, TBF is used to tell whether a packet can be sampled. Fast query of BF is retained and the false-positive error is reconsidered in TBF. Section 3.1 shows the principles of TBF and we explain the motivation for designing TBF in Section 3.2.

3.1 Time-Out Bloom Filter

TBF is similar to BF except that it does not have m bits, but m buckets instead, each of which contains a timestamp. The m timestamps are denoted as $t[0]$, $t[1]$, …, $t[m\text{-}1]$. Besides, it has a bucket time-out value t_0.

When a new packet with key c and timestamp t comes, the d timestamps, $t[h_1(c)]$, $t[h_2(c)]$, …, $t[h_d(c)]$, are compared with t. If any of the d timestamps, the ith for example, follows $t - t[h_i(c)] > t_0$ (or we say $b[h_i(c)]$ gets time-out), the packet is sampled, otherwise it is discarded. After comparison, all those d timestamps are updated to t even if the packet is not sampled. Figure 1 illustrates this process with d=3.

TBF differs from BF in two aspects: **(1)** TBF sets a timestamp for each bucket instead of simply setting it to 1, and updates the timestamps after every packet selec tion; **(2)** TBF samples a packet as long as any of the d buckets gets time-out while BF requires that all the d bits are set to 1.

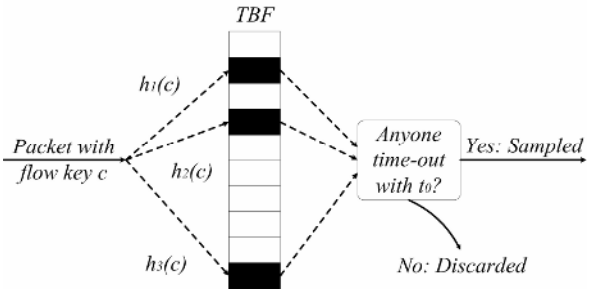


Fig. 1. A Time-out Bloom Filter with d=3

3.2 Why Time-Out Bloom Filter

Our initial motivation is to sample more flows, so ideally for each flow at least one packet should be sampled. A simple solution is to select the first packet of each flow and discard all the rest packets of it. In this process, we do not have to know any more information of the flow (e.g. IP addresses, ports). The only thing we want to know is whether an incoming packet belongs to an active flow or it is the first packet of a new flow. BF can tell the result quickly. We put all the keys of existed flows in BF and query it every time when a packet comes. According to the query result, an incoming packet is discarded if it is already in BF, or otherwise it is sampled as the first packet of a new flow. Hence a packet is sampled if any of the m bits is not set to 1. This is the origin of the second difference between BF and TBF mentioned in Section 3.1.

Unfortunately, there are two fatal drawbacks using BF.

(1). Not all first packets can be sampled because false positive error happens to mistake the first packet of a new flow for a packet of an existed flow. When this error occurs, the first packet will not be sampled and this flow is lost.

(2). More seriously, the longer the sampling is performed, the more existed flows are put into BF. Finally all m bits will be set to 1. A full-filled BF causes false positive error in every query, rejecting packets of newly generated flows to be sampled. A solution to this drawback is to clear the BF periodically, but the filter will be full-filled quickly on high-speed links (usually less than one second). There is not enough time for any practical sampling implementation to clear BF every second.

Thus TBF is applied to meliorate those drawbacks. In TBF, a bucket getting time-out can be viewed as a bit set to "0" and a bucket not getting time-out as a bit set to "1". When an incoming packet arrives at time t, only flows that have packets updated within $[t\text{-}t_0, t]$ are still kept in the filter. All other buckets are logically set to "0". As time elapsing, the "1" to "0" transforming process is automatically executed. If t_0 is small enough, TBF is never full-filled with "1". Hence drawback **(2)** is avoided.

Each flow can have multiple packets sampled in TBF rather than exactly one. When the time interval between two continuous packets of a flow is smaller than t_0, the latter one is definitely discarded since none of its d buckets gets time-out. When the interval is greater than t_0, the latter one may not be discarded. As long as any of those buckets is not updated by other flows during the interval, the packet is sampled,

or otherwise a false positive error occurs. A flow is lost only if all its packets encounter false positive errors, which greatly meliorates drawback **(1)**. But as a result, redundant sampled traffic is generated because flows unnecessarily have multiple packets sampled.

4 Comparing Sampling Methods

In this section, we compare our sampling method based on TBF with random sampling, and then explain why our sampling method can sample more flows.

4.1 Random Sampling: Sampling Bias Towards Long Flows

Assuming that 1 in every N packets is selected by random sampling, it is easy to figure out the probability of a packet to be sampled is

$$P_s=1/N \tag{1}$$

Let $F(k)$ denote all the flows with length k and $M(k)$ denote the number of $F(k)$. On average, $k\mathrm{M}(k)/N$ of packets of $F(k)$ are sampled. So the proportion of sampled flows of $F(k)$ has a minimum $1/N$ when $M(k)/N$ of $F(k)$ have one packet sampled for each, and a maximum k/N when k/N of $F(k)$ have all their packets sampled.

Usually, N is greater than 10 so that for k much smaller than N, the proportion of sampled flows is very small. For longer flows with k much greater than N, almost all of them are sampled. The discrepancy of sampling probabilities represents the sampling bias towards long flows.

4.2 TBF: Greater Packet Sampling Probability and Less Biased Sampling

First, we use the conclusion of BF in [15]. The probability that a bucket gets "1" is

$$P_1=1-(1-1/m)^{Ld} \tag{2}$$

where L is the number of $S(t_0)$, the set of flows sampled during previous t_0 interval. L can be measured by placing an empty TBF on the link for t_0 time and counting the flows in it. For example, set m=65,536, d=3, t_0=0.2s, then L for the trace in Section 5 is 5,882 on average (we divide the trace into thousands of t_0 periods, and measure L on each t_0 interval to get the average). Thus, a typical value of P_1 is 0.23. If L does not vary much whenever it is measured, P_1 is always around 0.23.

Table 1. P_1' and P_1 with m=65,536, t_0=0.2s and several d

d	1	2	3	4
P_1'/P_1	0.14/0.09	0.18/0.16	0.27/0.23	0.32/0.30

Now, we suppose a packet with a key c, which does not belong to $S(t_0)$, comes. For each i ($1\le i\le d$), one would easily expect the probability that $b[h_i(c)]$ gets "1" (denoted as P_1') is P_1. However, that's not the case. P_1' is usually greater than P_1. We think this

inconsistency appears due to the correlation among packets. For example, some applications start several flows with same source IP address at nearly the same time. Any hash function only uses source IP address for calculation will map the packets of all these flows to a same bucket. As long as any of these flows is in $S(t_0)$, P_1' for packets of other flows is definitely 1 rather than P_1. Besides, there are many other reasons that cause P_1' to be 1. So on average, P_1' is greater than P_1.

To minimize the correlation, hash functions must be carefully chosen. A hash function should use all fields of flow keys for calculation. Thus, even if two flow keys, c_1 and c_2, are only different in one field, $h_i(c_1)$ and $h_i(c_2)$ are not the same for each i ($1 \le i \le d$). We use such a group of *all-fields-dependent* hash functions in Section 5, and P_1' is measured and compared with P_1 in Table 1. As we see, P_1' is very close to P_1, especially when d >1. The correlation is effectively avoided. Thus, if a packet does not belong to $S(t_0)$, the probability that any of its d buckets gets "0" (time-out) is

$$P_s = 1 - (P'_1)^d \approx 1 - P_1^d \tag{3}$$

P_s is small if d is set to 3 or larger. Again with the example mentioned above, P_1 is 0.23 and P_s is 0.98. It is *much greater* than that of random sampling (in Equation 1).

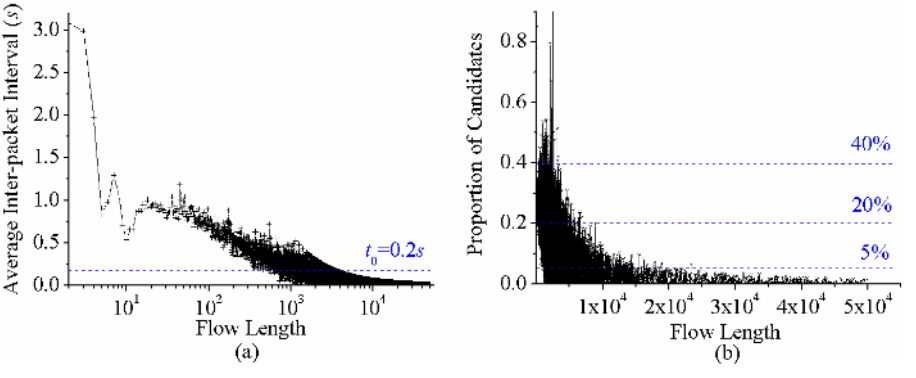


Fig. 2. (a) Average inter-packet interval (b) The proportion of "candidates" for t_0=0.2s

We define a "candidate" as a packet that has a probability P_s to be sampled. A packet that is definitely not sampled is not a "candidate". In random sampling, all the packets are "candidates". In TBF, however, only if the interval between two continuous packets in a flow is greater than t_0, the latter one can be a "candidate". In Figure 2(a), we can see that the longer the flow, the shorter the average inter-packet interval is. If a proper t_0 is selected, there are small proportions of "candidates" in long flows (<5%) but large proportions of "candidates" in short flows (20% to 40%), as shown in Figure 2(b) with t_0=0.2s. Compared with random sampling, the bias towards long flows is *reduced* since fewer packets of long flows are sampled.

In summary, our solution differs from random sampling in two aspects. **(1)** The probability of any packet to be sampled (P_s) is much greater. **(2)** The bias towards longer flows is alleviated because longer flows have fewer proportions of "candidates" than shorter flows. These two aspects determine that TBF can retrieve much

more flows from sampled data with same volume (or the same sampling rate) than random sampling.

5 Results of Comparison on Traces

This section shows the experimental results of TBF and its comparison with random sampling. We use a trace containing 681,268,937 packets captured from a gigabytes link, one of the outlets of THUNET (TsingHua University NETwork). It begins at 13:00, Aug 4, 2005 and lasts for an hour. Each packet is recorded with a flow key including four fields in IP header: source IP address, source port, destination IP address and destination port. A flow is terminated if none of its packets arrives within 30 seconds. Totally 8,475,966 flows are generated in the trace.

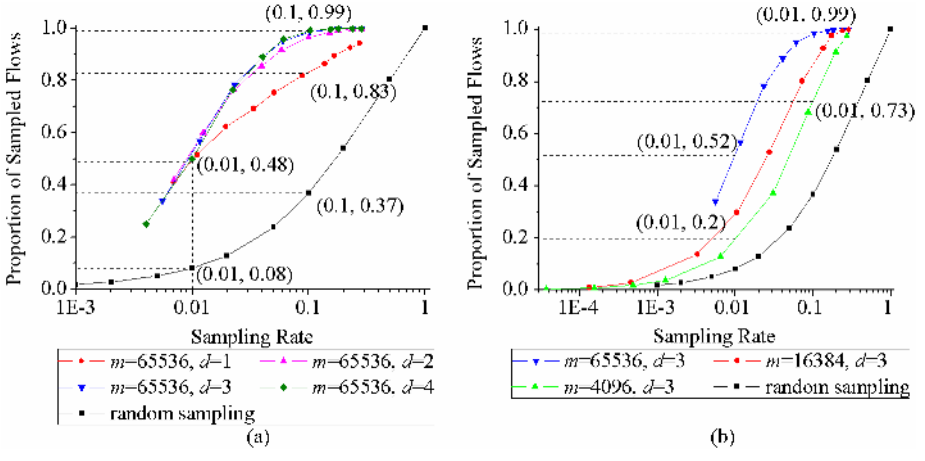


Fig. 3. The proportion of flows sampled by: (a) TBF with fixed m=4 (b) TBF with fixed d=3.

First, we perform random sampling 10 times on the trace with N=1, 2, 5, 10, 20, 50, 100, 200, 500, 1000 respectively. The sampling rate for random sampling is simply 1/N. Then we perform TBF sampling with different m, d and t_0. In our experiments, we simply use different combinations of mask on IP addresses and ports as all-fields-dependent hash functions.

Figure 3(a) shows the proportion of total sampled flows by TBF against the sampling rate with fixed m=65,536 and d=1, 2, 3, 4. Figure 3(b) shows the results with fixed d=3 and m=4,096, 16,384, 65,536. For each combination of m and d, TBF are performed 10 times with t_0=0.01s, 0.02, 0.05s, 0.1s, 0.2s, 0.5s, 1.0s, 2.0s, 5.0s, 10s respectively. The sampling rate decreases *monotonically* while t_0 increases.

As we can see, TBF samples much more flows than random sampling with same sampling rate. When the sampling rate is 0.1, TBF with m=65,536 and d=3 records 99% of original flows while random sampling (N=10) records 37% of them. The result of TBF is consistent with Equation 3: each flow at least has one candidate (the first packet), so at least P_s=98% of original flows can be sampled. Some flows have

multiple candidates, thus results in a higher 99% on the whole. When sampling rate is smaller, say 0.01, random sampling only samples 8% but TBF still records 48%.

Either increasing m or increasing d helps to enhance the number of sampled flows while the sampling rate is kept unchanged. But increasing m is more effective than increasing d. For sampling rate 0.01, varying d from 1 to 4 gives 16% more sampled flows while by changing m from 4,096 to 65,536, 26% extra flows are recorded. We also notice that for d=3 and d=4, the two curves have already overlapped. It means there is no more gain by adding more hash functions.

In Figure 4(a), proportions of sampled flows of $F(k)$ $(1 \le k \le 20)$ are shown. For each length k, TBF with m=65,536, d=3 and t_0=0.2s records more than P_s=98% of $F(k)$. The sampling rate of this TBF is about 0.1, so it is compared with random sampling N=10. When k is fewer than five, the number of sampled flows of $F(k)$ recorded by TBF is at least three times more than that by random sampling. To give a clearer view, various flow counts for each k are presented as a percentage of the total original flow count in Figure 4(b). As we expect, the sum of short flow counts, $\Sigma M(k)$ $(1 \le k \le N=10)$, represents majority of the total original flow count, 82.3% exactly.

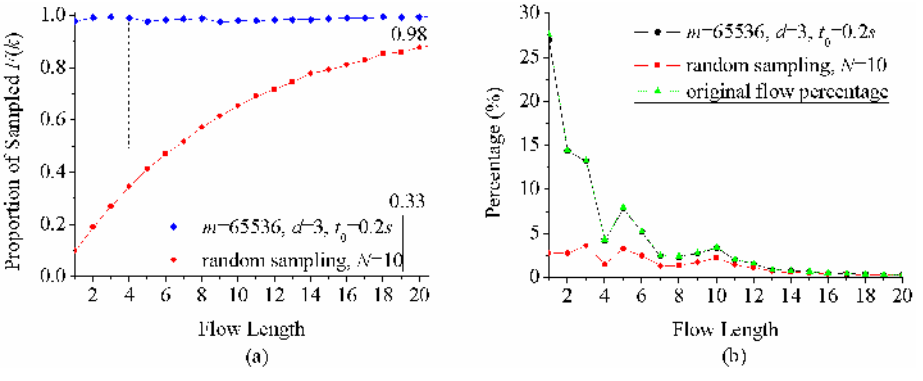


Fig. 4. Sampled flows distributed in flow lengths ranged from 1 to 20, in proportion to (a) $M(k)$ (b) the number of total original flows

6 Adaptive TBF Sampling System

In this section, we focus on implementing adaptive TBF sampling systems. To perform TBF sampling effectively, m, d and t_0 should be carefully chosen.

Although m should be as great as possible, it is limited by memory. In our implementation, we only use a 256KB SRAM for TBF. Since t_0 is at a level of 0.1s, each bucket records its updating time in milliseconds to keep accuracy. We set m=65,536, and each bucket has 2bytes to store the last 16 bits of packet arriving time in milliseconds. That is $2^6 \times 2^{10}$=64 seconds. To our experience, all buckets are updated within every 64 seconds, so a bucket getting time-out will not be mistaken as NOT-TIMEOUT. See the pseudo code below.

```
CHECK_BUCKET_TIMEOUT(timestamp t, bucket i)
if (t > t[i] AND t - t[i] < t0
    OR t < t[i] AND t + 64,000-t[i] < t0)
  return NOT-TIMEOUT;
endif
return TIMEOUT;
```

d is usually set to 3 or 4 as in Section 5 we analyzed that there is no gain to further add hash functions. On the other hand, for every packet, d hash values are calculated. If d is smaller, per-packet process is faster. We set d=3 and continue using the all-fields-dependent hash functions devised in Section 5.

When m and d are determined, t_0 is used to make a tradeoff between the sampling rate and the proportion of flows sampled. If t_0 is small, P_s is great (in Equation 3) and lots of flows are retrieved from redundant sampled data. If t_0 is large, low P_s causes few sampled flows but low sampling rate. In real applications there always exists a target for performance, say, recording more than 90% flows or sampling no more than 5% packets, which helps to choose a proper t_0.

In our implementation, we set the target as sampling no more than 5% packets. The network traffic varies all the time, and a fixed group of m, d, t_0 will cause the sampling rate much higher or lower than 5% sometimes. So we devised an adaptive TBF sampling system to keep sampling rate around the target. In the following pseudo codes, t_0 is adjusted based on the sampling rate measured every five seconds. If the sampling rate is over 0.05, t_0 is set larger. If the sampling rate keeps below 0.05 for 3 continuous measurements, t_0 is set smaller.

```
ADAPTIVE_TBF_SAMPLING
if (sampling_rate > target
    OR sampling_rate < target for 3 measurements)
  t0 = t0 * (sampling_rate / target)1/2;
endif
```

We already have a gigabytes measurement system based on Intel IXP2400 network processor to capture packet headers from one of THUNET outlets. To test the system for online sampling, we implement it on a host which receives the captured packet headers from the measurement system. In Figure 5, a one-hour result of the system is shown. On the whole, about 75.2% flows are sampled with sampling rate 0.05.

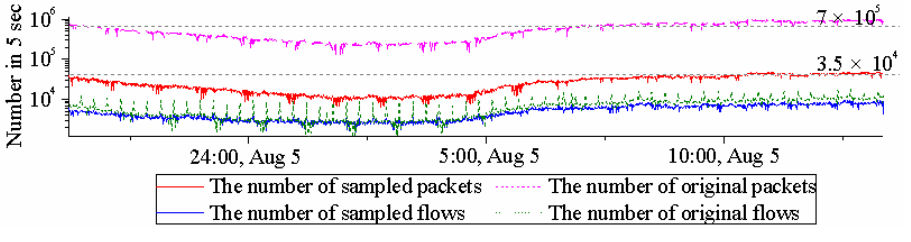


Fig. 5. One-hour result of adaptive TBF sampling system

7 Conclusion and Future Work

In this paper, we propose a novel sampling method based on Time-out Bloom Filter to remedy the short flow loss flaw in random sampling. We analyze the motivation of using TBF for packets selection and the reasons for its smaller sampling bias towards long flows. By comparing with random sampling on network trace, we find that TBF sampling can record up to 99% of total original flows and several times more short flows than random sampling. We have also discussed the proper choices of parameters and then devised an adaptive TBF sampling system for online sampling. In the future work, we are going to program the algorithm on IXP2400 network processor to integrate TBF sampling with the measurement system.

References

1. A. Feldmann, J. Rexford, and R. Cáceres. Efficient Policies for Carrying Web Traffic over Flow-switched Networks. IEEE/ACM Transactions on Networking, 6(6): 673 - 685, 1999.
2. T. Karagiannis, A. Broido, M. Faloutsos, et al. Transport Layer Identification of P2P Traffic. ACM SIGCOMM Internet Measurement Conference (IMC), 2004.
3. N.G. Duffield. Sampling for Passive Internet Measurement: A Review. Statistical Science, 19(3):472 - 498, 2004.
4. Ciso NetFlow. http://www.cisco.com/warp/public/732/netflow/index.html.
5. C. Estan, K. Keys, D. Moore, et al. Building a Better NetFlow, ACM SIGCOMM, 2004.
6. N.G. Duffield, C. Lund, and M. Thorup. Charging from Sampled Network Usage. ACM SIGCOMM Internet Measurement Workshop (IMW), 2001.
7. N.G. Duffield and C. Lund. Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure. ACM SIGCOMM IMC, 2003.
8. N.G. Duffield, C. Lund, and M. Thorup. Estimating Flow Distributions from Sampled Flow Statistics. ACM SIGCOMM, 2003.
9. N. Hohn and D. Veitch. Inverting Sampled Traffic. ACM SIGCOMM IMC, 2003.
10. N.G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. IEEE/ACM Transactions on Networking, 9(3):280 - 292, 2001.
11. N.G. Duffield and M. Grossglauser. Trajectory Engine: A Backend for Trajectory Sampling. IEEE Network Operations and Management Symposium, 2002.
12. A. Kumar, J. Xu, J. Wang, et al. Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement. IEEE INFOCOM, 2004.
13. C. Estan and G. Varghese. New Directions in Traffic Measurement and Accounting. ACM SIGCOMM, 2002.
14. R.R. Kompella, S. Singh, and G. Varghese. On Scalable Attack Detection in the Network. ACM SIGCOMM Internet Measurement Conference, 2004.
15. B.H. Bloom. Space/time Tradeoffs in Hash Coding with Allowable Errors, ACM Communicationsm 13(7), 1970.
16. K. Levchenko, R. Paturi, and G. Varghese. On the Difficulty of Scalably Detecting Network Attacks. ACM CCS, 2004.

A Performance Analysis Modeling of a QoS-Enabled Home Gateway

Ssang Hee Seo[1], Jung Tae Lee, and Kyung Jae Ha[2]

[1] Pusan National University, 30, Jangjeong Dong, Geumjeong Gu, Busan, Korea,
shseotwin@pusan.ac.kr,
[2] Kyungnam University, 449, Wolyong Dong, Masan, Kyungnam, Korea

Abstract. This paper presents a queueing model of a QoS-enabled home gateway. We use to PC based software router as home gateway. The proposed model is M/G/1/K processor sharing. The arriving process is assumed to be Poisson Process which is independent, identically distributed. The service time distribution is general distribution and the service discipline of server is processor sharing. Also, the total number of packets K that can be processed at one time is limited to K. We obtain performance metrics of software router such as blocking probability, throughput and system sojourn time. Validation results show that the model estimates the performance of the home gateway.

1 Introduction

The home is evolving rapidly into a networked environment. As data communication technology continues to penetrate the home, the intelligent devices are emerging in the home. There is a need to interconnect these devices to one another and to the outside world. At the same time, broadband networking technologies such as digital subscriber line (DSL) and cable access are becoming widespread. These access technologies provide always on connectivity as well as enough bandwidth to be shared by multiple devices [1].

A home gateway is a device that interconnects various home network segments among themselves as well as Internet through one of the broadband access technologies. A home gateway have more requirements such as home directory service, remote control, A/V and data transmission, network address translator (NAT), firewall, print service, quality of service(QoS) support. Among these requirements, QoS support on the home gateway is an important issue. Specially, for real time applications, QoS is a very important aspect for content delivery from content provider to end users via the service provider domain. Most of these IP QoS technologies are designed to provide differentiated delivery services for individual flows or aggregates by adding some intelligence to the Internet and improving the best effort service. Recently, the QoS issue has been widely discussed in the core network area, and many standards, RFCs are available. The major core network QoS architectures include IETF Integrated Services [2], IETF Differentiated Services [3], MPLS QoS Support [4]. But, most current commercial

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 600–609, 2006.

products still lack such a mechanism. Also, mounted on the outside of the residence, the home gateway network interface unit (NIU) connects the cable drop to the various locations inside the home. Because it contains RF tuner, the home gateway has a significant impact on the overall system performance of the home network.

Following this, we investigate and build a simple performance model of a QoS enabled home gateway on communication systems. We use PC based software router as home gateway. There are various types of devices for home gateway. But, these devices are essentially required of expandability. The PC based home gateway is a good point since they can be easily programmed to support new functionality. We viewed it as a queueing network of a home gateway with one node. Such a simple queueing model like the M/M/1/K with First Come First Served (FCFS) service dis-cipline can predict a home gateway performance quite well. But, conceptually it is difficult to assume that the service distribution is exponential and that the service discipline is always FCFS. In this paper, we present M/G/1/K model for a home gate-way. The arrival process to the server is assumed to be a Poisson Process and the service time distribution be arbitrary.

We use software IP router running Linux 2.4 operating system. Also, we use Traffic Control (TC)[6], differentiated services (DiffServ) mechanisms and Class Based Queueing(CBQ)[7],[8] link shared scheduler in order to support Quality of Service (QoS) in Linux.

This paper is organized as follows. In section 2, we describe the architecture and operation and the path of a packet within Linux. In section 3, we explain the characteristics of queuing model. In section 4, we describe closed form expression of performance metrics for home gateway. In section 5, we show the results and the discussion. Finally, we discuss our conclusions.

2 Home Gateway Architecture

2.1 System Architecture

The overall system architecture consists of home gateway between the external broadband and the home network.

2.2 Functional Architecture

We show the functional architecture and the path of a packet within a Linux based home gateway in Fig. 2 [9],[10].

1. The packet arrives from the network. It is placed in hardware memory on the NIC. The Card issues a hardware interrupt. The processor executes the device driver and copies the packet from the card to the main memory into a structure.
2. This structure is queued in a FIFO manner in the input queue. There exists one such input queue per processor.

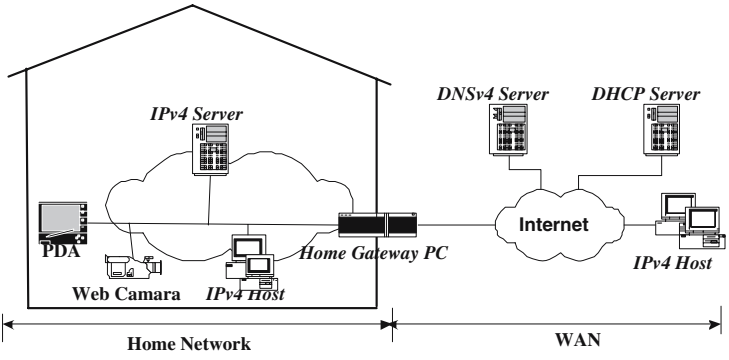


Fig. 1. The system architecture

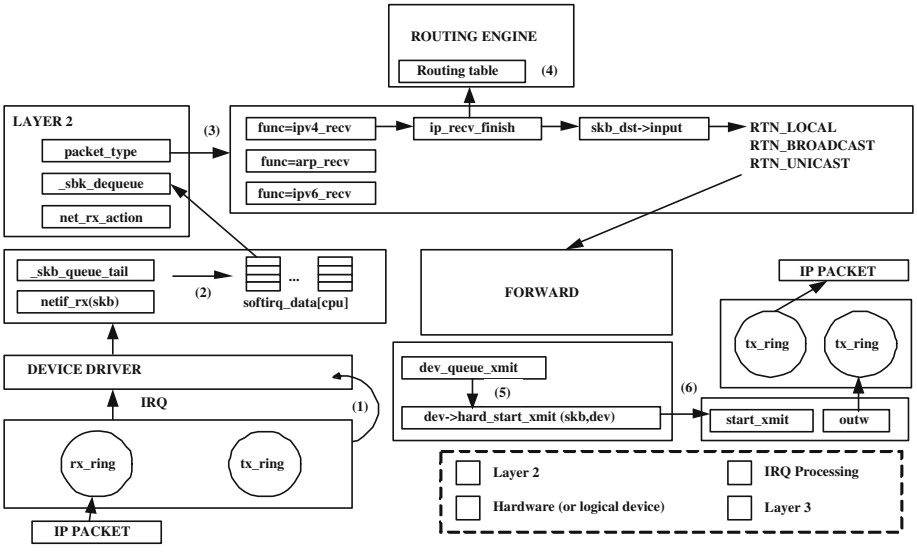


Fig. 2. The path of a packet in Home Gateway

3. Before returning from the interrupt, the driver issues a soft interrupt. Once the software interrupt is allowed to execute, the structure is dequeued and information of the ISO layer 2 is analyzed. According to the value of the type field, the packet is handed to the correct layer 3 function.
4. The destination IP address is extracted and the route cache is inspected. If the entry is not present in cache, a look up is performed in the routing table. The next hop information is recorded in the structure.
5. According to the routing decision, the forwarder is then queued to the correct outgoing interface. There exists one output queue per interface. At this point, Linux traffic control comes into play. Linux traffic control builds a complex

combination of queuing disciplines, classes and filters that control the packets. Then it sends the packets on the output interface.
6. The packet is transferred to the hardware memory and the card is instructed to send the packet on the network.

3 Queueing Model Characteristics

Fig. 3 illustrates the packet processing and the queuing model of proposed home gateway. This shows that home gateway is composed of two main parts. The first part consists of processors on network interface cards that have functions to receive and transmit packets. The second part is the CPU of the PC machine that forwards packets not destined for itself, and process packets destined for itself. Thus, the basic components of the home gateway are receiver, transmitter and router with traffic control. The following explains packet processing and

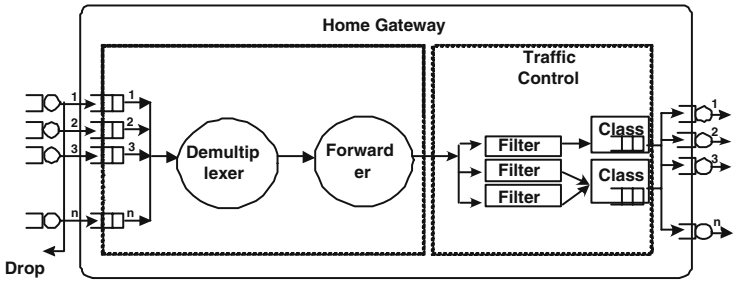


Fig. 3. A queueing model of home gateway

how to obtain the system sojourn time in models. The system sojourn time is one of the most important factors for performance metrics. When there is an incoming packet through the i^{th} interface ($i = 1, \cdots, n$), the receiver of that interface receives such a packet and stores it in the input buffer of Q_i if the input buffer is not full; otherwise it will drop those packets. In the M/G/1/K model, the receivers which do this are independent, identically distributed stochastic variables RT_i with mean rt_i. Once there is a packet in the output buffer of the j^{th} interface ($j = 1, \cdots, n$), the transmitter of that interface picks the packet from the output buffer and transmits it. The times for transmitters to do this are independent, identically distributed stochastic variable TT_j with mean tt_j. Through which interface a packet will be transmitted is dependent on the IP destination address of such a packet, and is random. Thus the average time of picking and transmitting a packet, tt, is defined as

$$tt = \sum_{j=1}^{N} P_j tt_j \tag{1}$$

A single service facility of the home gateway serves in a weighted round robin manner. The service times of packets of the i_{th} network interface are arbitrary, with first and second moments $E[S]$ and $E[s^2]$. They have different service times according to the type of traffic. Accordingly, the average time spent in the system by each packet through the i^{th} network interface $Q_i(i = 1, ..., n)$ is the expected sojourn time of a packet. This value is estimated from the measured average response time.

Finally, the system sojourn time $E[ST]$ spent in the system by each packet through the network interface can be defined as:

$$E\left[ST\right] = rt_i + E\left[W_i\right] + E\left[S_i\right] + tt = E\left[T\right] \tag{2}$$

$E[T]$ denotes the average response time and $E[W_i]$ denotes the mean waiting time of a packet in $Q_i(i = 1, ..., n)$. If the home gateway is directly connected, rt_i and tt are ignored.

4 Queueing Model of Home Gateway

4.1 M/G/1/K Processor Sharing Model

We model the home gateway using an M/G/1/K processor sharing queue. The packets arrive according to a Poisson process with rate λ. The average service time has a general distribution with mean $E(S)$. The $E(S)$ is the inverse of μ which is service rate. An arrival will be blocked if the total number of packets in the system has reached a predetermined value K. A packet in the queue receives a predetermined quantum of service and is then suspended until every other packet has received an identical quantum of service in a weighted round robin fashion. When a packet has received the amount of service required, it leaves the queue.

Thus, such a system can be viewed as a queuing network with one node [11]. We propose the M/G/1/K queueing model as the performance model for home gateway.

In the M/M/1/K FCFS model, the probability mass function (pmf) of the total number of packets in the system has the following expression, where ρ is the offered load and is mathematically defined as the packet arrival rate, λ , divided by the service rate, μ.

$$P[N = n] = \frac{(1 - \rho)\rho^n}{1 - \rho^{K+1}} \tag{3}$$

We note that an M/M/1/K FCFS queue has the same pmf as M/G/1/K processor sharing [12]. However the service time distribution of the M/M/1/K FCFS queue must be exponential and its service discipline must be FCFS.

From (3), we can derive the following performance metrics, average response time, throughput and blocking probability.

The probability of blocking $P_b$is equal to the probability that there are K packets in the system.

$$P_b = P[N = n] = \frac{(1-\rho)\rho^n}{1-\rho^{K+1}} \tag{4}$$

The throughput H is the rate of completed packets. When PC router reaches equilibrium, H is equal to the rate of accepted packets,

$$H = \lambda(1 - P_b) \tag{5}$$

The average response time $E[T]$ is the expected sojourn time of a packet. Following Little's law, we know that

$$E[T] = \frac{E[N]}{H} = \frac{\rho^{K+1}(K\rho - K - 1) + \rho}{\lambda(1-\rho^K)(1-\rho)} \tag{6}$$

$E[N]$ is the mean number of packets in system.

To get theoretical results we can make comparisons to, we can estimate λ and μ from measurements. So by a simple simulation program, we have attained theoretical results. The actual results will appear in section 5.

5 Experimental Result

5.1 Parameter Estimation

We estimate the parameters from the measurements. The result is presented in Table 1. Using the estimated parameters, we can predict the home gateway performance and compare it with the measurements.

Table 1. Parameter Estimation

Bandwidth (Mbps)	E[S] (sec)	Bandwidth (Mbps)	E[S] (sec)	K
0.1	0.0001916	1.0	0.0000190	10
0.2	0.0000960	1.1	0.0000173	10
0.3	0.0000640	1.2	0.0000173	10
0.4	0.0000476	1.3	0.0000146	10
0.5	0.0000383	1.4	0.0000153	10
0.6	0.0000316	1.5	0.0000126	10
0.7	0.0000273	1.7	0.0000153	10
0.8	0.0000240	1.8	0.0000110	10
0.9	0.0000213	2.0	0.0000103	10

We were interested in the following performance metrics: system sojourn time, throughput and blocking probability. The throughput is estimated by taking the number of bits per second between the total number of successful packets and

the time span of measurement. The system sojourn time is the time difference between when the source host sends a packet and when the receiver host receives the packet. This is the expected system sojourn time of a packet. The blocking probability is estimated as the drop ratio between when the source host sends a packet and when the receiver host does not receive the packet.

5.2 Test Environment

The test environment contains 3 PCs. The each one is equipped with a 100Mbps fast Ethernet interface. The home gateway is placed between PCs.

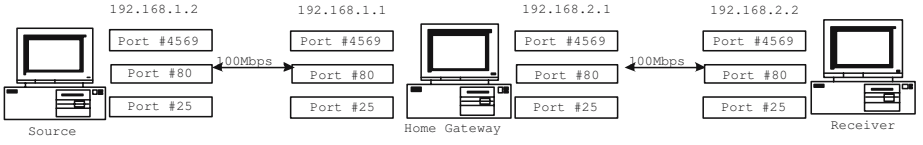


Fig. 4. Test Environment

The source host works on a 2.6GHz Pentium PC running Linux. The receiver host works on a 2.0GHz Pentium PC with Linux. The source and receiver host connect to 100Mbps Ethernet respectively and communicate with each other via the home gateway. The home gateway works on a AMD 2400+ MP running Linux. The traffic generator used is the iperf [13]. Offered bandwidth from 0.1Mbps through 2Mbps, and the duration of 60sec were used. We sent the three traffics to the network simultaneously. The table 2 lists parameters used for test. In order to QoS support, the different flows need different packet processing. And the rules have to be set on the home gateway. In this experiment, the classification is based on port number. Once the home gateway received a packet, it will extract the IP packet and will process differently according to the type of flows. The table 3 shows rules of this configuration.

Table 2. Test Parameters

Traffic Type	VoIP	UDP0	UDP1
Data Size per packet(bytes)	100	500	1000
Data Rate based on Bandwidth(%)	50	30	20

5.3 Result and Comparison

Using the estimated parameters, we could predict the home gateway and compare it with the measurements. Fig. 5, 6 and 7 show the average drop ratio, the average throughput, and average response time of M/G/1/K model. Fig. 5

Table 3. Rules for setting up the home gateway

Destination IP Address (port number)	Queue	CBQ Weight (packet for delivery each round)	Bandwidth Utilization Allowed
4569	Q0	5	50kbits/sec
5436	Q1	3	30kbits/sec
6473	Q2	2	20kbits/sec

shows that the packet drop was not occurred from 0.1M to 1.3M. The first drop of packets was started at 1.4M bandwidth. In case of 1.4M, the only UDP1 traffic was dropped. This means that the QoS mechanism was achieved. Since the UDP1 traffic has the lowest priority although the UDP1 traffic has the lowest sending rate. Also, UDP1 and VoIP traffic were dropped at 1.7M. But, the UDP0 traffic was not. In our analysis, it is cause of the drop of VoIP traffic. As greater bandwidth, the drop ratio of UDP0 and UDP1 traffic were increased than that of VoIP traffic. The average drop ratio of three traffics was similar to the average drop ratio of model.

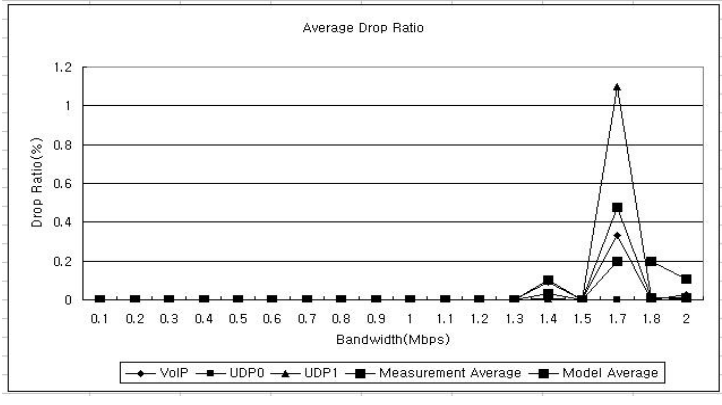


Fig. 5. Average Drop Ratio

The Fig. 6 shows the average throughput of three traffics. The throughputs of each of traffic are close to the allowed bandwidth respectively. The VoIP traffic had high throughput because of high priority. This figure shows that the home gateway supports the differentiated services. Also, the average throughput predicated by the model fit well to that of three traffics.

Fig. 7 shows the average response time. We estimated the average response time of model from experimental parameters. The obtained value had some difference. In our analysis, it is due to limited tcp and udp buffer size. The size of buffer influences the achievable response time. So, the average waiting time was increased by the limited buffer size. Fig. 7 shows the average response time

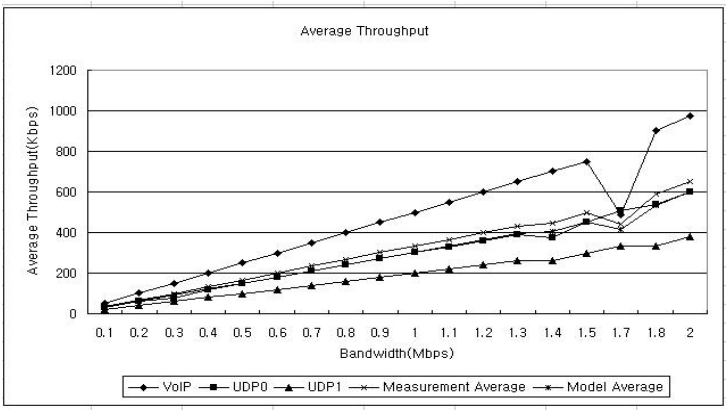


Fig. 6. Average Throughput

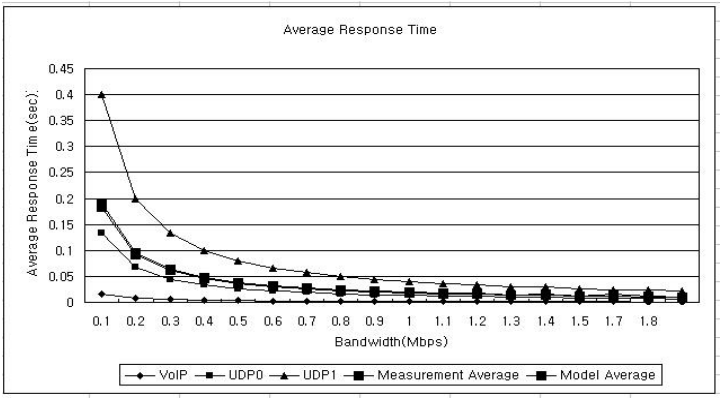


Fig. 7. Average Response Time

predicated by the model fit well to that of the experiment. As greater bandwidth, the average response time was decreased.

6 Conclusion

The QoS allows home networking applications to prioritize individual services. The QoS guarantee of home gateway is essential for entertainment based applications delivery over home networks. In order to analyze the performance of the QoS enabled home gateway, we present the M/G/1/K processor sharing queueing models. We have derived expressions for performance metrics such as system sojourn time, blocking probability and throughput using measured average response time. Moreover, we obtained the experimental results, which were similar to the results induced by our model. As a result of our experiment, we can verify

our model. Future works include additional measurements more accurate time synchronization using GPS. Also, a study of tcp and udp buffer size is needed to transfer large volumes of data. Also, we will study home gateway supporting IPv6-IPv4 translation because the IPv6-IPv4 translation is one of the important subjects in home network.

References

1. Gerard O'Driscoll. : The Essential Guide to Home Netwoking Technologies. Prentice Hall.(2001)
2. Braden, R. Clark, D. Shenker, S. : Integrated services in the Internet Architecture:An Overview. RFC 1633. June. (1994)
3. Blake, S. Black, D. Carlson, M. Davies, E. Wang, Z. Weiss, W. : An architecture for differentiated services. RFC 2475. IETF. Dec. (1998)
4. Rosen, E. Viswanathan, A. Callon, R. : Multiprotocol Label Switching Architecture. RFC 3031. Jan. (2001)
5. Leland, W. Taqqu, M. Willinger, W. Wilson, D. : On the self similar nature of Ethernet traffic (extended version). IEEE/ACM Trans. Networking. vol.2. no.1. (1994) 1–15
6. Werner Almesberger. : Linux Network Traffic Control - Implementation Overview. EPFL. ICA. (2001)
7. Nichols, K. Blake, S. : Differentiated Services Operational Model and Definitions. Internet Draft. Feb. (1998)
8. Black, D. Blake, S. Carlson, M. Davies, E. Wang, Z. Weiss, W. : An Architecture for Differentiated Services. Internet RFC 2475. Dec. (1998)
9. Saravanan Radhakrishnan. : Linux-Advanced Networking Overview. ITTC. University of Kansas.
10. Netherlabs, Gregory Maxwell, Remco van Mook Martijn van Oosterhout, Paul B Schroeder, Jasper Spaans. : Linux2.4 Advanced Routing Howto. TLDP.
11. King, P. J. B. : Computer and Communication Systems Performance Modeling. Prentice Hall. (1990)
12. Kleinrock, L. : Queueing Systems, Volume 1: Theory. John Wiley and Sons. (1975)
13. National Laboratory for Applied Network Research (NLANR). Iperf Version 2.0.2. http://dast.nlanr.net/Projects/Iperf/

Time-Driven vs Packet-Driven: A Deep Study on Traffic Sampling

Xiaoxin Shao[1], Tao He[2], Shijin Kong[1], Changqing An[2], and Xing Li[1]

[1] Department of Electronic Engineering, Tsinghua University,
Beijing, 100084, P.R. China
sxx03@mails.tsinghua.edu.cn, ksj00@mails.tsinghua.edu.cn,
xing@cernet.edu.cn
[2] China Education and Research Network,
Beijing, 100084, P.R. China
hetao@cernet.edu.cn, acq@tsinghua.edu.cn

Abstract. Traffic sampling technology has been widely deployed in front of many high-speed network applications to alleviate the great pressure on packet capturing. Packet-driven sampling mechanism is believed better than time-driven one and no in-depth comparison is given to these two kinds of mechanisms. In this paper, a systematic comparison is conducted on three sampling methods, 1/N packet-driven, 1/T time-driven and t/T time-driven samplings, with a result showing that t/T sampling achieves similar accuracy as 1/N sampling in most aspects, and surpasses 1/N sampling in estimating the interval time distribution. Then we try to optimize performance of t/T sampling by tuning its parameters, and verify putative conclusions with both real and simulation traffic. The experiment in a real measurement application also indicates that these two kinds of sampling mechanisms achieve similar online estimation performance.

1 Introduction

Both packet-level and flow-level network measurement are based on packet capturing. As link speed rises, processing overload increases under the demand of catching all of the packets transmitted on networks. Moreover, storage of traffic traces is also a big challenge. For instance, an OC-48 link will produce 180GB data per hour even if only a 64-byte record is kept for every packet[4]. It is hard to capture and record all the packets in high speed links.

To deal with high-speed traffic measurement, a variety of sampling methods have been proposed and widely deployed. After carefully comparing packet-driven sampling and time-driven sampling [1], Kimberly C. Claffy concludes that 1/N packet-driven sampling is much better than 1/T time-driven sampling. This conclusion is later followed by many other researchers. Much more methods have been improved based on packet-driven to achieve more accurate results[2]. A commonly used technique is making use of heavy-tailed behavior of network

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 610–619, 2006.

traffic, identifying and capturing only the packets within elephant flows[3]. However, less attention is paid to time-driven sampling, which could also achieve a good result and easily be implemented in some scenarios.

In this paper, we first present an improved time-driven sampling method, t/T sampling, and compare it with two traditional methods, 1/N packet-driven and 1/T time-driven samplings. Our experiments show that t/T sampling could get similar accuracy as 1/N sampling in most aspects and surpasses 1/N sampling when estimating the interval time distribution. Furthermore, we also address the parameter optimization of t/T sampling.

The rest of the paper is organized as follows. After reviewing the previous work on sampling in Section 2, we introduce the sampling methods and mechanisms used in this study in Section 3 and compare their accuracies in Section 4. Then in Section 5 we present a theoretical analysis on parameter tuning for t/T sampling. We implement 1/N sampling and t/T sampling in real measurement system and make a detailed comparison in Section 6. Finally We conclude the whole paper with Section 7.

2 Previous Work

As a fundamental component of network measurement, sampling technology has been extensively studied in previous work.

Kimberly C. Claffy, in her paper, studies the systematic, stratified random and simple random sampling methods [1]. She compares above three methods on 1/T time-driven and 1/N packet-driven mechanisms and shows that time-driven technique performs worse than packet-driven one.

Following Claffy's conclusion, people begin to focus on packet-driven sampling. *Sampled NetFlow*[7], a widely used flow-level measurement system, uses a 1/N systematic packet sampling. One of every N packets is sampled and then used to update the flow status for flow level measurement.

C. Estan et al. propose two algorithms for identifying large flows: *sample and hold* and *multistage filters*, which keep a constant number of memory references per packet and use a small amount of memory[3]. Later the 1/N systematic sampling is improved by adapting the sampling frequency to dynamic throughput[2], which decreases memory consumption and computation complexity.

Similar sampling goal is achieved by Duffield et al. [5]. They use *smart sampling* to select big flows, and at the same time discard small ones, which are not important in pricing. The selection rule is based on flow size, which can be the total packets or bytes in a flow according to different applications.

No more attention is paid to time-driven sampling until 2005, a new variation of static systematic sampling, *biased systematic sampling* (BSS), is devised by G. He et al.[6]. BSS makes more accurate estimations to the mean of the traffic. However, they only compare BSS with other time-driven sampling methods, without comparing it with packet-driven samplings.

3 Sampling Methods and Mechanisms

In this section, we try to clarify two conceptions, *sampling mechanism* and *sampling method*. Traffic trace is consisted of continuous elements. Sampling mechanism defines what an element is, and sampling method decides which elements are to be sampled.

Packet-driven and time-driven mechanisms are the most widely used mechanisms. Packet-driven mechanism views the trace as a sequential packet serial, in which a packet is an element. Time-driven mechanism views the trace as a time serial consisted of sequential time slots, in which a time slot is an element.

There are three basic sampling methods, simply random sampling, stratified random sampling and systematic random sampling. Since all three methods generate similar results[1], we choose systematic sampling as representative in this paper, which is easy to realize. Systematic sampling selects the *kth* element of every N elements.

A complete sampling implementation is the combination of one sampling method and one sampling mechanism. In this paper, we focus on three sampling implementations:

- **1/N packet-driven sampling:** It views the trace as a packet serial, in which a packet is an element. It picks one packet out of every N packets. The estimation of original traffic is simply multiplying N to captured packets.
- **1/T time-driven sampling:** It views the trace as a time serial, each 1-span time slot is an element. It selects one packet out of every T 1-span time slots. Therefore we can not know the exact sampling ratio in packets or bytes, and it is impossible to estimate some original traffic statistics, such as the whole throughput or packet number within a time slot.
- **t/T time-driven sampling:** It also views the trace as a time serial. Different from 1/T sampling, t/T sampling considers each t-span time slot as an element. It selects all packets within a t-span time slot out of every T/t t-span time slots. The estimation of t/T sampling is as easy as 1/N sampling by simply multiplying T/t.

4 Comparison of Three Samplings

4.1 Trace Description

The trace we used in this paper is called T1, which is obtained from Tsinghua Campus network. It is consisted of a set of 64-byte packet units. The content of each unit includes 14-byte basic information of the captured packet and first 50 bytes of the IP packet. Within the basic information, there is an 8-byte timestamp, which records a relative timestamp given by a generic capturing system [8]. The timestamp is obtained as soon as the whole packet arrives in the capturing system, with a frequency of 37.5MHz, which is the clock frequency of capturing system. Trace T1 has a duration of 1 minute, which is 2,250,054,291 in capturing system clock. The size of T1 is 739,225,600B. The average interval time between two adjacent packets of T1 is 195.4 in capturing system clock.

4.2 Comparison of Three Sampling Methods

First of all, we want to clarify that the main difference of the above sampling implementation does not exist in the overhead of algorithms themselves. Sampling is deployed to record some packets based on pre-defined rules, and the overhead of implementation varies in accordance with the application scenarios. For instance, it is easy to sample the packets transmitted on a low-speed network link, such as a 2Mbps DSL line, which only produces less than ten thousand packets in one second. But it is hard to do the same processing on a high-speed network link because sometime even network interface card could not awake to the arrival of packet before it was dropped. Packet-driven implementations were often used in short term traffic observation, but time-driven sampling is more suitable to deploy in a long term traffic capturing system. Periodically distributing the sampled packets could greatly facilitate researchers to analyze the traffic.

We use the same parameters as[1], packet size distribution and packet interval time distribution, to evaluate the effects of these three sampling methods. We also adopt the same evaluation method, ϕ metric in[1], to evaluate the degree of difference between sampled and original traces. ϕ metric compares the sampled and original distributions within a set of bins which span the range of whole trace, as defined in Equation (1), where B is the number of bins, E_i is the sampled distribution in the *ith* bin, and O_i is the original distribution in the *ith* bin.

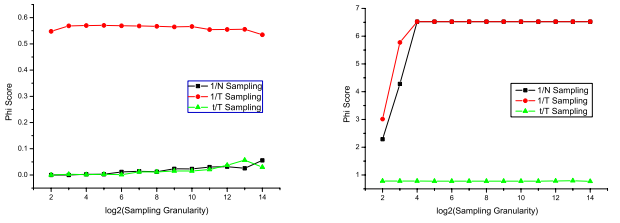
$$\phi = \sqrt{\frac{\chi^2}{n}}, n = \sum_{i=1}^{B}(E_i + O_i), \chi^2 = \sum_{i=1}^{B}\frac{(O_i - E_i)^2}{O_i} \quad (1)$$

Different from[1], our range bin selections of interval time and packet length are based on long term observation of Tsinghua Campus network. We set up 16 length bins, ranging from 0 to 1600 bytes with 100 bytes per step.

For interval time distribution, we set up 8 bins: less than 100 ($<2.67\mu s$), between 100 and 200 ($2.67\mu s$-$5.33\mu s$), between 200 and 300 ($5.33\mu s$-$8\mu s$), between 300 and 400 ($8\mu s$-$10.67\mu s$), between 400 and 500 ($10.67\mu s$-$13.33\mu s$), between 500 and 600 ($13.33\mu s$-$16\mu s$), between 600 and 700 ($16\mu s$-$18.67\mu s$), and greater than 700 ($>18.67\mu s$).

Figure 1 shows the ϕ scores of packet length and packet interval time distributions for three samplings on T1. All ϕ scores are the average of 4 trials. In t/T sampling, t=1024. The x-axis (in log2 scale) corresponds to sampling granularity. The granularity is N for 1/N sampling, while T/t for t/T sampling. When it comes to 1/T sampling, we set the granularity to the ratio of original trace size to sampled trace size.

For packet length distribution, 1/N sampling is much better than 1/T sampling, but almost the same as t/T sampling. When it comes to interval time distribution, t/T sampling behaves best. Because both 1/N and 1/T samplings catch discontinuous packets and heavily increase the interval time between each two packets, while t/T sampling capture continuous packets within each t slot, hence has a smaller error. Keeping records of continuous packets is an



(a) Packet length distribution (b) Interval time distribution

Fig. 1. ϕ scores of packet length and interval time distributions of T1

advantage of t/T sampling, and is useful for applications relying on continuous packets information.

There are two main drawbacks of 1/T sampling. Firstly, 1/T sampling samples a whole packet in each time slot T, whose time duration varies in the whole trace. Nevertheless 1/T sampling does not take these variations into account, but simply using 1 for all sampled packets to represent them. Therefore it is impossible to achieve an accurate estimation of original traffic, especially when the packet duration varies greatly in the trace.

Secondly, according to original trace statistics, small packets (less than 100 bytes) and big packets (greater than 1400 bytes) share similar percentages in whole trace. However, big packets have longer duration time, and their probabilities to be sampled in 1/T sampling are greater than small ones. Thus 1/T sampling is biased by big packets and can not keep the original packet size distribution in its sampled trace. This also explains why the packet length ϕ score of 1/T sampling in Figure 1 is extremely worse than other two samplings.

Comparing to 1/T sampling, t/T sampling gets a much better result because it uses a sub slot t instead of 1. Within the sub slot t, it samples a group of continuous packets, thus takes account into the packets' duration time and avoids missing small packets.

5 Optimal t and T For t/T Sampling

In this section we discuss how to set optimal parameters for t/T sampling. Sampling bias comes from two aspects, B_s and B_t. B_s is the bias caused by sampling granularity T/t. B_t is the bias related to sub time slot t.

As shown in Figure 1, sampling accuracy decreases with the increase of sampling granularity. Hence a proper granularity, T/t, can be chosen according to required accuracy. Given a granularity, once sub time slot t is decided, time slot T is also fixed. To choose a proper t, we present a theoretical analysis and verify our analysis with experiments on real traces.

5.1 Theoretical Analysis

B_t mainly comes from two aspects. Firstly, sampled packets may not be able to represent the real contents in t for operation reason. Secondly, if there is an

oversized interval between each two sampling actions, sampled trace can not approximate original trace precisely, since there may be big changes during each interval.

To select a proper t with an affordable B_t, we assume that all the values on each time point in the whole time serial are i.i.d random variables. Hence we only focus on one T slot.

t/T sampling selects packets whose timestamps are within the sub time slot t. Therefore the time duration from the beginning of the first sampled packet to the end of the last sampled packet may not be precisely t. Let Δt denote this time duration, which represents $\frac{\Delta t}{T}$ of this T slot. Then B_t is calculated with Equation (2). Next we discuss the problem according to different t.

$$B_t = \frac{|E_{real} - E|}{E_{real}} \times 100\% = \frac{|sample \times \frac{T}{\Delta t} - sample \times \frac{T}{t}|}{sample \times \frac{T}{\Delta t}} \times 100\% = \frac{|t - \Delta t|}{t} \times 100\% \tag{2}$$

- **t is much smaller than average interval:**
  if no packet is sampled in t, B_t equals to 100%. Otherwise only one packet is sampled, the time duration Δt is around the average interval, $B_t = \frac{|t-\Delta t|}{t} \rightarrow \frac{\Delta t}{t} \geq 100\%$. The range of bias is too large and unacceptable.
- **t is middle sized:**
  The upper bound of $|t - \Delta t|$ is the length of a packet interval. Therefore $B_t's$ upper bound is $\frac{aInterval}{t}$ according to Equation (2). B_t gets smaller with the increase of t. If t is greater than the average interval, the average B_t can be smaller than 100%. According to 4.1, t greater than 200 can get an acceptable B_t.
- **t is over sized:**
  Although greater t leads to a smaller B_t, the time interval between two sampling actions $(T - t)$ hurts the sampling accuracy. In t/T sampling, same granularity does not mean same interval between two actions. Given a granularity $G = T/t$, interval between two actions is $T - t = (G - 1)t$. Hence the increase of t leads to a corresponding increase of interval, which makes it not as accurate as small-interval sampling.

5.2 Experimental Results on Different t

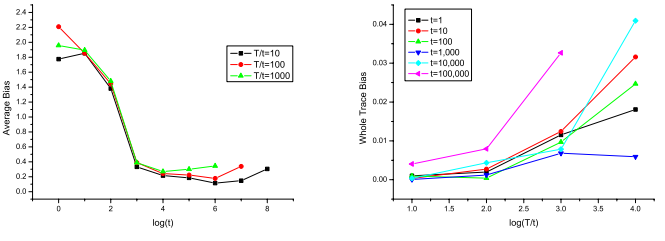
Firstly, we calculate the average number of sampled packets in sub time slot t on trace T1. As shown in Table 1, the average packet number sampled in t can not reach one unless t is 256 (greater than the average interval time between two adjacent packets), just as we have indicated in 5.1.

Next, we verify our analysis in 5.1. Figure 2 shows the average byte throughput bias and whole trace byte throughput bias of trace T1 with different combinations of t and T. All results are the average values of four trials. The left plot gives the average bias of all T slots with three sampling granularities. The x-axis (in log scale) denotes t. It proves that when t is smaller than the average interval

Table 1. Average packet number sampled in sub time slot t of trace T1

T/t	10	100	1,000	10,000	100,000
t=16	0.06	0.09	0.09	0.09	0.09
t=64	0.36	0.34	0.34	0.35	0.42
t=256	1	1	1	1	2
t=1,024	6	6	5	6	5
t=4,096	22	22	22	22	20
t=16,384	87	85	85	77	49

time of original trace (195.4), the average bias is above 100%. The right plot of whole trace bias shows that oversized t also leads to a great bias. Different from average bias, the whole trace bias of small t is not so great. This is because different T slots compensate each other, which leads to an acceptable final statistical result. However, the capturing under small t may have the same problem as 1/T sampling, which has a preference in capturing big packets.



(a) Average bias (b) Whole trace bias

Fig. 2. Byte throughput bias of T1

5.3 Simulation on Skewed Trace

After all the tests on smooth traces, now we evaluate the two samplings' performance on estimating skewed trace. The skewed simulation trace we use is generated by a stable sender (IXIA 1600 network traffic generator) by sending traffic at different rates. Figure 3 shows packet number versus packet timestamp of original trace and sampled traces after 1/N sampling and t/T sampling with two granularities, 100 and 1000. Each packet's timestamp in Figure 3 is the difference between its own timestamp and the first packet's timestamp in the trace. Different slopes correspond to different bandwidths. 1/N sampling approximates the original trace a little better than t/T sampling at the granularity of 1000. Because the T slot at granularity 1000 is too big for simulation trace, which contains only 44,416 packets. If the original trace had a longer duration, the result of t/T sampling would be much better. When granularity becomes smaller, the behaviors of two samplings are almost the same.

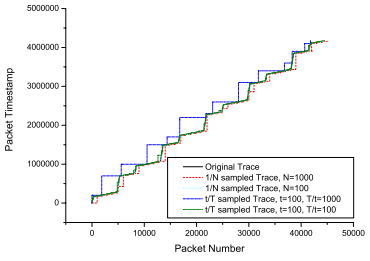


Fig. 3. Packet number VS timestamp

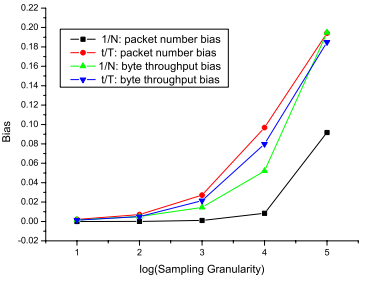


Fig. 4. Average bias of a fifteen minute trace. t=1024

In conclusion, t should be greater than the average interval of whole trace, but not oversized. Proper t and T can be chosen according to different accuracy demands.

6 Performance Evaluation of 1/N and t/T Sampling

We evaluate 1/N and t/T samplings' performance with real online system in this section.

The application scheme is shown in Figure 5, which is built inside CERNET (China Education and Research NETwork), the largest education network in China. An NP-based capturing system [8] is employed, which listens on the outgoing links of Tsinghua Campus Network to Internet, and provides pre-processed data traces to generic clients by multicast. We implement 1/N and t/T samplings in generic clients separately as well as a non-sampling client, which keeps original traffic for evaluation. All three clients share a same clock, and the two sampling clients write down part of all packets according to their sampling mechanisms. Then we compare each sampling's capturing result with the non-sampling client and obtain their performance in packet-level and flow-level statistics. In the rest of this section, we use the 15 minute traces recorded on each client simultaneously for comparison.

Firstly, we evaluate the biases between sampled trace and non-sampled trace in packet number and byte throughput. The biases are calculated every 10

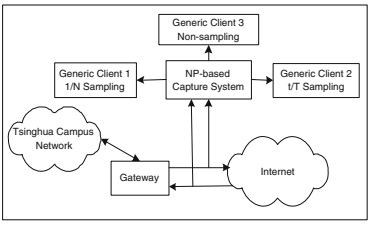
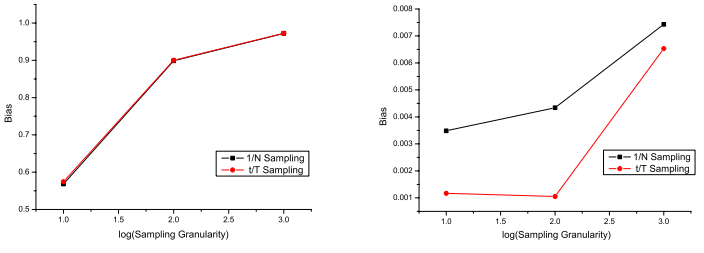


Fig. 5. Application Environment

seconds. Average biases of all 10-second units are shown in Figure 4. To be more accurate, we run the whole experiment two times. The x-axis (in log scale) corresponds to sampling granularity. Similar results are found in byte throughput estimation. When it comes to packet number estimation, 1/N sampling behaves better since it is based on packet count. For both metrics, sampling granularity should be below 1,000 in order to limit the bias within 5%.

Flow level comparison is illustrated in Figure 6 in two aspects. One is the number of flows in the trace, the other is the size of the biggest flow. We use the five-tuple to identify a flow and modify the termination mechanisms in [2] to end a flow. As shown in the left plot, the flow number biases of both two samplings are similarly great. The great biases are caused by the dropping of small size flows, because they have fewer packets and are difficult to be sampled. However, the right plot shows that both two samplings have extremely small biases in estimating the biggest flow size, and t/T sampling provides better accuracy than 1/N sampling. Hence both samplings are suitable for accounting-based applications, which focus on big flows.



(a) Flow number bias (b) Biggest flow size bias

Fig. 6. Flow level comparison. t=1024

In conclusion, the results above indicate the similar performance of two samplings, 1/N packet-driven sampling and t/T time-driven sampling. Users can select either of them according to their own requirements and applications. However, if an application needs timing on sampling or continuous packets records, t/T time-driven sampling is a better choice.

7 Conclusions

Packet-driven sampling has been thought to be a convenient and accurate sampling mechanism for several years. However it is not the only choice. In this paper we show that time-driven sampling can achieve a similar accuracy with a proper implementation.

Firstly, we compare three sampling methods. For packet length distribution, 1/N and t/T samplings achieve similar accuracies, both much better than 1/T

sampling. For interval time distribution, t/T sampling behaves best since it can sample continuous packets.

After discussing optimization of proper sampling parameters in t/T sampling, we introduce t/T sampling and 1/N sampling into a measurement application and compare them by using derived packet-level and flow-level statistics and find out that these two methods could achieve almost the same performance.

References

1. Kimberly, C.Claffy.: Internet Traffic Characterization. Ph.D Dissertation, University of California, San Diego, (1994) 50–65
2. Estan, C., Keys, K., Moore, D., Varghese, G.: Building a better netflow. Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (2004) 245–256
3. Estan, C., Varghese, G.: New directions in traffic measurement and accounting. Proceedings of the 2001 ACM SIGCOMM Internet Measurement Workshop (2001) 75–80
4. Fraleigh, C., Diot, C., Lyles, B., Moon, S., et al.: Design and deployment of a passive monitoring infrastructure. Proceedings of the Thyrrhenian International Workshop on Digital Communications (2001) 556–575
5. Duffield, N.G., Lund, C., Thorup, M.: Charging from sampled network usage. ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA (2001)
6. He, G., Hou, J.C.: An In-Depth, Analytical Study of Sampling Techniques For Self-Similar Internet Traffic. To appear in the 25th International Conference on Distributed Computing Systems (ICDCS 2005), Columbus OH
7. Sampled NetFlow: http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801a7618.html
8. He, T., Liu, J., Kong, S.J., Shao, X.X., et al.: Generic Network Traffic Capture Platform building on Network Processor. poster in IEEE INFOCOM 2005, Miami, USA (2005)

Performance Evaluation of an Enhanced Distance-Based Registration Scheme Using the Normal Distribution Approximation

Jae Young Seo and Jang Hyun Baek*

Dept. of Industrial and Information Systems Eng., Chonbuk Natl. University, Korea
{jaeyoung, jbaek}@chonbuk.ac.kr

Abstract. Having an efficient location management scheme for mobile stations (MSs) is very important for optimizing the performance of mobile cellular systems. In this study, a distance–based registration (DBR) scheme and an enhanced DBR scheme are examined. According to computer simulations on the DBR scheme, it is shown that most MSs tend to distribute around the center of the location area. This centralizing property of the MS in the DBR scheme enables us to approximate the distribution of the MS in the location area to a normal distribution. We analyze the performance of the DBR scheme according to the above approximation and our mobility model. We also propose distance–based registration with an implicit registration (DIR) scheme in order to improve the performance of the DBR scheme. We evaluate the performance of the proposed scheme using the normal distribution approximation to compare the performance of the DBR scheme. The numerical results show that the DIR scheme not only outperforms the DBR scheme.

1 Introduction

In a mobile cellular system, location registration is the process by which the mobile station (MS) notifies the system of its location, status and other characteristics [1]. The MS informs the system of its location and status so that the system can efficiently page the MS when establishing an MS–terminated call. Without registration, the system does not know where MS is, and therefore the system should page the MS in every cell of the system. On the other hand, frequent registrations allow the system to know the location of the MS with great accuracy, and therefore the system pages the MS in only a few cells of the system. However, frequent registrations require a high load on the access channels and a moderate load on the paging channels. Thus, a trade off is required between the paging and registration load in order to optimize the use of the radio channels and system equipments.

In general, the DBR scheme is evaluated to be superior to a movement–based registration (MBR) scheme in all the cases [2, 3]. We present a new analysis for the DBR scheme and an enhanced DBR scheme. The DBR scheme causes an

* Corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 620–629, 2006.

MS to register its location when the distance between the current base station (BS) and the BS where MT last registered its location exceeds a predetermined threshold. The MS determines that it has moved a predetermined distance by computing the measured distance based on the difference in latitude and longitude between the current BS and the BS where the MS last registered. If this distance measure exceeds the threshold value, the mobile station registers [1].

Some previous works analyzed the zone–based registration (ZBR) scheme [4] and compares the performance of the ZBR scheme with that of the DBR scheme [5, 6]. The ZBR scheme, which has been adopted by most mobile cellular networks, shows good performance in general. One of the main problems of the ZBR scheme is ping–pong phenomenon which occurs excessive location registrations when an MS zigzags between two neighboring location areas. On the contrary, this ping–pong phenomenon does not occur in the DBR scheme and the registration load of the DBR scheme distributes equally to all cells in the location area. So, the DBR scheme may need a smaller registration load than the ZBR scheme, which was described using mathematical analysis by Baek et al. [5].

One of the main assumptions in [5] is that the MS uniformly distributes over the circular location area. However, such an assumption does not coincide with real situations and so it underestimates the performance of the DBR scheme. Therefore, if properly estimated, the DBR scheme may be superior to the ZBR scheme in most cases.

In this study, our computer simulations of the DBR scheme show that most MSs have a tendency to distribute around the center of the location area. Using this tendency of the MSs, we analyze the performance of the DBR scheme. We also propose a hybrid registration scheme called DIR (distance–based registration with an implicit registration) scheme which improves the performance of the DBR scheme. We evaluate the performance of the DIR scheme and show that the DIR scheme is superior to the DBR scheme in all the cases and better than the ZBR scheme in most cases.

When a call occurs in the DBR scheme, the system can implicitly know the MS's location without any additional registration procedure. In other words, when a call arrives at an MS or an MS originates a call, the MS implicitly notifies the system of its location information by using the call set up messages without an additional registration message. This process is referred to as implicit registration. Hence, if the DBR scheme incorporates implicit registration, then the network can know the MS's location whenever there is an call either to or from the MS without any explicit registration process. The network sets up a new location area in which the MS's location is the center of the new location area, and it reduces the number of registrations. However, the implicit registration does not have any influence on the number of registrations in the ZBR scheme.

Following this introduction, Section 2 describes the mobility model for the DBR scheme and the normal distribution approximation for the DBR scheme. Section 3 describes the proposed DIR scheme and the analytic model for the DIR scheme. The numerical results are shown in Section 4. Finally, concluding remarks are given in Section 5.

2 Distance–Based Registration and Its Performance

2.1 System Description

To analyze and compare the performance of the DBR scheme requires a reasonable yet simple mobility model. Several mobility models have been proposed; a simple one–dimensional model [2], a random walk model [3, 7, 8], a fluid flow model [9], a four–directional model [4, 5] and the Brownian motion model [10]. Among them, a simple one–dimensional model is too simple to be applied for exact analysis. The random walk model and the fluid flow model are also relatively simple and make it easy to analyze the performance of one registration scheme but difficult to analyze other schemes. For example, the random walk model is good for analyzing the performances of the MBR scheme, but it is not adequate to apply to the ZBR scheme [6]; and the fluid flow model is good for obtaining the performance of the ZBR scheme, but it cannot be applied to the DBR scheme. The four–directional mobility model can handle direction changes of the subscribers simply and realistically, which enables us to effectively analyze the performance of both of the DBR and ZBR schemes. Therefore, in this paper we extend the four–directional mobility model [4, 5] to analyze the DBR and DIR schemes.

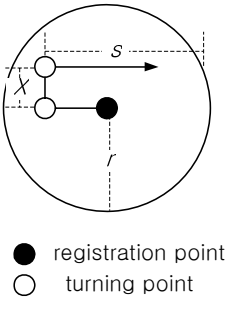


Fig. 1. Mobility of a mobile subscriber

The following are assumed in our model [4, 5].

- An MS moves in a straight line, until it reaches a turning point.
- When it reaches the turning point, it can choose any direction with equal probability.
- The distance between two consecutive turning points is exponentially distributed.

For the performance evaluation, the following parameters are defined.

X : the distance between two consecutive turning points
K : the number of registrations between two consecutive turning points
N : the number of registrations per subscriber for an hour
L : the distance moved for an hour
s : the distance from the last turning point to the border of the location area

2.2 Distance-Based Registration

The DBR scheme causes an MS to register its location when the distance between the current BS and the BS where it last registered exceeds a predetermined threshold. The MS determines that it has moved a certain distance by computing a measured distance based on the difference in latitude and longitude between the current BS and the BS where the MS last registered its location. If this measured distance exceeds the threshold value, the MS registers its location [1].

From the definition of the DBR scheme, the location area of the DBR scheme can be modelled as a circle whose radius is equal to the threshold as shown in Figure 1. Baek et al. [5] assumes that the MS distributes uniformly over the circular location area and calculates the number of registrations to evaluate the performance using those assumptions. If the MS distributes uniformly over the circular location area of the DBR scheme, the probability density function of the random variable, s, when the radius of location area is r, can be formulated as follows [5]:

$$f_S(s) = \frac{2}{\pi r^2}\sqrt{r^2 - (\frac{s}{2})^2}, \qquad 0 < s \leq 2r \tag{1}$$

2.3 Centralizing Tendency and Normal Distribution Approximation

In equation (1), it is assumed that MT is uniform distribution in LA. However, DBR has circular-shaped LA and after registration, MS starts to move from the center of LA. In this chapter, we show whether the distribution of MT in LA is changed in DBR with 1-dimensional random walk[2].

The method of random walk model is easy and simple to check the distribution of MT. We calculate the steady-state probability with hexagonal cell configuration, CMR (call to mobility ratio) = 0.5 and d = 6. The steady-state probability of ring–i is present as π_i and the each value is like $\pi_0 = 0.062$, $\pi_1 = 0.569$, $\pi_2 = 0.253$, $\pi_3 = 0.086$, $\pi_4 = 0.023$, $\pi_5 = 0.004$. Even we consider the number of cells in each ring, MT is distributed near the center, not uniformly distributed. So we call this centralizing tendency.

This centralizing property of the DBR scheme makes our new analysis for the DBR scheme quite different from the previous approach. First, the probability density function of the random variable, s should be derived again according to the centralizing property.

This centralizing property of the MS in the DBR scheme enables us to approximate s to a normal distribution. In order to statistically examine whether s follows a normal distribution, the goodness–of–fit test is performed for the data from the simulations. We use the Anderson–Darling (A–D) test [11] to examine the goodness–of-fit of our approximation.

In the Anderson–Darling test, it is known that the data can be said to follow a normal distribution if the p–value is larger than 0.05 [11].

Table 1 shows the p–value of the Anderson-Darling test for each simulation result. From the table, it is shown that random variable s follows a normal distribution as the expected distance between two turning points, E(X), increases.

However, it is also observed that, when E(X) is very small, it is somewhat difficult to say that random variable s follows a normal distribution. Based on numerous simulation results for various situations, it is concluded that random variable s approximately follows a normal distribution.

Table 1. E(X) vs. p–value of the A–D test (r=5)

E(X)	Result 1	Result 2	Result 3
1	0.586	0.020	0.109
2	0.253	0.260	0.413
3	0.090	0.265	0.076
4	0.172	0.095	0.083
5	0.282	0.066	0.087

Next, let us estimate mean and standard deviation of s, two parameters of a normal distribution. In order to find the relation between random variable s and E(X), we use the simulation results for various situations. Some of these simulation results are shown in Table 2. From the table, it is shown that the mean value of s tends to decrease linearly as E(X) increases. Considering this relation between random variable s and E(X), the generalized formula for the mean value of s can be derived as a function of E(X). Similarly, the standard deviation can be derived as a progression of differences with a geometric ratio of 0.11. Equation (2) shows the final formula for the mean and standard deviation of s as a function of E(X). This equation enables us to express random variable s as a function of E(X) and get the mean and standard deviation without performing complicated and time–consuming computer simulations.

Table 2. E(X) vs. the mean value (μ) and standard deviation (σ)(r=5)

E(X)	Test 1	Test 2	Test 3
1	μ = 4.862,σ=1.759	μ = 4.837, σ=1.759	μ = 4.728, σ=1.759
2	μ = 4.621,σ=1.809	μ = 4.870, σ=1.950	μ = 4.811, σ=1.835
3	μ = 4.474,σ=1.854	μ = 4.635, σ=1.953	μ = 4.729, σ=1.945
4	μ = 4.674,σ=1.986	μ = 4.558, σ=1.854	μ = 4.542, σ=2.145
5	μ = 4.521,σ=2.090	μ = 4.662, σ=1.978	μ = 4.435, σ=1.997

$$f_S(s) = N(4.81 - 0.06\theta, 1.87 + \sum_{i=1}^{\theta-1} 0.11^i) \quad \text{where, } \theta = E(X) \tag{2}$$

Using the probability density function of s, the expected number of registrations between two consecutive turning points, E(K) can be derived as follows [12]:

$$E(K) = \sum_{k=1}^{\infty} \int_{0}^{2r} \int_{s+(k-1)r}^{s+kr} \frac{k}{\theta} e^{-\frac{x}{\theta}} f_S(s) dx ds \tag{3}$$

Finally, the expected number of registrations per subscriber per hour, E(N) can be derived as follows:

$$\begin{aligned} E(N) &= \frac{E(L)}{E(X)} E(K) \\ &= \frac{E(L)}{\theta} \sum_{k=1}^{\infty} \int_{0}^{2r} \int_{s+(k-1)r}^{s+kr} \frac{k}{\theta} e^{-\frac{x}{\theta}} f_S(s) dx ds \end{aligned} \tag{4}$$

3 Distance-Based Registration with an Implicit Registration and Its Performance

According to CDMA technical requirements [1], the BS can infer the MS's location when an MS successfully sends an Origination Message or Page Response Message. This process is called implicit registration. In other words, when an outgoing call from an MS successfully completes or an incoming call to an MS successfully completes, the BS can infer the location of the MS from the information in Origination Message or Page Response Message, respectively, without an additional registration message. Hence, if the DBR scheme incorporates implicit registration in a mobile cellular network, then the network can determine the MS's location whenever there is an outgoing or incoming call to or from the MS without any additional registration process. The network sets up a new location area in which the MS's location is the center of the new location area, which reduces the number of registrations.

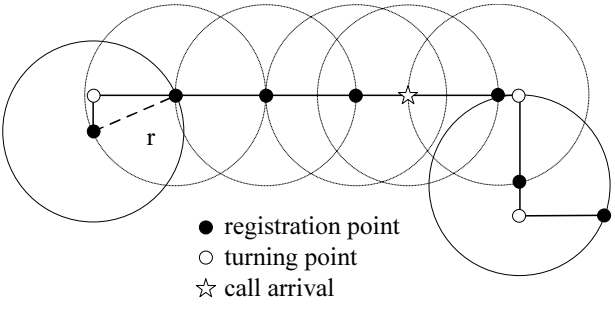


Fig. 2. Location registration procedure in the DIR scheme

As shown in Figure 2, the DBR scheme can be improved in a number of registrations by combining with an implicit registration, and the performance of this hybrid scheme becomes better as the number of calls generated by the MS increases. In this study, a hybrid scheme that combines a distance-based registration with an implicit registration (DIR) scheme is proposed. Furthermore, the performance of the DIR scheme is evaluated using the normal distribution approximation in order to compare it with that of the DBR scheme.

In order to evaluate the performance of the DIR scheme, it is assumed that the distance between two consecutive call generation points, denoted by a new random variable z, is exponentially distributed with mean λ. Concerning call generation, three possible intervals determined by x and z, should be considered in the DIR scheme. Then, the probability that the MS registers its location k times for an hour is expressed as follows [12]:

$$\begin{aligned} &P[\text{MS registers } k \text{ times}] \\ &= P[z+kr < x < z+(k+1)r \,|\, 0 < z < s] \\ &\quad + P[s+(k-1)r < x < s+kr \,|\, s+(k-1)r < z] \\ &\quad + \sum_{j=2}^{k} P[z+(j-1)r < x < z+jr \,|\, s+(k-j)r < z < s+(k-j+1)r] \end{aligned} \tag{5}$$

Using Equations (2) and (5), the expected number of registrations between two consecutive turning points, $\mathrm{E}(K)$, can be calculated as follows:

$$\begin{aligned} E(K) = \sum_{k=1}^{\infty} k[&\int_0^{2r}\int_0^{s}\int_{z+kr}^{z+(k+1)r} f(x,z,s)dxdzds \\ &+ \sum_{j=2}^{k}\int_0^{2r}\int_{(k-j)r+s}^{(k-j+1)r+s}\int_{z+(j-1)r}^{z+jr} f(x,z,s)dxdzds \\ &+ \int_0^{2r}\int_{s+(k-1)r}^{\infty}\int_{s+(k-1)r}^{s+kr} f(x,z,s)dxdzds] \\ &\text{where } f(x,z,s) = \tfrac{1}{\lambda}e^{-\frac{z}{\lambda}}\tfrac{1}{\theta}e^{-\frac{x}{\theta}}f_S(s) \end{aligned} \tag{6}$$

Finally, using Equation (4) and (5), the expected number of registrations per subscriber per hour in DIR can be calculated as follows:

$$\begin{aligned} E(N) = \tfrac{E(L)}{\theta}[\sum_{k=1}^{\infty} k\{&\int_0^{2r}\int_0^{s}\int_{z+kr}^{z+(k+1)r} f(x,z,s)dxdzds \\ &+ \sum_{j=2}^{k}\int_0^{2r}\int_{(k-j)r+s}^{(k-j+1)r+s}\int_{z+(j-1)r}^{z+jr} f(x,z,s)dxdzds \\ &+ \int_0^{2r}\int_{s+(k-1)r}^{\infty}\int_{s+(k-1)r}^{s+kr} f(x,z,s)dxdzds\}] \\ &\text{where } f(x,z,s) = \tfrac{1}{\lambda}e^{-\frac{z}{\lambda}}\tfrac{1}{\theta}e^{-\frac{x}{\theta}}f_S(s) \end{aligned} \tag{7}$$

4 Numerical Results

For the numerical analysis and performance comparison, the following are assumed.

- Radius of the location area: 5km
- Distance moved for an hour (L): 20km
- Mean distance between two consecutive call generation points ($\mathrm{E}(z)$): 4km
- Mean distance between two consecutive turning points ($\mathrm{E}(X)$): 3km.

The performances of the DBR and DIR schemes are evaluated using the normal distribution approximation and the performance of the DIR scheme is compared with that of the ZBR scheme.

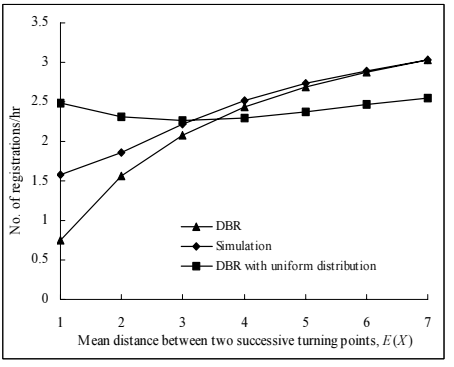


Fig. 3. E(X) vs. number of registrations in the DBR scheme and simulation result

Figure 3 shows that our approach is closer to the simulation results than the previous model [5]. From this figure, it is shown that the proposed model coincides very closely with the simulation results as the mean distance between two consecutive turning points, E(X), increases. However, there is a small gap between the simulation result and our calculation when E(X) is very small. In conclusion, it is shown that random variable s exactly follows a normal distribution for large E(X), and approximately follows a normal distribution for small E(X).

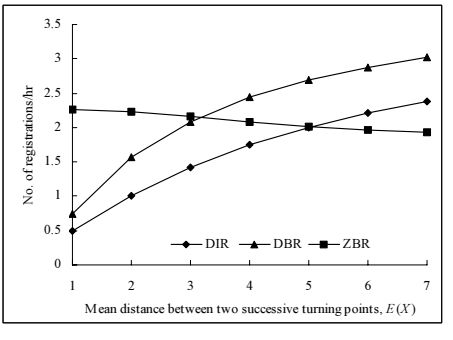


Fig. 4. E(X) vs. number of registrations in the DIR, DBR and ZBR schemes (E(z)=4)

In Figure 4, the DIR scheme is compared with the DBR and ZBR schemes. For the sake of fairness, it is assumed that the size of the quadrangular location area in the ZBR scheme is the same as that of the circular location area in the DBR scheme (i.e., d=8.86 since $\pi \times r^2 = 25\pi = d^2$ where d is the length of a side of the quadrangular location area in the ZBR scheme).

The simultaneous paging scheme is also assumed. When the simultaneous paging is assumed, then we do not need to consider the paging cost because

every scheme has the same paging cost if each scheme has the same location area size. Figure 4 shows that the DIR scheme outperforms the DBR scheme for every E(X). When the mean distance between consecutive turning points, E(X), is smaller than the radius of the location area (5km), the DIR scheme shows a better performance than the ZBR scheme. On the other hand, if E(X) is larger than the radius, the ZBR scheme shows a better performance than the DIR scheme. However, such a large value of E(X) rarely occurs in real situations. Therefore, we can conclude that the DIR scheme performs better than the DBR scheme in every case and better than the ZBR scheme in most cases.

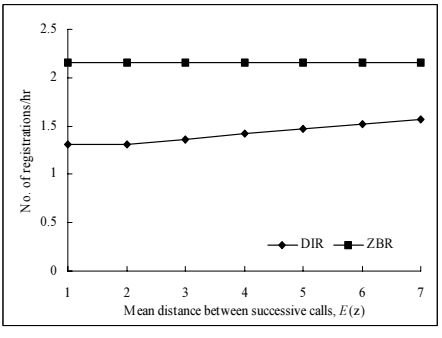


Fig. 5. E(z) vs. number of registrations in the DIR and ZBR schemes (E(X)=4)

Figure 5 shows the number of registrations as E(z) varies given that E(X) is 3km. Note that a small E(z) means a high call generation rate and the ZBR scheme is not affected by call generation at all. Figure 5 shows that the high call generation rate influences the DIR scheme to reduce location registration and outperform the ZBR scheme. In conclusion, we can say that the DIR scheme outperforms the ZBR scheme in most cases as E(z) increases.

5 Conclusions

In this study, the DBR scheme and an enhanced DBR scheme were researched. The DBR scheme causes a mobile station to register its location when the distance between the current BS and the BS at which it last registered its location exceeds a predetermined threshold. According to computer simulations on the DBR scheme, it was shown that most MSs tend to distribute around the center of the location area. This centralizing property of the MS in the DBR scheme allows us to approximate the distribution of the MS in the location area to a normal distribution. Using the above approximation and our mobility model, the performance of the DBR scheme was analyzed. This study also proposed distance-based registration with an implicit registration (DIR) scheme to improve the performance of the DBR scheme and evaluated the performance of the DIR scheme using the normal distribution approximation. The numerical results

showed that the DIR scheme outperforms not only the DBR scheme in every case, but also the zone-based registration scheme in most cases.

Acknowledgment

This work was supported by grant No.R01-2006-000-10668-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

[1] EIA/TIA/IS–95–B, MS–BS Compatibility Standard for Wideband Spread Spectrum Cellular System(1999)
[2] Bar-Noy A., Kessler I., Sidi M.: Mobile users: to update or not to update?, Wireless Networks, **1(2)** (1995) 175–185
[3] Ryu B.H., Ahn J.H., Baek J.H.: Comparative performance evaluation of movement–based registration and distance–based registration, IEICE Tr. on Communications **E86-B(3)** (2003) 1177–1180
[4] Baek J.H., Ryu B.H., Lim S.K., Kim K.S.: Mobility model and performance analysis for zone–based registration in CDMA mobile communication system, Telecommunication Systems **14(1)** 2000 13–29.
[5] Baek J.H., Lie C.H.: Performance analysis of location registration methods: zone–based registration and distance–based registration, Journal of the Korean Institute of Industrial Engineer **23(2)** (1997) 385–401.
[6] Kim K.H., Baek J.H., Faloutsos M.: Modeling and Performance Analysis of Zone–Based Registration for Mobile Communication Networks, IEICE Tr. on Communications, submitted
[7] Li J., Kameda H., Li K.: Optimal dynamic mobility management for PCS networks, IEEE/ACM Tr. on Networking **8(3)** (2000) 319–327
[8] Baek J.H., Ryu B.H.: An Improved Movement–Based Location Update and Selective Paging for PCS Networks, IEICE Tr. on Communications **E83-B(7)** (2000) 1509–1516
[9] Xie H., Tabbane S., Goodman D.J.: Dynamic location area management and performance analysis, International Conference on Vehicular Technology (1993)
[10] Rose C., Yates R.: Location uncertainty in mobile networks: a theoretical framework, IEEE Communications Magazine **35(2)** (1997) 94–101
[11] Carver R.H., Carver M.R.: Doing data analysis with Minitab 14, Duxbury Press (2003)
[12] Ross S.: Stochastic Processes, Wiley (1996)

Interoperability Experiences on Integrating Between Different Active Measurement Systems*

Jaeyoung Choi[1], Geraldine Texier[2], Yongho Seok[1], Taekyoung Kwon[1], Laurent Toutain[2], and Yanghee Choi[1]

[1] School of Computer Science and Engineering
Seoul National University, Seoul, Korea
{jychoi, yhseok, tkkwon, yhchoi}@mmlab.snu.ac.kr
[2] RSM Department
ENST Bretagne, Rennes, France
{geraldine.texier, laurent.toutain}@enst-bretagne.fr

Abstract. Traffic measurement is an important issue in IP networks for both internet service providers and users. Over the past decade, a number of active measurement tools have been implemented to measure and analyze IP networks. By making these tools interoperable, we can measure a wide network that encompasses different administrative domains.

In this paper, we present the accumulated observation from our project, which integrates two different traffic measurement systems : Active Measurement Tool (AMT) and Saturne. The integrated measurement infrastructure, STAR, can measures one-way delay, one-way delay jitter, and packet loss rate metrics, as defined at the IETF. By using STAR, we have measured the network between Korea and France. As a result, we found that the network between Korea and France connected by TEIN link is stable and has low loss rate.

Keywords: Active Measurement, Traffic Measurement, System Integration, Interoperability.

1 Introduction

As the Internet is getting more and more complex and larger, measurement infrastructures and methodologies become essential to characterize network traffic. The Internet Protocol Performance Metrics (IPPM) working group of the Internet Engineering Task Force (IETF) has developed several metrics for this purpose, such as one-way delay [1], one-way packet loss [2], instantaneous packet delay variation [3]. With measurement data obtained, we can effectively perform the network management and can understand the characteristics of network well. For these purposes, there have been proposed many measurement tools like RIPE [4], Surveyor [5], Active Measurement Tool (AMT) [6], Saturne [7], PingER [8], AMP [9].

* This work was supported by STAR project, 2004.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 630–638, 2006.

We can classify these tools into two categories. In Active measurement category, a measurement machine will explicitly inject measurement packets called probe packets in a path. Also, the sending machine and the receiving machine have to agree on the format of probe packets. On the other hand, the passive methodology doesn't use an explicit measurement packet. Passive measurement is a means of tracking the performance and behaviour of packet streams by monitoring the traffic without creating or modifying it.

Most of active measurement infrastructures have developed its own measurement daemons. However, the characteristics of a specific network along the measured path is often limited by the scope of the path [10]. To measure a long network, there can be a lot of problems like economic cost or human resource unavailability. Alternatively, it can be a good solution to make multiple active measurement infrastructures interoperable.

When different active measurement infrastructures have been integrated, what to measure and how to measure have to be resolved before integration. What to measure is mainly related to metrics that will be provided by integrated measurement infrastructure. How to measure represents the functionalities of a measurement system. Even though several measurement systems provide the same metric, the detailed implementation of measurement can be quite different. There can be many issues int integrating different measurement infrastructures in the above two aspects. Examples of the most important issues are a method of time synchronization, a format of a probe packet, how to timestamp, a way to gather results, and generation of probe packets. We will describe each of these issues in Section 2.

The basic objective of our paper is to describe our experiences on integrating two different active measurement infrastructures, AMT of Korea and Saturne of France. To understand the network link between Korea and France, we make our infrastructures interoperable and perform measurements with Trans-Eurasia Information Network (TEIN) link. This paper is organized as follow; Section 2 presents general issues in integrating and compares the conventional active measurement tools. In Section 3, we describe our active measurement infrastructure, AMT and Saturne, respectively. Then, we explain detailed issues on interoperability and how to solve those issues in Section 4. We also show the result of one-way delay measurement in real network (TEIN) in Section 5. Finally, we conclude this paper.

2 Issues in Integration

In this section, we will talk about design issues in integrating two different measurement systems. As mentioned before, we consider five consideration points in integration.

1. Time synchronization
   - In the view point of interoperability, it will be recommendable to synchronize different measurement machines with same method. It is because the accuracy of metric measured seperately along two asymmetric paths can be fully guaranteed only when the resolution of each emission point is same.

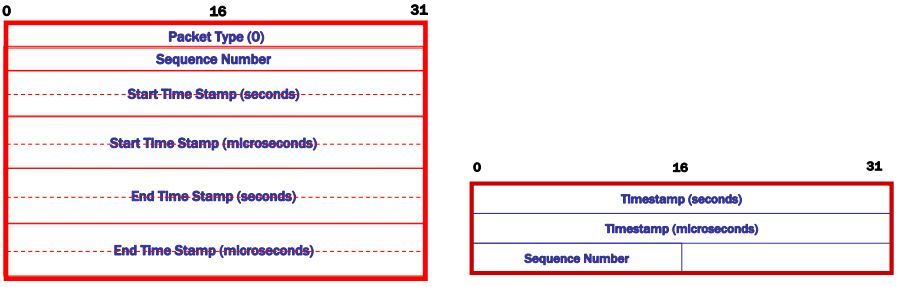
Table 1. Comparison of Characteristics of Active Measurement Tools

	AMT	Saturne	RIPE	PingER	AMP
Time synchronization	GPS	GPS + NTP	GPS	NTP	NTP
Time-stamping location	data link	data link	data link	IP layer	IP layer
Measured delay	one-way	one-way	one-way	two-way	two-way
Result processing	socket	RPC	rcp	local	N/A
Flow generation	poisson	linear	poisson	bursty	linear random

2. The format of probe packet
   - Before providing interoperability between two different systems, it has to be decided what to measure with the integrated system. There can be some systems that cannot be integrated together. For example, one system measures some metrics in an end-to-end manner while another system does hop-by-hop. Between these systems, it is impossible to provide interoperability.
   - After the agreement on measuring metrics, the format of a probe packet has to be decided. We recommend to keep the probe packet format same among interoperating systems to easily make the infrastructures integrated.
3. Timestamping
   - This issue is related to the policy where the stamping is done. When the timestamp field is filled at an application layer, layering delay can occur. From the one-way delay metric definition of IPPM, the layering delay has to be removed in active measurements.
   - In integrating different systems, time-stamping has to be done at least the same layer to minimize the error bound of accuracy.
4. Result processing
   - Each interoperable system must provide an interface to gather the measured data. When a machine calculates result metrics from probe packets emitted by peer, it transmits them to its peer or a central database for storage. Therefore each measurement machine must provide an interface for this functionality.
5. Flow generation
   - Flow generation means how often an measurement system generates probe packets. RIPE [4] schedules in proportion to poisson distribution which total average arrival rate is 1, PingER [8] uses the bursty form as flow distribution and AMP [9] generates according to linear random function for first 15 seconds per minutes. There is a trade-off between the accuracy of measurement and network overhead. And the flow of probe packets itself can even affect the network characteristics in some extreme cases.

Table 1 summarizes comparison with the characteristics of some active measurement tools in the point of issues described above.

3 Two Active Measurement Systems



(a) AMT (b) Saturne

Fig. 1. Probe Packet Format

3.1 Active Measurement Tool (AMT)

Active Measurement Tool (AMT) is an active measurement infrastructure made by Seoul National University (SNU) in 2000. The AMT measurement architecture can measure one-way delay, one-way packet loss and one-way delay jitter. Measurement machines of AMT is synchronized by using "GPS Clock 200" [11]. And time-stamping is performed at link layer by modified Free BSD kernel.

3.2 Saturne

Saturne is an active measurement infrastructure made by "Groupe des ecoles des telecommunications / Ecole Nationale Superieure des Telecommunications de Bretagne" (GET/ENST) in 2003. The Saturne architecture performs end-to-end active measurements of one-way delay and packet loss rate. The Saturne architecture uses Trimble smart antenna [12] as a source of network time protocol (NTP) [16]. Trimble's GPS system generates a pulse-per-second PPS synchronized to UTC within +/- 100 ns. Time-stamping is achieved by ADServ/ALTQ [13] [14] tools.

4 Interoperability Between AMT and Saturne

As described in the previous section, we integrated with AMT and Saturne. Figure 2 shows the architecture of our integrated infrastructure, STAR, and message flow for traffic measurement. Each measurement machine located at Korea and France is configured with IP address of peer measurement machine, start time and duration of measurement and traffic generation parameters (total average arrival rate of poisson process) before measurement. When the measurement starts, measurement machines periodically send probe packets. Once receiving a probe packet, STAR daemon calculates metrics and stores them to both database server maintained by Korea and France using Remote Procedure Call (RPC). The graph of the measured traffic is serviced by database server and can be displayed in real time (http://star.apan.snu.ac.kr). In this section, we will list some issues in interoperability and describe how to solve each issue.

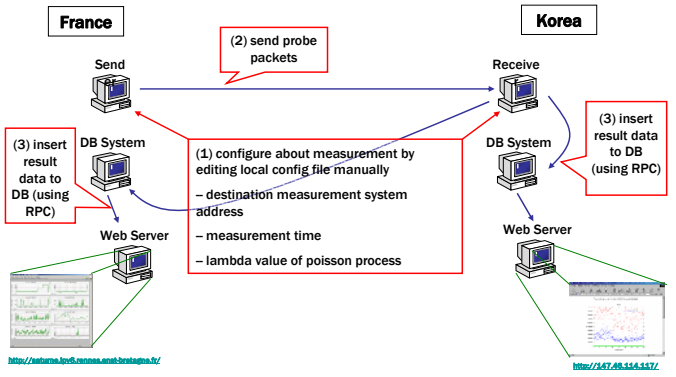


Fig. 2. Integrated Infrastructure

4.1 Time Synchronization

Synchronization is one of the most important element in active measurement architecture in order to measure IP networks accurately. Both AMT and Saturne uses GPS as a source of NTP even though they use different GPS product. It can be possible to use CDMA technology as the GPS is so expensive and we have to maintain several measurement machines. The way to use CDMA is more cheap, but its accuracy about time resolution does not match with GPS.

4.2 Probe Packet Format

The most important fields in the probe packet format are timestamp and sequence number fields. Both AMT and Saturne have these two fields though the sequence of these fields is different. But, the problem in designing new probe packet format was a field for AMT's control plane. We will provide a control plane in future work.

Figure 1 shows the probe packet format of AMT and Saturne before integration. While AMT needs extra payload field to enable the application to access the receiving timestamp field, Saturne uses BSD Packet Filter (BPF) [15] and therefore Saturne doesn't need extra receiving timestamp field. Besides AMT has a packet type field in order to distinguish a probe packet from control packets. Figure 3 shows the probe packet format of STAR. In order to integrate without much modification of each system, we decide to keep its own timestamping mechanism of AMT and Saturne. A receiving measurement machine of STAR in Korea reads a timestamp, TTL, and TOS field at application layer while a receiving machine in France does the same functions at link layer. There are two kinds of timestamp field, start and end. A start timestamp field is filled at the link layer of sending machine, and an end timestamp at the same layer of receiving machine. In order to modify the value of a start timestamp at link layer, we disable a checksum function of UDP. The TTL and TOS fields in payload is equal to those of IP header. These fields are added because of the limitation

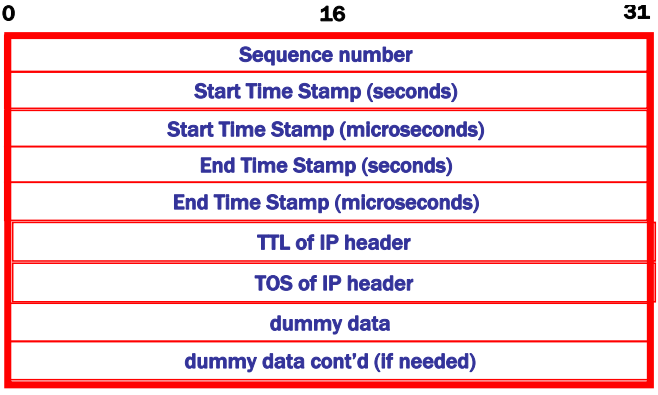


Fig. 3. Probe Packet Format of STAR

about AMT's implementation method. Lastly the dummy field ensures the probe packet to be of the same size even after inserting some extra fields later. It is important for the exact mathematical analysis to keep the size of probe packets same between several experiments.

4.3 Time-Stamping Location

AMT uses the socket library to send and receive probe packets and there are no access mechanism to write/read timestamp fields in ethernet layer. In AMT, modified ethernet layer performs to fill the value of receive time the stamp field in application payload with system time synchronized by GPS. To do this, FreeBSD kernel of AMT measurement machines have been modified. This is why the receiving field is needed in AMT unlike Saturne. Unlike AMT, Saturne uses the ADServ/ALTQ mechanism for time-stamping. By using ADServ/ALTQ mechanism, Saturne can read or write directly probe packet header in link layer.

In integrating two different measurement tools, it is important to record the emission and reception timestamp at the same layer to remove layering delay. In order to approximate our result one-way delay value to its definition of IPPM (it is defined at RFC 2679 [17]), we agreed to timestamp at link layer. And because we don't want to modify too much parts of each system for interoperability, we decided to have time-stamp fields in probe packet format and AMT accesses them in application layer, while Saturne does the same functions in link layer.

4.4 Gathering Results

When the user gives a gathering command to the control server in AMT, the control server signals to specific measurement machines to inform data gathering. Right after receiving that signal, "gathering thread" of each measurement machine will send its measured data to "db daemon" of the control server by means of TCP socket communications. In Saturne, data gathering process is invoked

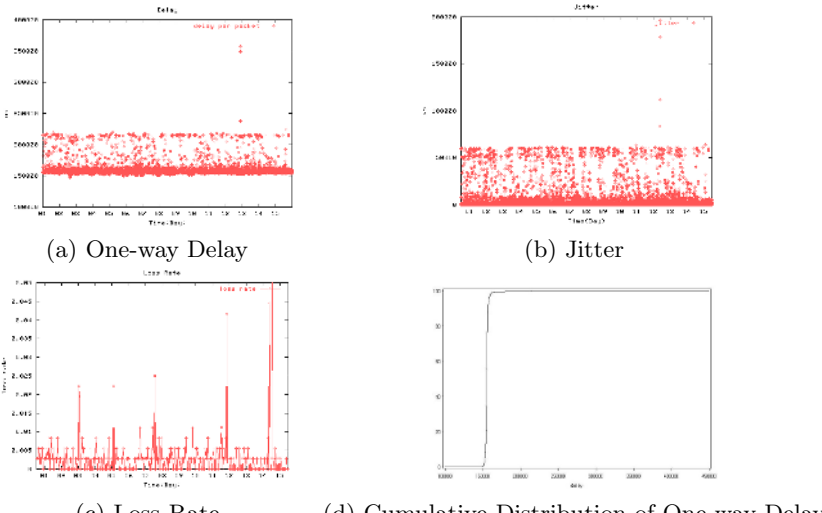
whenever a probe packet arrives. After receiving and calculating with arriving probe packets, the result is stored at the central database by using Remote Procedure Call (RPC) communications. The central database server provides an RPC interface to store the result data to its database, and administers an RPC server daemon.

Regarding integrated infrastructure, it is simpler and easier to implement to use RPC for gathering results which is adopted in STAR.

5 Experimental Results

The STAR architecture has been operating between Korea and France through TEIN link. TEIN link connects KOrea REsearch Network (KOREN) in Korea and Renater in France together. We install two measurement machines at KOREN and Renater, respectively. In order to validate the implementation of STAR architecture, we have performed several sets of experiments. In this section, we show the measurement results of an experiment that is carried out from January 1st 2005 to January 15th 2005. The graphs of measurement results are available at http://star.apan.snu.ac.kr.

Figure 4 shows graphs of these three metrics. One-way delay is calculated from values of time-stamp field of received probe packets. As described before, there are two time-stamp fields such as sendtime and recvtime in the probe packet format, and obviously one-way delay is recvtime minus sendtime. These values are calculated by a micro-second unit for a more accurate measurement. As showing at figure 4 (a), one-way delay values of this network look stable,



(a) One-way Delay (b) Jitter

(c) Loss Rate (d) Cumulative Distribution of One-way Delay

Fig. 4. Measurement results

and they mainly range from about 150 ms to about 220 ms. Therefore the jitter value of one-way delay is quite small as shown at figure 4 (b). Lastly, loss rate of this network is low, too. At figure 4 (c), most of loss rate values are distributed below 0.5%. We hastily thought that this network path, that is KOREN-TEIN-Renater, has a stable characteristics than other path between Korea and France. In the following section, we are going to show some statistical data to confirm this judgment.

Table 2. Statistics of one-way delay

	value (unit μs)
Total Number	129120
Maximum	408596
Minimum	147488
Average	155632.75
Median	155450
Lower Quartile	154743.0
Upper Quartile	156046.0
Standard Deviation	3657.80

Table 2 shows statistical data about figure 4 (a). Maximum one-way delay is about 400 ms and minimum is about 140 ms. Average value is about 155 ms and median is also about 155 ms. And from two quartile values and standard deviation of one-way delay values, we can conclude there is not much variation of one-way delay values in this network path because standard deviation is only 3 ms, and it is quite small value. Figure 4 (d) supports this conclusion, showing us that most of one-way delay values are around 150 ms. But, we must not jump to the conclusion that this network path is better than others, since we don't know whether the average value of one-way delay, 150 ms, is smaller than others or not. The comparison with other network paths will be depicted in the analysis paper. From table 2, we suggest that this network path is partially qualified to serve a multimedia application since variation of one-way delay is small and loss rate of this path is quite low.

6 Conclusion

In order to measure the network, there has been implemented several measurement tools. By making different active measurement infrastructures interoperable, we can get significant benefits. To illustrate the feasibility, SNU in Korea and GET/ENST in France integrate their own active measurement tools, AMT and Saturne. In integrating different measurement systems, there are many design issues such as how to synchronize measurement machines, how to time-stamp a probe packet, how to gather results, what to measure. We have integrated the format of probe packet, time-stamping location, result metrics, database schema, and how to gather results. We successfully implement the integrated

active measurement infrastructure, STAR. To validate this system and to analyze the characteristics of network path, KOREN-TEIN-Renater, we performed a measurement for about two weeks and found that a variation of one-way delay in this network path is small and loss rate is low too.

References

1. G. Almes et al, "A One-way Delay Metric for IPPM," RFC 2679, September. 1999.
2. G. Almes et al, "A One-way Packet Loss Metric for IPPM," RFC 2680, September. 1999.
3. C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IPPM," draft-ietf-ippm-ipdv-08.txt, November 2001.
4. F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhelm. "Providing Active Measurements as a Regular Service for ISP's," In Passive and Active Measurements Workshop, April. 2001.
5. http://www.advanced.org/surveyor.
6. Jaehoon Jeong, Seungyun Lee, Yongjin Kim, and Yanghee Choi. "Design and Implementation of One-way IP Performance Measurement Tool," ICOIN, 2002.
7. J. Corral, G. Texier, and L. Toutain, "End-to-End Active Measurement Architecture in IP Networks (SATURNE)," Proceedings of Passive and Active Measurement Workshop PAM'03, La Jolla, CA, 2003.
8. W. Matthews and L. Cottrel, "The PingER project: Active Internet performance monitoring for the HENP community," IEEE Communications, vol. 38, no. 5, pp. 130–136, May 2000.
9. http://amp.nlanr.net/AMP.
10. Maheen Hasib and John A. Schormans, "Limitations of Passive & Active Measurement Methods In Packet Networks," London Communications Symposium 2004, September. 2004.
11. http://www.gpsclock.com.
12. http://www.trimble.com.
13. http://www.rennes.enst-bretagne.fr/~medina/ds-imp/.
14. K. Cho, "The design and implementation of the altq traffic management system," Ph.D. dissertation, Keio University, 2001.
15. S. McCane and V. Jacobson, "The bsd packet filter: A new architecture for user-level packet capture," 1993.
16. http://www.eecis.udel.edu/~ntp.
17. http://www.faqs.org/rfcs/rfc2679.html.

Network and Transport Protocols

Receiver-Based Rate Control with One-Way Trip Time for Multimedia Applications

Myungsik Yoo, Min-Cheol Hong, and Younghan Kim

School of Electronic Engineering, Soongsil University, Seoul, Korea,
{myoo, mhong, younghak}@ssu.ac.kr

Abstract. In the congestion control algorithm, the round trip time (RTT) has been used as a key parameter to estimate the degree of network congestion. However, RTT indicates the packet delays taken in both the forward path (from source to destination) and the backward path (from destination to source). Therefore, the congestion control algorithm based on RTT may falsely trigger the rate control due to the congestion built in the backward path. In this paper, we propose a receiver-based congestion control algorithm in which the network condition is estimated by using the one-way trip time (OTT). By virtue of the use of OTT, the proposed algorithm can effectively decouple the forward path from the backward path when estimating the network congestion. The simulation results confirm the effectiveness of the proposed algorithm.

1 Introduction

As the demand on the multimedia applications increases, it becomes important for Internet to deliver the real-time packets within the acceptable delay and jitter bounds. There are two approaches in enhancing the quality of multimedia services. One is to enhance the quality of multimedia service in the network level. In this approach, the routers actively participate in differentiating quality of service (QoS) based on the contents that packets carry. The QoS can be provisioned on the connection basis (e.g., IntServ)[1] or on the packet basis (e.g., DiffServ)[2]. However, none of these are widely deployed yet and, Internet supports only *best effort service* where all IP datagrams are equally treated by IP routers regardless of their contents (e.g., real-time video streams).

The other is to enhance the quality of multimedia service at the network edge (i.e., at the application level). In this approach, the hosts engaging in multimedia session takes the control on the multimedia quality. The multimedia QoS control at the network edge involves many technical areas, which include streaming server, video compression, media synchronization, error control and congestion control (See more surveys in [3]). All of them are important and cross-related for multimedia QoS enhancement. Among them, the congestion control plays a crucial role since the multimedia packets may suffer from the unpredictable delays and losses while traversing Internet where the congestion takes place randomly.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 641–649, 2006.

The classification of the congestion control algorithms may be done in many different ways [3,4]. They can be classified as either the window-based or the rate-based [5,6,7,8]. The former controls the number of pending packets in the network (e.g., the packets that has not been acknowledged yet) by adjusting congestion window based on network condition, while the latter controls the packet transmission rate based on the congestion feedback information. They can also be classified either as the source-based [5,7] and the receiver-based [6,8]. In the former the source monitors the network condition and regulates the transmission rate, while in the latter the receiver does. They can also be classified as either the unicast-based [5,6,7] or the multicast-based [8] or as either the single-rate-based [5,6,7,8] or the multi-rate-based.

In general, the congestion control algorithm is desirable to have following properties. Firstly, it should be TCP-friendly. Secondly, it should not give the servers (e.g. streaming servers) too much of burden since the server is busy with serving many clients simultaneously (i.e. one to many relationship between server and clients) and in addition, most of packet transmissions takes place at the server side rather than at the client side. Thirdly, the congestion control algorithm should not cause too much of overhead. In other words, it is good to keep the feedback information as small as possible. Fourthly, the network congestion should be able to measure the network condition as accurately as possible. Otherwise, the congestion control algorithm may falsely alarm the congestion. In such a case, the transmission rate is unnecessarily decreased, and consequently QoS of multimedia service is degraded.

In this paper, we propose a new congestion control algorithm called the receiver-based rate control with one way trip time (RRC-OTT). The proposed RRC-OTT algorithm is designed to meet the properties that mentioned above. Especially, unlike the congestion control algorithm using the round trip time (RTT), RRC-OTT algorithm estimates the network condition using the one-way trip time (OTT). By the virtue of the use of OTT, RRC-OTT algorithm can effectively decouples the forward path from the backward path when estimating the network congestion level.

The rest of paper is organized as follows. We describes the proposed congestion algorithm in Section 2. Section 3 presents some simulation results and shows the effectiveness of our proposed algorithm. Finally, we conclude this paper in Section 4.

2 RRC-OTT: Receiver-Based Rate Control with One-Way Trip Time

When the sender (multimedia server) sends the data (e.g., multimedia contents) to the receiver (client), we call the path from the sender to the receiver the forward datagram path (or simply forward path) and the path from the receiver to the sender the backward datagram path (or simply backward path). The congestion control algorithms aim at adapting the transmission rate to the network capacity available on the forward path. Typically, the RTT is used to

measure the network condition (i.e., network capacity available). However, the use of RTT implicitly couples two separate paths (i.e., the forward and backward paths) tightly. Thus, it may happen that the congestion on the backward path triggers the congestion control. In other words, the congestion control algorithm using the RTT may falsely alarm the congestion, which does not exist in the forward path.

On the other hand, the RRC-OTT uses the OTT on the forward path to determine the network condition. Thus, it is possible for the RRC-OTT to decouple the forward path from the backward path when estimating the network congestion, and consequently is able to accurately estimate the congestion condition on the forward path.

2.1 Algorithm at the Receiver

In RRC-OTT, the sender timestamps the packet (say, i-th packet) to record the timing information. Upon arrival of a packet, the receiver obtains the OTT for the i-th packet (OTT_i) by using the timestamp value. Based on the OTT_i, the receiver obtains the smoothed OTT (SOTT) by (where $m = 0.125$ as recommended in [9]),

$$SOTT_i = (1 - m) \times SOTT_{i-1} + m \times OTT_i \tag{1}$$

Note that RRC-OTT algorithm needs the clock synchronization between the sender and the receiver to get the accurate timing information such as OTT, SOTT. However, RRC-OTT algorithm works well even with the timing skew. This is because RRC-OTT algorithm makes the decision on the network congestion based on the relative value of timing information (e.g., comparison between OTT_j and $EOTT_i$) as to be explained later in this section.

The major role of the receiver is to make a decision whether the network is congested or not. The decision mechanism of RRC-OTT is illustrated in Fig 1. In RRC-OTT, the network condition on the forward path at a specific time (say, when the i-th packet arrives at the receiver) is indicated by $SOTT_i$. Based on the $SOTT_i$ the receiver sets the expected OTT ($EOTT_i$), which functions similarly as the retransmission timeout value in TCP. Therefore, under the network condition of $SOTT_i$, a packet normally arrives at the receiver within $EOTT_i$. Thus, the $EOTT_i$ is used as a criterion for deciding network congestion in the RRC-OTT algorithm, and can be obtained by (where $n = 0.25$ and $p = 4$ as recommended in [9]),

$$EOTT_i = SOTT_i + p \times SDEV_i \tag{2}$$

$$SDEV_i = (1 - n) \times SDEV_{i-1} + n \times |SERR_i| \tag{3}$$

$$SERR_i = OTT_i - SOTT_i \tag{4}$$

In order for the receiver to make the decision on network congestion, the $EOTT_i$ needs to be compared to the delay of the packet that arrives later on. Since it is difficult to catchthe changes of network condition if two packets are

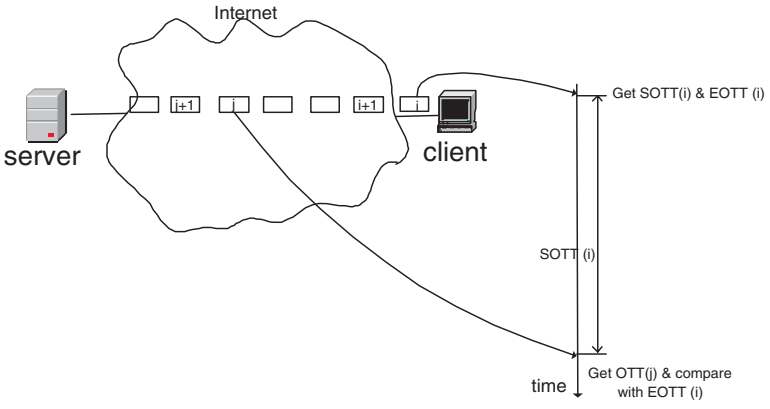


Fig. 1. Congestion control mechanism of RRC-OTT

too close or too far in time, it is set for the receiver to wait $SOTT_i$. When the packet (say, j-th packet) arrives after $SOTT_i$, the receiver obtains OTT_j, which indicates the changes of network condition. If the j-th packet arrives at the receiver within $EOTT_i$, the forward path is regarded as a good state. Otherwise, it is regarded as a congested state. The algorithm at the receiver can be described in detail as follows.

- Upon arrival of i-th packet, obtain OTT_i, $SOTT_i$, and $EOTT_i$.
- Start the timer for the interval of $SOTT_i$.
- After the timeout, waiting for the packet that arrives first (e.g., j-th packet).
- Upon arrival of j-th packet, obtain OTT_j, and decide the network condition by comparing OTT_j to $EOTT_i$.
  - If $OTT_j \leq EOTT_i$, then send the increase acknowledgment (INC ACK).
  - If $OTT_j > EOTT_i$, then send the decrease acknowledgment (DEC ACK).
- Whenever detecting the packet loss, send DEC ACK immediately.

2.2 Algorithm at the Sender

The major role of the sender is to regulate the sending rate based on the feedback information from the receiver.

- Whenever receiving the DEC ACK, decrease the current rate by half ($IPG_{i+1} = IPG_i \times 2$).
- Whenever receiving the INC ACK, increase the current rate as

$$IPG_{i+1} = \frac{IPG_i \times (\alpha \times SOTT_i)}{IPG_i + (\alpha \times SOTT_i)}. \tag{5}$$

- Running the safety timer, which is set to $K \times SOTT$, for the case when there is no feedback information from the receiver due to the severe congestion.
- Whenever the safety timer expires, decrease the current rate by half.

2.3 Enhancement

The RRC-OTT can be refined further in two aspects. Firstly, the RRC-OTT can limit the amount of feedback information (i.e., INC ACK) generated by the receiver. Since the feedback information may aggravate the congestion, it is desirable to keep the number of ACKs as small as possible unless the performance is not degraded. The RRC-OTT can limit the amount of INC ACK by the parameter β. Specifically, the receiver controls the number of INC ACKs by sending the INC ACK out at every $\beta \times SOTT$ instead of at every INC ACK event.

- At the receiver
- At every event of INC ACK
  - if $\beta \times SOTT$ has elapsed since last INC ACK, send INC ACK immediately.
  - it not, ignore the INC ACK event

Secondly, the accuracy of RRC-OTT can be enhanced by adding the function to handle the feedback information being delayed. When the backward path suffers from the severe congestion, the feedback information such as INC ACK and DEC ACK may be delayed. Since the feedback information being excessively delayed may change the transmission rate in an opposite way (e.g., decrease when need to increase or increase when need to decrease), the sender checks the validity of feedback information. For this purpose, the sender maintains SOTT on the backward path ($SOTT_{backward}$) with Eq. (1). Whenever the feedback information arrives at the sender, its validity is checked by comparing the delay of feedback packet ($OTT_{backward}$) to the threshold ($\eta \times SOTT_{backward}$). If the feedback information has been delayed longer than the threshold, then it is discarded. Otherwise, the feedback information is applied to the rate control.

- At the sender
- Upon arrival of feedback information, obtain $OTT_{backward}$ and $SOTT_{backward}$
  - if $OTT_{backward} \leq \eta \times SOTT_{backward}$, use the feedback information for the rate control
  - if $OTT_{backward} > \eta \times SOTT_{backward}$, discard the feedback information

3 Simulation Results and Discussion

The performance of RRC-OTT is evaluated through the simulations. The topology used for the simulation is shown in Fig. 2. It is assumed that the data and ACK packets are 100 byte and 40 byte long, respectively. The buffer size at the switch 1 is assumed to be 8 Kbyte. The bottleneck link capacity is set to 5 Kbyte/sec.

The queueing delay of RRC-OTT at the bottleneck link is shown in Fig. 3 for a single flow.We set α to 2 and β to 2 so that $2 \times SOTT$ in RRC-OTT is

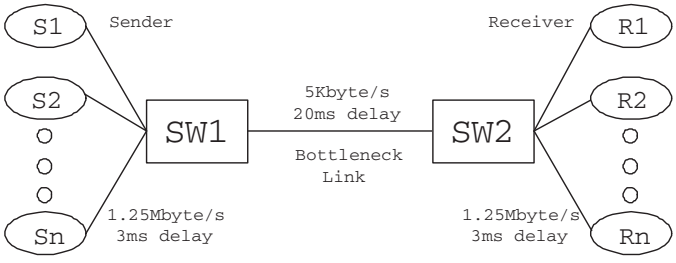


Fig. 2. Simulation topology

approximately equal to $SRTT$ in RAP [5]. The simulation results show that RRC-OTT keeps the queuing delay much lower than RAP. In addition, RRC-OTT achieves higher bottleneck link utilization (97 %) than RAP (95 %). Even though it is not shown, the RRC-OTT results in better performance in both queuing delay and link utilization for the multiple flows as well.

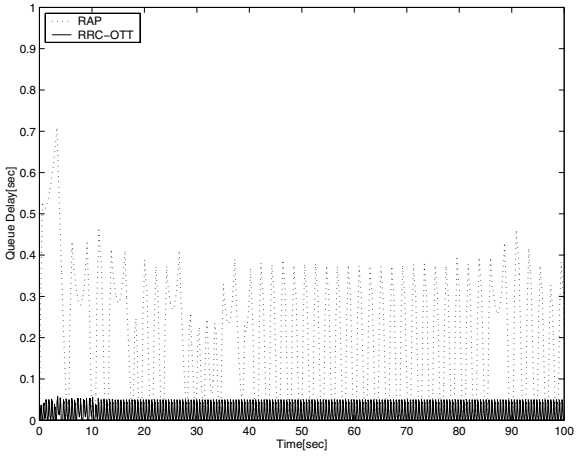


Fig. 3. Queuing delay for a single flow

The performance of RRC-OTT depends on the parameters such as α and β. The parameter α in the Eq. (5) is related to the sending rate. As α becomes larger, the increment of the sending rate decreases, and vice versa. As shown in Table 1, the bottleneck link utilization decreases with α. Similarly, the parameter β in the Section 2.3 is related to the rate of sending INC ACK. As β becomes larger, the INC ACK is sent out less, and vice versa. Thus, the bottleneck link utilization decreases with β.

To prove the TCP-friendliness of RRC-OTT, several TCP and RRC-OTT flows are fed into a bottleneck link. As the number of flows increases, the

Table 1. Bottleneck link utilization

fixed parameter	variable parameter	utilization
$\beta = 2$	$\alpha = 2$	97%
	$\alpha = 4$	96.66 %
	$\alpha = 8$	83.38 %
	$\alpha = 16$	77.64 %
$\alpha = 2$	$\beta = 2$	97 %
	$\beta = 4$	88.5 %
	$\beta = 8$	88.42 %
	$\beta = 16$	77.44 %

bottleneck link capacity is proportionally increased. In Fig. 4, the solid lines show the average link capacity consumed by TCP flows and RRC-OTT flows, while the dotted lines show the minimum and maximum link capacity consumed. As shown in Fig. 4, RRC-OTT fairly shares the link capacity internally with other RRC-OTT flows and externally with other TCP flows, each of which consumes approximately 5 Kbyte/sec.

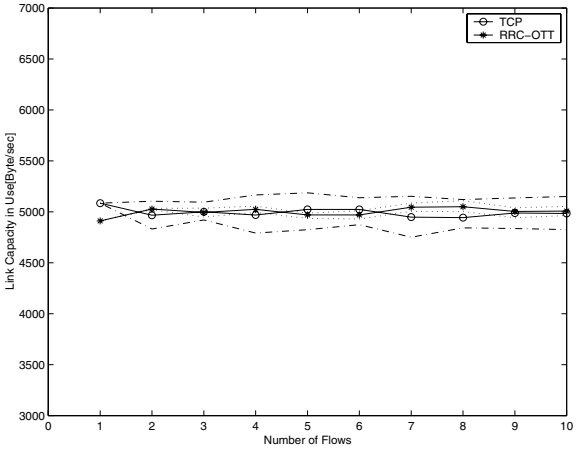


Fig. 4. Fairness of RRC-OTT

The use of OTT enables RRC-OTT to accurately measure the network condition on the forward path. To show the effectiveness of RRC-OTT, we feed the interference traffic into the backward path. Thus, the feedback information may be delayed or dropped randomly due to the congestion caused by the interference traffic in the backward path. As shown in Fig. 5, RAP shows a poor performance as the interference traffic increases, where the link utilization decreases

as low as around 30 %. This can be explained from the fact that the congestion built on the backward path falsely triggers the rate control. Consequently, as the congestion on the backward path gets worse, the link on the forward path gets under-utilized further. On the other hand, RRC-OTT keeps the forward link utilization high even when the backward path gets highly congested.

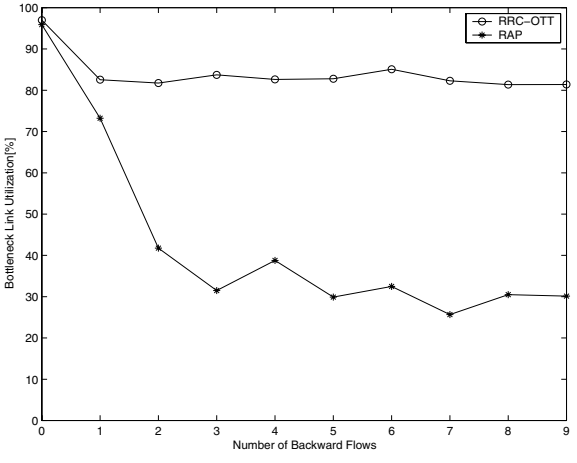


Fig. 5. The effect of congestion on the backward path

4 Conclusion

Conventionally, the congestion control algorithms use the RTT to estimate the network congestion. Since the packet delay is accumulated in both forward and backward paths during the RTT, the congestion control algorithm using the RTT may falsely triggers the rate control on the forward path due to the congestion in the backward path. In order to solve such problem, we propose a new congestion control algorithm, called RRC-OTT. RRC-OTT algorithm measures the degree of network congestion using the OTT, which enables to decouple the forward path from the backward path. Without any clock synchronization requirement between source and receiver, RRC-OTT algorithm accurately measures the network condition even with a heavy congestion in the backward path. With the computer simulations, RRC-OTT is proven to be an efficient and TCP-friendly congestion control algorithm for the multimedia applications.

Acknowledgement

This work was supported by the Korea Research Foundation Grant (KRF-2004-005-D00147).

References

1. M. Welzl and M. Muhlhauser, "Scalability and Quality of Service: A Trade-off ?," *IEEE Communications Magazine*, pp. 32-36, Jun. 2003.
2. B. E. Carpenter and K. Nichols, "Differentiated Services in the Internet," *Proceedings of the IEEE*, Vol. 90, No. 9, pp. 1479-1494, Sep. 2002.
3. D. Wu, Y. T. Hou, W. Zhu, Y. Zhang, and J. M. Peha, "Streaming Video over the Internet: Approaches and Directions," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, pp. 282-300, Mar. 2001.
4. J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-Friendly Congestion Control," *IEEE Network*, pp. 28-37, May/Jun. 2001.
5. R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet," *IEEE Proceedings in Infocom 99'*, pp. 1337-1345, Mar. 1999.
6. M. Handley et. al., "TCP Friendly Rate Control (TFRC) Protocol Specification," *IETF RFC 3448*, Jan. 2003.
7. D. Sisalem, and A. Wolisz, "LDA+: A TCP-Friendly Adaptation Scheme for Multimedia Communication," *IEEE International Conference on Multimedia and Expo*, pp. 1619-1622, Jul. 2000.
8. I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP Emulation at receivers - flow control for mulitmedia streaming," *Technical Report, North Carolina State University*, Apr. 2000.
9. V. Paxson and M. Allman, "Computing TCP's retransmission timer," *IETF RFC 2988*, Nov. 2000.

Enhancing TCP Throughput and Fairness with a Timer-Based Transmission Control over Heterogeneous Networks

Jongmin Lee[1], Hojung Cha[1], and Rhan Ha[2]

[1] Dept. of Computer Science, Yonsei University,
134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea
{jmlee, hjcha}@cs.yonsei.ac.kr
[2] Dept. of Computer Engineering, Hongik University,
72-1 Sangsoo-dong, Mapo-gu, Seoul 121-791, Korea
rhanha@cs.hongik.ac.kr

Abstract. This paper presents a sender-side TCP congestion control mechanism that improves both throughput and fairness in wired as well as wireless networks. A traditional windows-based congestion control scheme adjusts the transmission rate only when the TCP sender receives an acknowledgement message or it detects a transmission timeout. Therefore, competing flows with different packet roundtrip times experience throughput unfairness because a flow with a short packet roundtrip time increases the transmission rate more quickly than the others do. Moreover, the cumulative acknowledgement message induces burst traffic, which overloads the network. In addition, packet losses delay the packet transmission until the amount of date transmitted without acknowledgements from the receiver equals to the reduced congestion window size, and it degrades transmission performance. The proposed mechanism adjusts the transmission rate based on a timer maintained by the TCP sender. As the mechanism adjusts the transmission rate regardless of the reception of acknowledgement messages, it improves fairness among flows with different packet roundtrip time, and prevents burst traffic caused by the cumulative acknowledgement message. The mechanism has been implemented in the Linux platform, and experienced with various TCP variants in real environments. The experimental result shows that the mechanism improves both throughput and fairness in wired as well as wireless networks.

1 Introduction

TCP employs a window-based AIMD (Additive Increase Multiplicative Decrease) congestion control scheme to adjust the transmission rate [1]. The transmission rate is controlled by the size of the window, which indicates the amount of data that can be transmitted without receiving ACK (ACKnowledgment) messages from the receiver. Since a TCP sender adjusts the window only when it receives ACK, or it detects a packet transmission timeout, the transmission rate

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 650–659, 2006.

increase depends on the ACK reception rate. As RTT (packet RoundTrip Time) becomes longer, ACK arrives more slowly. Therefore, when flows with different RTT share the same bottleneck link, they experience throughput unfairness, because the flow with a short RTT increases the transmission rate more quickly than the others do.

Whereas the additive increase induces the throughput unfairness among flows with different RTT, the multiplicative decrease degrades the transmission performance especially in wireless networks, which have high packet loss rates. When packets are lost, the TCP sender reduces the congestion window, and holds the transmission until the amount of data transmitted without acknowledgements from the receiver equals to the reduced congestion window size. As TCP considers that packets are lost due the network congestion, it decreases the transmission rate whenever it detects packet losses. However, in wireless networks, link error can cause the lost packet, and it induces frequent transmission holds, which degrade the transmission performance severely.

CRWI (Constant Rate Window Increase) [2] increases the congestion window with the consideration of RTT in order to improve fairness among flows with different RTT. However, CRWI does not scale well in heterogeneous networks because it does not consider wireless networks. Moreover, the appropriate parameter value used to determine the amount of the transmission rate increase remains to be investigated. TCP Westwood+ [3], which is the enhanced version of TCP Westwood [4], estimates the network bandwidth, and restores the previous transmission rate quickly when the transmission rate is reduced due to the packet loss. The quick increase can improves the fairness among flows with different RTT, however it still have throughput unfairness, because it is based on the traditional window-based congestion control mechanism. BIC (Binary Increase Congestion control) [5] increase the congestion window additively or logarithmically based on the current window size and the designated window size. BIC improves the transmission performance by increasing the transmission rate quickly. However, it does not consider burst traffic caused by the cumulative ACK.

This paper proposes a new TCP congestion control mechanism called T-TCP (Timer-based TCP). T-TCP maintains a timer, and adjusts the transmission rate based on the timer expiration period. Since T-TCP adjusts the transmission rate regardless of the ACK reception rate, it improves the throughput fairness among flows with different RTT, and prevents burst traffic caused by the cumulative ACK. T-TCP maintains the traditional TCP semantics, therefore it works well with the traditional TCP. Moreover, since it is a sender-based congestion control mechanism, neither routers nor receivers require modifications. T-TCP has been implemented in real systems, and is experienced with other TCP variants such as BIC or TCP Westwood+ in order to validate its performance characteristics. The paper is organized as follows: Section 2 discusses the design of T-TCP, Section 3 describes the details of the implementation, experiments, and the experimental result, and Section 4 concludes the paper.

2 T-TCP (Timer-Based TCP)

T-TCP maintains a timer, which adjusts the transmission interval, in order to adjust the transmission rate. When applications transmit data, the data are inserted into the transmission queue. When the timer has expired, packets in the transmission queue are removed, and are transmitted to the receiver. At the same time, the packet is inserted into the retransmission queue for the purpose of retransmissions in case of the packet is lost. The timer adjusts the transmission rate, and the expiration period is dynamically adjusted regardless of the ACK reception rate. When the sender receives ACK, the corresponding packet is removed from the retransmission queue, and is discarded. When the lost packet is detected, the retransmission occurs using packets in the retransmission queue. Figure 1 illustrates the overall system structure.

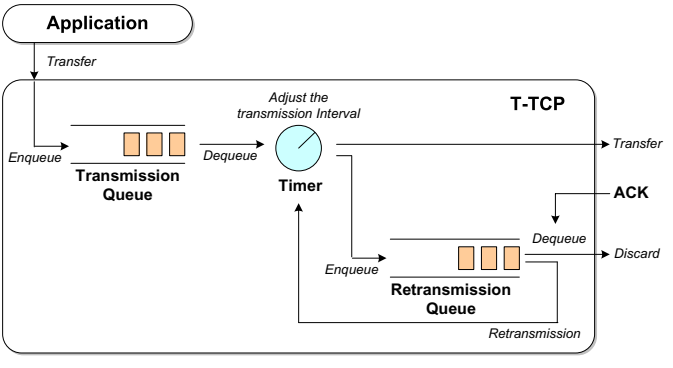


Fig. 1. System overview

T-TCP logically maintains two buffer spaces: a transmission queue and a retransmission queue. The transmission queue temporarily stores packets to transmit, and the retransmission queue stores packets transmitted in order to recover packets lost in networks. However, these queues physically shares one buffer space, and can be explained with the sliding window mechanism. In the sliding window mechanism, packets after the right edge of the window are data to be transmitted, and the transmission queue of T-TCP is the same purpose. Packets between the left edge and the right edge of the window indicate data, which are transmitted but are not acknowledged, yet, so does the retransmission queue of T-TCP. Whereas the sliding window mechanism moves the window depends on the ACK reception, T-TCP moves the window depends on the timer expiration. Table 1 shows the definitions of the parameters used in this paper.

2.1 Determining the Transmission Rate

The traditional TCP controls the network congestion in two phases: a slow start phase and a congestion avoidance phase. In the slow start phase, the traditional

Table 1. Parameter definitions

Parameters	Definitions
ACK	Acknowledgement message
RTT	Packet roundtrip time (milliseconds)
RTT_{dev}	Variation of RTT (milliseconds)
W_{cn}	Congestion window (bytes)
T_p	Timer expiration period (milliseconds)
T_p^{prev}	T_p when the last packet loss occurs (milliseconds)
$T_p^{current}$	Current T_p (milliseconds)
S_t	Amount of data to transmit at once (bytes)
S_t^{prev}	S_t when the last packet loss occurs (bytes)
$S_t^{current}$	Current S_t (bytes)
N_t	Number of period with the current transmission rate
N_c	Number of congestion occurrence with the current transmission rate
α	Parameter used to determine the cause of the packet loss
β	Scaling factor used to determine the transmission rate

TCP increase the congestion window W_{cn} exponentially in order to determine the network capacity. In the congestion avoidance, the traditional TCP increase W_{cn} linearly to prevent further packet losses. T-TCP also controls the network congestion in two phases as the same as the traditional TCP does. Assume that the traditional TCP doubles W_{cn} each packet roundtrip time RTT in the slow start phase, the transmission rate increases as shown as Equation 1 [2].

$$\frac{W_{cn}}{RTT}\ bytes/sec \quad or \quad \frac{W_{cn}}{RTT^2}\ bytes/sec. \tag{1}$$

The increase of the transmission rate is proportional to RTT^2, hence if flows with different RTT compete at the same bottleneck link, the flow with longer RTT experiences a throughput unfairness. However, T-TCP maintains a timer, and the transmission rate is determined by both the expiration period T_p and the amount of data to transmit at once S_t. In the slow start phase, T-TCP reduces the expiration period T_p by half after the number of period N_t passes. At the same time it doubles N_t. Equation 2 calculates the transmission rate increases of T-TCP in the slow start phase.

$$\frac{S_t}{T_p}\ bytes/T_pN_t \quad or \quad \frac{S_t}{T_p^2 N_t}\ bytes/sec. \tag{2}$$

As Equation 2 shows, the transmission rate increase of T-TCP is independent of packet roundtrip time RTT, hence flows competing at the same bottleneck link shares the network bandwidth fairly regardless of their RTT. When packet losses occur, the traditional TCP reduces the congestion window by half, and initiates the congestion avoidance phase. In the congestion avoidance phase, the traditional TCP increases the congestion window W_{cn} by the packet size S_p each

RTT. Equation 3 shows the transmission rate increases of the traditional TCP in the congestion avoidance phase.

$$\frac{S_p}{RTT}\ bytes/RTT \quad or \quad \frac{S_p}{RTT^2}\ bytes/sec. \tag{3}$$

The transmission rate increase in the congestion avoidance phase is also proportional to RTT^2, thus it still have throughput unfairness among flows with different RTT. T-TCP detects packet losses at the same method as the traditional TCP does. When packet losses are detected, T-TCP compares between the current transmission rate and the previous transmission rate when the packet loss occurs recently. If the difference between two transmission rates is small, the packet loss is probably due to the network congestion. In this case, T-TCP increases the transmission slowly to prevent further packet losses after reducing the transmission rate. Equation 4 shows the condition for determining the cause of the packet loss.

$$\left| \frac{S_t^{prev}}{T_p^{prev}} - \frac{S_t^{current}}{T_p^{current}} \right| \leq \alpha, \quad where \quad \alpha = \frac{S_t^{prev} RTT_{dev}}{T_p^{prev} RTT}. \tag{4}$$

Since α considers the variation of RTT, as RTT fluctuates greatly, α becomes large. If T-TCP decides the packet loss is caused by the network congestion, it reduces the transmission rate by half, and increases the transmission rate inverse proportional to the number of the network congestion occurrence with the current transmission rate N_c, as shown in Equation 5.

$$\beta\frac{S_t}{T_p}\ bytes/T_p N_t N_c \quad or \quad \beta\frac{S_t}{T_p^2 N_t N_c}\ bytes/sec. \tag{5}$$

As Equation 5 shows, T-TCP in the congestion avoidance phase increases the transmission rate more slowly as the network congestion occurs frequently. When T-TCP determines the packet loss is caused by the link error, it reduces the transmission rate by half, and reinitiates the slow start phase. With this method, T-TCP copes with the bandwidth changes efficiently, and improves the transmission performance even in networks, which have a large BDP (Bandwidth Delay Product).

2.2 Congestion Control

In a slow start phase, T-TCP increases the transmission rate exponentially to determine the network capacity. T-TCP detects packet losses when the sender receives three duplicated ACK or detects the transmission timeout as the same method as the traditional TCP does. When packet losses are detected, T-TCP examines the lost is caused by the network congestion. If so, T-TCP reduces the transmission rate by half, then initiates a congestion avoidance phase. Otherwise, T-TCP reinitiates the slow start phase in order to improve throughput. In the congestion avoidance phase, T-TCP increases the transmission rate slowly inverse proportional to the number of network congestion occurrences. Figure 2 illustrates the mechanism.

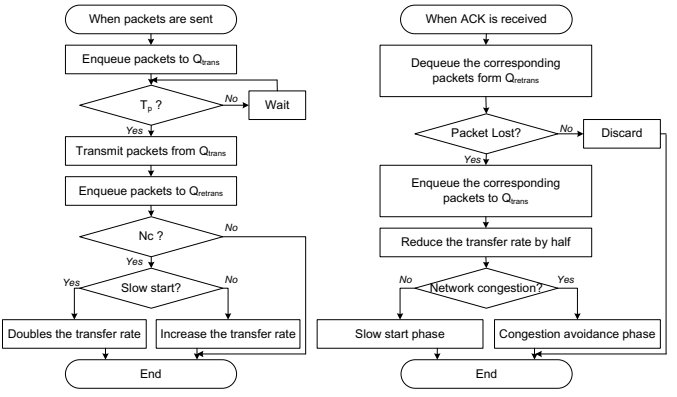


Fig. 2. Congestion control mechanism of T-TCP

Since the timer adjusts the transmission interval regardless the ACK reception, bulk traffic caused by the cumulative ACK can be avoidable. The transmission rate of T-TCP is controlled by both the timer expiration period T_p and the amount of data to transmit at once S_t. When T-TCP detects packet losses, the corresponding packets are inserted into the front of the transmission queue in order to transmit them prior to the others.

3 Experiments

T-TCP has been implemented by modifying the Linux kernel 2.6.11. The experimental system used to verify T-TCP consists of the server and the client. In addition, the network emulator called $NISTnet$, is used to emulate various network environment. The server and the client are connected directly to NISTnet using 100base-T Ethernet and Wireless LAN networks. The well-known protocol analysis tools *tcpdump*, *tcptrace*, and $netperf$ are used to gather and analyze the experimental data [6]. Table 2 shows the parameters and values used in the experiments.

The initial transmission period T_p and the number of packet to transmit at once N_t are set to 50 milliseconds and 1, accordingly with the consideration of that the average packet roundtrip time of the Internet [7] and the large initial window recommendation [8]. The packet size S_t is set to 1448 bytes, which is the largest unfragmented packet size of the application layer. A total of 10 groups of experiments are conducted to evaluate the performance of T-TCP. In addition, the performance of TCP-Reno, BIC-TCP, and TCP-Westwood+ are also experimented for comparison purpose.

Figure 3 shows the trace of transmitted packets of TCP-Reno and T-TCP. As packet round trip time RTT increases, TCP-Reno transmits packets more slowly. Since TCP-Reno increases the congestion window only when the sender receives acknowledgement message ACK, or detects packet transmission timeout, the transmission rate increase depends on RTT. This results the throughput

Table 2. Experimental parameters and values

Parameters	Values
T_p	50 milliseconds
S_t	1448 Bytes
N_t	1
β	0.9
Network Bandwidth	11Mbps (Wireless), 100Mbps (Wired)
RTT	50, 100, 200, 400, 800 milliseconds (emulated)
Drop Probability	0, 0.01, 0.1, 1, 10 percents (emulated)

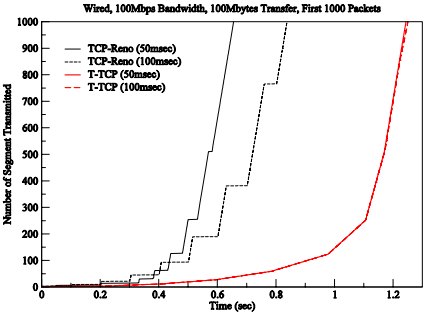


Fig. 3. Trace of transmitted packets

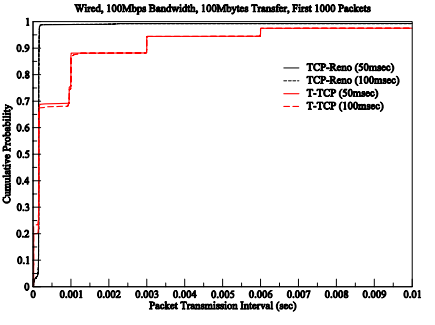


Fig. 4. CDF for transmission interval

unfairness among flows with different RTT. However, T-TCP transmits packets regardless of ACK receptions, therefore the transmission rate increase of T-TCP are the same regardless of their RTT. Consequently, T-TCP flows share the network bandwidth fairly although they have different RTT. Figure 4 shows the CDF (Cumulative Distribution Function) graph for packet transmission interval. The almost all packet transmission interval of TCP-Reno is smaller than 1 millisecond. It means that packets are usually transmitted with a bunch, and it results burst traffic. However, the packet transmission interval of T-TCP spreads over a few milliseconds. Since T-TCP controls the packet transmission interval using the timer, burst traffic can be avoidable.

The throughput comparison with different RTT and different packet drop probability are shown in Figure 5 and 6, accordingly. When RTT increases, throughput of T-TCP decreases a little compared to that of the others. Since the other TCP variants adjust the transmission rate based on the ACK reception, their transmission rate increase depends on RTT. However, the throughput of T-TCP has a little change, because it adjusts the transmission rate regardless of the ACK reception. When packet drop probability is high, the throughput of all TCP variants decreases. However, T-TCP achieves the highest transmission performance compared to the others.

Figure 7 shows the throughput comparison in the Wireless LAN. Two spatial places have been selected based on the signal strength in order to differ

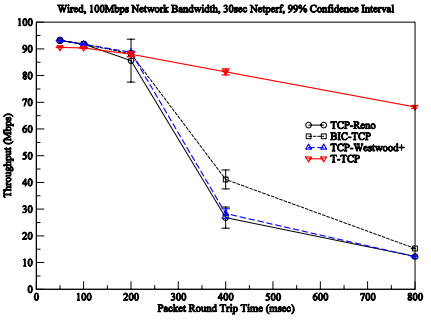


Fig. 5. Throughput with different RTT

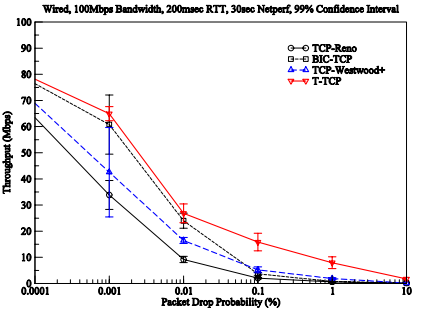


Fig. 6. Throughput with different drop rate

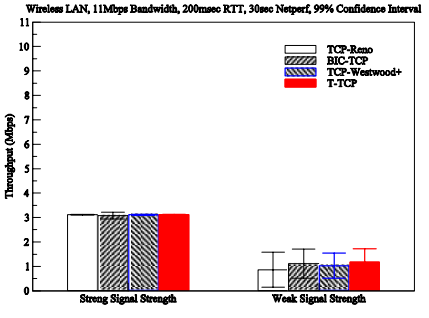


Fig. 7. Throughput in wireless networks

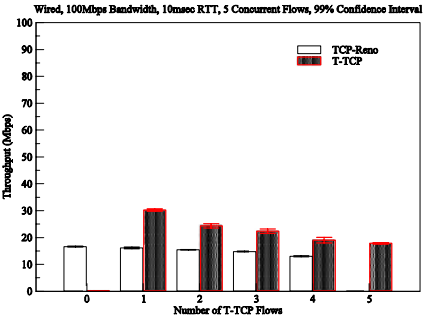


Fig. 8. Concurrent traffic

the packet drop probability. When the signal strength is strong, all TCP variants achieve the similar throughput, since the packet drop probability is low. However, when the signal strength is weak, the packet drop probability becomes high, and T-TCP achieves the highest throughput compared to the others. BIC-TCP achieves the similar throughput compared to T-TCP because of the limited network bandwidth and the small RTT. However, when the packet drop probability becomes high, T-TCP can improve the transmission performance than the others as Figure 6 shows.

An additional 10 groups of experiments were also conducted to evaluate the fairness among T-TCP flows, and the friendliness between T-TCP and the traditional TCP. The fairness implies that T-TCP flow can coexist with the traditional TCP without decreasing the throughput of others. Figure 8 shows the throughput comparison with different combination of five simultaneous T-TCP and TCP-Reno flows sharing the same link. As Figure 8 shows, the average throughput of TCP-Reno does not change regardless of the number of T-TCP flow. This validates that T-TCP is friendly with TCP-Reno. To examine the fairness among T-TCP flows, the fairness index function [9] is used as Equation 6.

$$F(x) = \frac{(\sum x_i)^2}{n(\sum x_i^2)}. \tag{6}$$

Here, x_i is the throughput of the i-th flow, and n is the number of connections. The closer result of $F(x)$ to 1 means that flows share the network bandwidth more fairly. Table 3 shows the fairness index of TCP-Reno and T-TCP.

Table 3. Fairness index

	Number of Simultaneous Flow				
	1	2	3	4	5
TCP-Reno	1	0.99	0.99	0.99	0.99
T-TCP	1	0.99	0.96	0.95	0.94

The fairness index of TCP-Reno is closer to 1 compared to that of T-TCP. Since the transmission rate of T-TCP in the congestion avoidance phase is diverse according to the number of network congestion occurrences, the flow that detects the network congestion repeatedly increases the transmission rate more slowly than the others do. However, this unfairness is tolerable and can be reduced as the transmission time increases.

4 Conclusions

This paper proposed a timer-based TCP congestion control mechanism (T-TCP) where the sender adjusts the packet transmission interval regardless of the ACK reception. Hence, flows competing at the same bottleneck link share the bandwidth fairly regardless of their packet roundtrip time. Moreover, since T-TCP controls the packet transmission interval, bulk traffic, which are occurred due to the cumulative ACK, can be avoidable. When T-TCP detects packet losses, it reduces the transmission rate without holding the transmission during the lost recovery process, therefore transmission performance can be improved especially in wireless networks, which have high packet loss rates.

T-TCP has been implemented and experimented in real environments. As the experimental result shows, T-TCP improves fairness and transmission performance both in wired and wireless networks. As T-TCP adjusts the transmission rate regardless the ACK reception, it can be used in the service differentiation combining with the queue management mechanism, which usually controls the class of flows rather than individual flows. Future research will improve the fairness among T-TCP flows. Developing analytical models and calculating the processing overhead of T-TCP are also yet to be studied.

Acknowledgements

This work was supported in part by the National Research Laboratory (NRL) program of the Korea Science and Engineering Foundation (2005-01352), and the ITRC programs (MMRC, HY-SDR) of IITA, Korea.

References

1. Van Jocobson: Congestion Avoidance and Control, Proceedings of ACM SIGCOMM, (1998) 314–329
2. Sally Floyd: Connections with Multiple Congested Gateways in Packet-Switched Networks Part 2: Two-way Traffic, ACM Computer Communication Review, **21(5)** (1991) 30–47
3. Luigi A. Grieco, Saverio Mascolo: Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control, ACM SIGCOMM Computer Communication Review, **34(2)** (2004) 25–38
4. Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang: TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, Mobile Computing and Networking (2001) 287–297
5. Lisong Xu, Khaled Harfoush, and Injong Rhee: Binary Increase Congestion Control (BIC) for Fast, Long Distance Networks, Proceedings of IEEE INFOCOM'04, Hong Kong (2004)
6. S. Parker, C. Schmechel: Some Testing Tools for TCP Implementers, RFC2398, IETF, (1998)
7. Network Service & Consulting Cooperation: Internet Traffic Report, http://www.internettrafficreport.com
8. M. Allman, S. Floyd, and C. Partridge: Increasing TCP's Initial Window, RFC3390, IETF, (2002)
9. R. Jain, D. Chiu, and W. Hawe: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems, Technical Report DEC-TR-301, Digital Equipment Corporation (1984)

TCP-Friendly Rate Control Scheme Based on RTP

Sunhun Lee and Kwangsue Chung

School of Electronics Engineering, Kwangwoon University, Korea
sunlee@adams.kw.ac.kr, kchung@daisy.kw.ac.kr

Abstract. The multimedia traffic of continuous video and audio data via streaming service accounts for a significant and expanding portion of the Internet traffic. This streaming data delivery is mostly based on UDP. However, UDP does not support congestion control mechanism. For this reason, UDP causes the starvation of congestion controlled TCP traffic which reduces its bandwidth share during overload situation. In this paper, we propose a new TCP-friendly rate control scheme called "TF-RTP (TCP-Friendly RTP)". In the congested network state, the TF-RTP precisely estimates the competing TCP's throughput by using the improved parameters so that it can control the sending rate of the video streams. Therefore, the TF-RTP is able to adjust its sending rate in a TCP-friendly manner and reduce a rate fluctuation. Through the simulation, we prove that the TF-RTP correctly estimates the competing TCP's throughput and improves the stability and TCP-friendliness.

1 Introduction

The Internet has recently been experiencing an explosive growth in the use of audio and video traffics. Such multimedia traffics are delay-sensitive, semi-reliable, and rate-based. However, today's Internet does not attempt to guarantee an upper bound on end-to-end delay and a lower bound on available bandwidth. Accordingly, most of multimedia applications use UDP (User Datagram Protocol) that has no congestion control mechanism. However, the emergence of non-congestion-controlled multimedia traffics threatens unfairness to competing TCP (Transmission Control Protocol) traffic and possible congestion collapse [1], [2].

To avoid such a situation, studies on the congestion controlled rate control schemes have been increasingly done [3], [4]. These researches collect, exchange, and process information about the network status and adjust the sending rate of the end systems based on this information. The information about the network status is extracted from the RTP (Real-time Transport Protocol) that was designed by the IETF (Internet Engineering Tasking Force)'s audio-video transport working group [5]. The RTP is widely used for multimedia communication in the Internet, because it offers the necessary mechanisms for collecting and exchanging information about network load, packet losses, and end-to-end delays [6].

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 660–669, 2006.

In this paper, we propose a novel TCP-friendly rate control scheme called TF-RTP (TCP-Friendly RTP). Our TF-RTP considers the aspects of friendliness of RTP traffic towards competing TCP traffic. It precisely estimates the competing TCP's throughput by using the Padhye's TCP throughput analysis with modified parameters. To estimate the competing TCP's throughput more precisely, TF-RTP improves the parameters related to packet losses and round trip time. Then, the TF-RTP adjusts its sending rate in a TCP-friendly manner and it has fair share with competing TCP traffics. The TF-RTP is designed to achieve an optimal adaptation behavior of streaming transmission rate.

The rest of this paper is organized as follows. In Section 2, we review some of the previous RTP-based works and in Section 3, we discuss the problems in previous works and present various algorithms in the TF-RTP. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper and discusses some of our future works.

2 Related Works

Recently, there have been several proposals for TCP-friendly adaptation schemes [7], [8], [9]. In this paper, TCP-friendliness is the terminology used for non-TCP flow. Non-TCP flows are defined as TCP-friendly when "their long-term throughput does not exceed the throughput of a conformant TCP connection under the same conditions" [10]. In this section, we discuss the previous RTP-based works that control the sending rate of the video streams in a TCP-friendly manner.

2.1 Padhye's TCP Throughput Analysis

One of the most important goals in the previous works is to reach TCP-friendly behavior. The TCP throughput analysis is suitable in helping to describe a TCP-compatible flow. Padhye et al. present an analytical model for the available bandwidth share (T) of TCP connection with S as the segment size, p as the packet loss rate, t_{RTT} as the round trip time, and t_{RTO} as the retransmission timeout [11]. The average bandwidth share of TCP depends mainly on t_{RTT} and p as shown in Eqn. (1):

$$T = \frac{S}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \tag{1}$$

According to [12], this approximation is reasonable for reaching TCP-friendliness. For this reason, many TCP-friendly rate control schemes use this analytical model.

2.2 LDA(Loss-Delay Based Adaptation) Algorithm

LDA controls the sending rate of video streams to the rate given by the TCP throughput model as described in 2.1 [3]. It relies solely on the RTCP (Real-time

Transport Control Protocol) feedback information. While LDA is essentially an AIMD (Additive Increase Multiplicative Decrease) congestion control scheme, it uses some interesting additional elements. The amount of additive increase is determined as the minimum of three independent increase factors to ensure that (a) flows with a low bandwidth can increase their rate faster than flows with a higher bandwidth, (b) flows do not exceed the estimated bottleneck bandwidth, and (c) flows do not increase their bandwidth faster than a competing TCP connection. If packet losses are reported to the sender by RTCP message from receiver, the sender decreases a sending rate by multiplying $(1-\sqrt{p})$. After that, the sending rate is reduced at most to the rate given by the Padhye's equation as shown in Eqn. (2):

$$T_n = MAX(T_{n-1} \times (1-\sqrt{p}), T_{TCP}) \tag{2}$$

LDA uses the Padhye's modeling equation to control the sending rate in a TCP-friendly manner. However, if LDA directly adjusts sending rate to the rate given by the modeling equation, it induces oscillations in the achievable transmission rate. Frequently applying TCP-friendly rate change to the video streams would seriously degrade the video quality. To overcome this problem, LDA adjusts its sending rate to the maximum of two decrease factors, as shown in Eqn. (2). The additional selection rule is applied due to incorrect calculation of TCP-friendly rate, T_{TCP}. It means that the packet loss rate, RTT (Round Trip Time), and RTO (Retransmission Time Out), obtained from receiver's RTCP message, are not appropriate to the Padhye's modeling equation.

2.3 SRTP(Smart RTP) Algorithm

Similar to LDA, SRTP was designed to use RTP and RTCP to exchange feedback information about the RTT and packet loss rate. SRTP is a new rate adaptation scheme which controls the sending rate as smoothly as possible, based on the available bandwidth share. For this purpose, SRTP calculates the TCP-friendly rate in the same way as LDA, and then it determines the sending rate to the maximum of two independent decrease factors to ensure that rate oscillation is a small as much as possible [4].

$$If\ (\overline{T_{SRTP}} > T_{TCP})$$
$$T_{SRTP} = MAX\,(\beta \times \overline{T_{SRTP}} + (1-\beta)(\overline{T_{SRTP}} \times (1-\sqrt{p})), T_{TCP})$$

Fig. 1. Rate control scheme in SRTP

For the case of congested state, the sending rate (T_{SRTP}) is determined as Fig. 1, where $\overline{T_{SRTP}}$ is the current sending rate and β is the weighting parameter for rate smoothing. However, incorrect calculation of TCP-friendly rate, T_{TCP},

remains unchanged due to inaccuracy of three parameters (packet loss rate, RTT, and RTO). Therefore, SRTP applies additional selection rule for smoothing rate control similar to LDA.

3 TF-RTP's Algorithm

In this section, we discuss the problems in previous works and present various algorithms in the TF-RTP. The process of optimal transmission rate control in the TF-RTP consists of three stages: (a) more improved estimation of packet loss rate, RTT, and RTO, (b) calculating the TCP-friendly rate, and (c) adjusting the sending rate.

3.1 Problems of Previous Works

LDA and SRTP use Padhye's modeling equation for rate control in a TCP-friendly manner. However previous works induce oscillations in the achievable transmission rate because they directly adjust sending rate of video stream to the rate given by the Padhye's modeling equation. Moreover, calculated transmission rate is lower than average TCP's throughput. To resolve this problem, LDA and SRTP apply the additional selection rule for smoothing rate control. In LDA and SRTP, packet loss rate and RTT are estimated by Eqn. (3) and Eqn. (4). N_{real} is number of actually received packet, N_{max} is maximum number of received packet, N_{first} is number of first received packet [3]. $\overline{t_{RTT}}$ is the current RTT, α is a weighting parameter that is set to 0.8.

$$p = \frac{N_{real}}{N_{max} - N_{first}} \tag{3}$$

$$t_{RTT} = (\alpha \times \overline{t_{RTT}}) + (1 - \alpha) \times t_{RTT} \tag{4}$$

We performed a simple experiment to verify the problems in previous works. Fig. 2 presents the estimated TCP-friendly rate in previous works on a 10Mbps with two competing TCP connections. In the steady state, packet losses are occurred at 12, 18, 32, 39, 40th RTCP intervals. At each loss situation, packet loss rate, RTT, and estimated TCP-friendly rate (T) are follows:

- 12th RTCP interval: loss rate=0.0022, RTT=265ms, T=1.91Mbps
- 18th RTCP interval: loss rate=0.0023, RTT=250ms, T=1.9Mbps
- 32th RTCP interval: loss rate=0.0022, RTT=265ms, T=1.94Mbps
- 39th RTCP interval: loss rate=0.0025, RTT=265ms, T=0.74Mbps
- 40th RTCP interval: loss rate=0.0022, RTT=107ms, T=1.9Mbps

Fig. 2 shows that the estimated TCP-friendly rate in previous works is lower than the competing TCP's throughput (3.0Mbps). Also, oscillation of the estimated rate is very large in consecutive packet losses. It means that the competing TCP's throughput in previous works can be underestimated, because of overestimated RTT and packet loss rate.

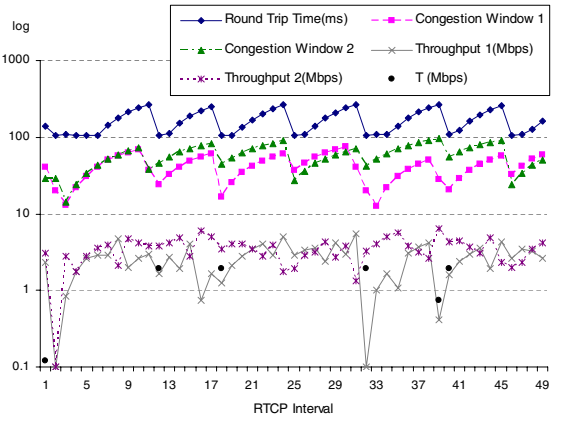


Fig. 2. Incorrect throughput estimation

3.2 Improved Parameters

The TF-RTP has more precisely calculated packet loss rate and RTT to estimate the competing TCP's throughput. Unlike previous works, the TF-RTP uses event-based and cumulative packet loss rate, where a loss event consists of one or more packets dropped within a single RTCP interval. Therefore, the TF-RTP estimates packet loss rate to small value, compared with packet-based method. Event-based and cumulative packet loss rate can be estimated as follows:

$$p = \frac{\text{loss event}}{\text{Number of received packets/Number of loss event}} \tag{5}$$

TF-RTP calculates the RTT and RTO, similar to conventional TCP. However, TF-RTP uses a quadratic low-pass filtered RTT to reduce the periodic fluctuations. Low-pass filtered RTT is reported to the sender by RTCP Receiver Report and then it is low-pass filtered at a sender side, once more. In case of RTO, the TF-RTP could derive the RTO value according to the usual TCP algorithm [13]. TF-RTP can estimate the quadratic low-pass filtered RTT and RTO as follows:

$$t_{QSRTT}(n) = (\Delta \times t_{QSRTT}(n-1)) + (1-\Delta) \times t_{SRTT} \tag{6}$$

$$t_{RTO} = t_{QSRTT} + 4 \times SRTT_{Variation} \tag{7}$$

where t_{SRTT} is the low-pass filtered RTT at a sender side, Δ is a weighting parameter that is set to 0.875 and $SRTT_{Variation}$ is the variance of the SRTT.

With improved parameters, the TF-RTP can estimate the competing TCP's throughput, more precisely. Therefore, it improves the fairness with competing TCP traffic and reduces the oscillation of transmission rate. Fig. 3 shows that TF-RTP with improved parameters precisely estimates the competing TCP's

throughput (3.11Mbps), compared with previous works, which presented in Fig. 2. Moreover, the oscillation of the estimated throughput is considerably decreased in the consecutive packet losses. At each loss situation, packet loss rate, RTT and estimated TCP-friendly rate (T) are follows:

- 12th RTCP interval: loss rate=0.0004, RTT=161ms, T=2.95Mbps
- 18th RTCP interval: loss rate=0.0004, RTT=168ms, T=2.84Mbps
- 32th RTCP interval: loss rate=0.0003, RTT=179ms, T=3.11Mbps
- 39th RTCP interval: loss rate=0.0003, RTT=188ms, T=2.92Mbps
- 40th RTCP interval: loss rate=0.0004, RTT=178ms, T=2.86Mbps

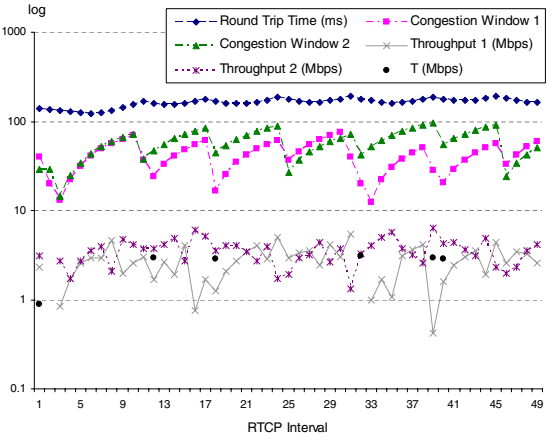


Fig. 3. Throughput estimation in TF-RTP

Through the simple experiments, it is confirmed that LDA and SRTP incorrectly estimate the competing TCP's throughput. However, our TF-RTP can estimate the TCP-friendly rate based on the improved three parameters.

3.3 TF-RTP's Rate Control Scheme

Having got the TCP-friendly rate, the TF-RTP can adjust its sending rate to the estimated rate. The TF-RTP is a kind of AIMD algorithm with increase and decrease values determined dynamically on the basis of the current network state [14]. The TF-RTP decides network state based on the packet loss rate that can be reported from the Receiver Report of RTCP. If the packet loss rate is higher than zero, then the network state is decided to be congestion_state. In this case, packet losses were caused by network congestion. Otherwise, the network state is decided to be stable_state.

Fig. 4 shows the pseudo code of our TF-RTP's rate control algorithm. For the congestion_state, the sender has to reduce the sending rate for the network stability and the TCP-friendliness. Unlike LDA and SRTP, the TF-RTP directly

adjusts the sending rate of video streams to the rate given by the Padhye's modeling equation as described in Eqn. (1), without any additional selection rule. For the case of stable_state, the sending rate can be increased by additive increase factor. This increase factor does not exceed the increase of the competing TCP connection under the same network conditions. Thus, the TF-RTP sender should make the sending rate to be same with TCP connection, in order to ensure the TCP-friendliness. For this requirement, TF-RTP increases sending rate by R_{Inc} at each RTCP interval instead of each RTT.

$$
\begin{aligned}
&If\,(p > 0)\ : \text{Congestion\_State}\\
&\quad Sending\_Rate = T;\\
&Else(p == 0)\ : \text{Stable\_State}\\
&\quad P_{Inc} = \frac{RTCP\_Period}{t_{SRTT}};\\
&\quad R_{Inc} = \frac{P_{Inc}}{t_{SRTT}};\\
&\quad Sending\_Rate = Current\_Rate + R_{Inc}
\end{aligned}
$$

Fig. 4. Rate control scheme in TF-RTP

4 Simulation and Evaluation

4.1 Simulation Environment

In this Section, we present our simulation results. Using the ns-2 simulator [15], the performance of the TF-RTP has been measured, with respect to throughput, fairness, and TCP-friendliness. To emulate the competing network conditions, background TCP traffics are introduced.

Fig. 5 shows the topology for our simulations. All of our experiments use a single bottleneck topology and the bottleneck queue is managed with the drop-tail mechanism.

4.2 Performance Evaluation

Fig. 6 (a) shows that our TF-RTP dynamically controls the sending rate of video stream on the basis of the current network state. A simulation was performed using a set of 1 TF-RTP and competing 2 TCP connections. On the beginning, we set the sending rate of video stream to 3Mbps. If the packet losses are detected by RTCP message at sender side, then the TF-RTP calculates the event-based packet loss rate, smoothed RTT, and RTO. And then, it estimates the average throughput of the competing TCP connections and adjusts the sending rate to the estimated rate. Because our scheme more appropriately estimates the TCP-friendly rate with improved three parameters, it provides TCP-friendliness and lower rate fluctuations, without any additional

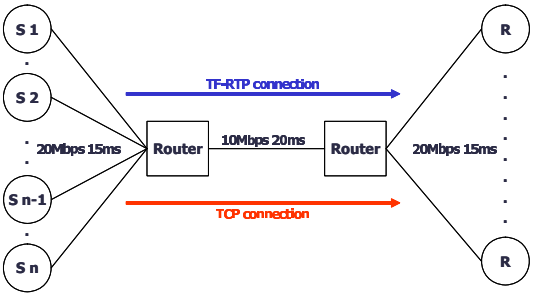
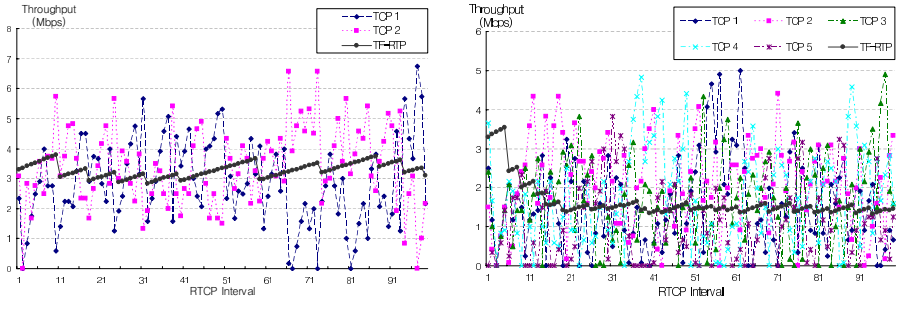


Fig. 5. Simulation environment



(a) TF-RTP vs. 2 TCP (b) TF-RTP vs. 5 TCP

Fig. 6. Throughput comparison between TF-RTP and TCP

mechanisms. Approximately, TCP connections share about average 3.2Mbps and TF-RTP about average 3.1Mbps.

To extend the experiment, a simulation was performed using a set of 1 TF-RTP and competing 5 TCP connections. Fig. 6 (b) shows our TF-RTP has a fair share of bandwidth with the competing TCP connections and high stability, as same as Fig. 6 (a). Approximately, TCP connections share about average 1.7Mbps and TF-RTP about average 1.63Mbps.

We use the fairness index to better understand the TCP-friendly behavior when introducing TF-RTP connections to share a common bottleneck link with TCP connections. In many experiments, fairness indexes are described in Table 1.

Table 1. Fairness evaluation

Competing TCP flows	vs. 1 TCP	vs. 2 TCP	vs. 5 TCP	vs. 9 TCP
Avg. TCP's Throughput	4.78 Mbps	3.2 Mbps	1.71 Mbps	1.02Mbps
TF-RTP's Throughput	4.32 Mbps	3.1 Mbps	1.63 Mbps	0.97 Mbps
Fairness	0.9	0.96	0.95	0.95

$$Fairness = \frac{\text{TF-RTP's Thoughput}}{\text{Avg. TCP's Throughput}}$$

We see that for all cases, except for competing with 1 TCP connection, the TF-RTP has a fair share of bandwidth with competing TCP connections.

5 Conclusion

It is very important for multimedia streaming to be TCP-friendly, because a dominant portion of today's Internet traffic is based on TCP. The multimedia streaming traffics are expected to react on congestion by adapting their transmission rate and have to maintain the fairness with other traffics, in order to be efficiently transported over the Internet.

In this paper, to overcome limitations of the previous works based on RTP, we propose a new rate control scheme. Our TF-RTP more precisely estimates the average throughput of the competing TCP connections with improved parameters and adjusts the sending rate of video streams in a TCP-friendly manner. As a result, the TF-RTP has less variation in the transmission rate and improves the TCP-friendliness. Simulation results have shown that the TF-RTP has a better performance than previous works, with improved three parameters.

In the future, we will further enhance the stability of sending rate and perform more experiments in various network environments.

Acknowledgement

This research has been conducted by the Research Grant of Kwangwoon University in 2005. This research also has been conducted by Korea Science and Engineering Foundation under contract number R01-2005-0000-10934-0(2005).

References

1. S. Floyd and F. Kevin: Router mechanisms to support end-to-end congestion control. Technical Report, LBL-Berkeley. (1997)
2. S. Cen, C. Pu, and J. Walpole: Flow and congestion control for internet streaming applications. Multimedia Computing and Networking. (1998)
3. D. Sisalem, and H. Schulzrinne: The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV). (1998)
4. B. Song, K. Chung, and Y. Shin: SRTP: TCP-friendly congestion control for multimedia streaming. 16th International Conference on Information Networking. (2002)
5. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson: RTP: A transport protocol for real-time applications. IETF, RFC 1889. (1996)
6. I. Busse, B. Deffner, and H. Schulzrinne: Dynamic QoS control of multimedia applications based on RTP. IEEE Computer Communications. (1996)
7. R. Rejaie, M. Handley, and D. Estrin: RAP: An end-to-end rate based congestion control mechanism for real-time streams in the Internet. IEEE INFOCOMM. (1999)
8. I. Rhee, V. Ozdemir, and Y. Yi: TEAR: TCP emulation at receivers - flow control for multimedia streaming. Technical Report, NCSU. (2000)

9. S. Floyd, M. Handley, J. Padhye, and J. Widmer: Equation-based congestion control for unicast applications. In Proceedings of SIGCOMM. (2000)
10. S. Floyd and K. Fall: Promoting the Use of End-to-end Congestion Control in the Internet. IEEE/ACM Transactions on Networking. (1999)
11. J. Padhye, V. Firoiu, D. Towsley, and J. Kurpose: Modeling TCP throughput: A simple model and its empirical validation. ACM SIGCOMM. (1998)
12. J. Padhye, J. Kurose, D. Towsley, and R. Koodli: A model based TCP-friendly rate control protocol. International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV). (1999)
13. V. Jacobson: Congestion avoidance and control. ACM SIGCOMM. (1998)
14. W. Stevens: TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms. RFC2001. (1997)
15. UCB LBNL VINT: Network Simulator ns (Version 2). http://www.isi.edu/nanam/ns/

Improved Wireless TCP by Discriminative Control Using Loss Cause Reasoning

Junseo Son[1] and Sungchang Lee[2]

[1] Department of Information and Telecommunication Engineering,
Graduate School of Hankuk Aviation University, Korea
jsson@hau.ac.kr
[2] School of Electronics, Telecommunication, and Computer Engineering,
Hankuk Aviation University, Korea
sclee@hau.ac.kr

Abstract. TCP is originally developed for wired networks, which have lower packet loss from transmission errors. TCP controls the amount of packet transmission efficiently and the confidence of data transmission and high throughput is guaranteed. However the legacy wired TCP does not fit wireless networks which have high Bit Error Rate (BER) inducing frequent packet loss. In this paper, we discriminate the cause of packet loss and based on that, adapt the control algorithm so that the TCP used in wired networks is improved to perform well in wireless networks. The proposed algorithm distinguishes whether the reason for a packet loss is due to the network congestion or some random error, and accordingly adapts the congestion window and slow starts threshold value. For maintaining stable state after random error, the proposed algorithm keeps current values. Detailed simulations are provided to show how the detection of packet loss and the bandwidth estimation scheme serves to control excessive decrease of the congestion window caused by transmission errors in wireless networks.

1 Introduction

In the past decade, we used a cable modem which makes use of a cable network to connect at Internet. At the present time, wireless mobile communication has become widespread with the increasing popularity of wireless network and wireless device. Mobile users request to equal to Internet transmission quality of a wired net service, and need to be able to perform data transfers while they are on the move. In wired networks, many applications run on top of the transmission control protocol (TCP). TCP implements flow control by means of a sliding window algorithm. TCP Tahoe, Reno and New Reno, which includes the slow start (SS), additive increase and multiplicative decrease (AIMD), congestion avoidance (CA) and fast retransmit/recovery algorithms to adjust the window size, have contributed much success to date. In particular, in wired network, the TCP congestion avoidance mechanisms are sufficient to provide good service. Most of the packet loss and irregular delay occur from router congestion, and the network

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 670–679, 2006.

congestion causes segment losses and brings about the reception of duplicate acknowledgments (DUPACKs) or the expiration of a timeout. However, in wireless network, TCP has some major problems because wireless links are subject to high bit-error-rate (BER) and wireless networks may suffer high packet loss rate. Wireless links exhibit much poorer performance than wired and, even more importantly, the effects of this behavior are erroneously interpreted by TCP, which reacts to network congestion every time it detects packet loss, even though the loss itself may have occurred for other reason (e.g., a fading, a noisy, hand-off). Several schemes have been proposed to design a reliable transport protocol for the wireless networks. The algorithms are designed to reduce the negative effects of non-congestion related losses. Therefore, several approaches are proposed, like the split-connection, the localized link layer methods, and modifications to the base TCP etc.[4][5][7][10][11] Some solution is better for a particular topology, some other performs better under different conditions. For examples, Snoop introduces snooping agent at the base-station to monitor every packets sent across the TCP connection that have not yet been acknowledged by the receiver and maintain a buffer of TCP segments. But the disadvantage of snoop is that it cannot detect disconnection due to handoff. And end-to-end TCP semantics are maintained. WTCP, TCP Westwood[2][12][13], TCP Veno[1][9] are just in that approach. The rest of this paper is organized as follows. In section 2, The related works and their background will be discussed. In section 3 describes our proposed scheme. This scheme is composed of the sensing of a packet loss and bandwidth estimation. Simulation results will be discussed in section 4 and at finally conclusions will be presented in section 5.

2 Congestion Control Schemes for Performance Improvement

As illustrated Fig. 1, the state of TCP is composed of each state - Slow start, Congestion Avoidance, Fast retransmit & Fast recovery.

2.1 The Bandwidth Estimate Scheme

When congestion occurs, the source can avoid an unnecessary bandwidth reduction by the available bandwidth estimation. TCP Westwood [2][12][13] estimates the available bandwidth by means of the rate of returning ACKs. Using this scheme which estimates an available bandwidth, we can get an improved data throughput by controlled slow start threshold. In TCP Westwood algorithm, a sample of bandwidth is measured as:

$$S\_BWE_k = \frac{d_k}{(t_k - t_{k-1})}$$

where t_k is the time the source receives an ACK, t_{k-1} is the time the previous ACK was received and d_k is a corresponding amount of data. This is computed

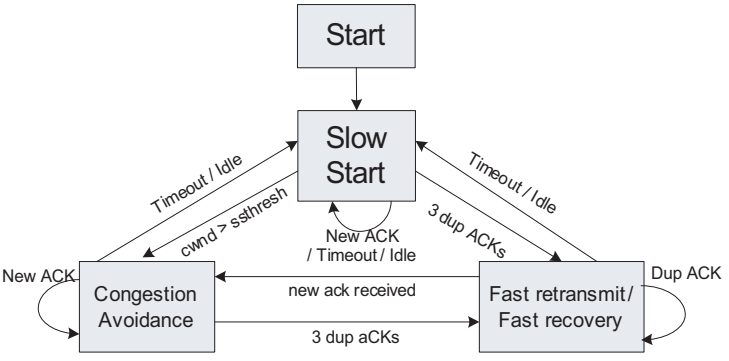


Fig. 1. State Diagram of TCP

every time t_k, and samples S_BWE_k are low-pass filtered using the time-varying filter:[2]

$$BWE_k = \frac{\frac{2\tau}{t_k - t_{k-1}} - 1}{\frac{2\tau}{t_k - t_{k-1}} + 1} BWE_{k-1} + \frac{S\_BWE_k + S\_BWE_{k-1}}{\frac{2\tau}{t_k - t_{k-1}} + 1}$$

where τ is the filter time constant. This low-pass filter becomes a filter with constant coefficients as

$$BWE_k = BWE_{k-1} * \alpha + (S\_BWE_{k-1} + S\_BWE_k) * \frac{(1-\alpha)}{2}$$

where α is constant coefficient (a typical value is α=0.93548). TCP Westwood is to use the BWE_k to set the congestion window (cwnd) and the slow start threshold (ssthresh) , during a congestion phase.

2.2 Distinguishing the Reason for a Packet Loss Scheme

Distinguishing between congestion loss and random loss is important for adjusting window size and utilizing available bandwidth efficiently. In TCP Vegas, the approach is to use an estimation scheme of backlog value N at the router queue for measuring state of network [3]. First the round trip time (RTT), base RTT and congestion window size are measured. Then the sender computes the so-called Expected-rate and Actual-rate using values referred above,

$$Actual\_rate = cwnd/RTT$$
$$Expected\_rate = cwnd/BaseRTT$$
$$DIFF = Expected - Actual$$

where cwnd is the current TCP window size, BaseRTT is the minimum of measured RTT, RTT is the smoothed RTT, and Diff is the difference between the expected and actual rates. If RTT>BaseRTT, there is a bottleneck link. The RTT estimation must be different way and N is measured as

$$RTT = BaseRTT + N/Actual\_rate$$
$$N = Actual\_rate * (RTT - BaseRTT) = DIFF * BaseRTT$$

But TCP Vegas doesn't consider which packet losses have occurred by congestion or not. As mentioned, we need to distinguish between congestion loss and random loss for improving TCP performance. TCP Veno[1][9] does that by using the parameter N of Vegas, i.e, if packet loss is detected while connection is in the congested state (N> β), it assumes that the loss is due to congestion, otherwise it assumes that the loss is random loss. By using such discrimination, it is possible to improve TCP throughput because the slow start threshold and congestion window size are not decreased wastefully. In this paper, in case of packet loss due to non-congestion reasons, we assume there is no need to decrease cwnd. Although if it is a random loss, if the packet losses are detected consecutively, we assume they are due to congestions.

3 Improved TCP Using Loss Cause-Discriminative Control

In this section, we propose a solution to improve the performance of TCP throughput over wireless networks based on the end-to-end approach. It makes up the bandwidth estimation and loss cause-discriminative schemes. We design the IDC(Improved Discriminative Control) TCP as taking the benefits of the both scheme. IDC keeps current cwnd and ssthreshold value in Bit-error state. Because of that, IDC maintains stable state than the other TCP mechanism. Most important advantages are as follows:

1. Suitable to the situation congestion processing,
2. The decrease of unnecessary bandwidth waste,
3. The increase of the throughput from such results.

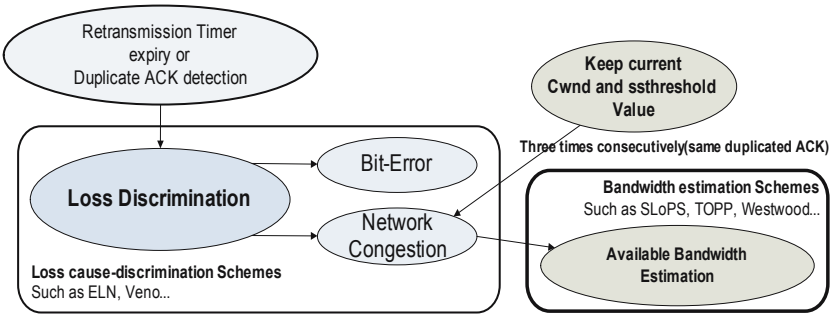


Fig. 2. IDC TCP Architecture Diagram

For using our proposed algorithm, the state of Fast retransmit/Fast recovery a part of the existing TCP state diagram can be changed as shown Figure 3.

The general idea is to use a value N to distinguish whether the factor of loss is due to the network congestion or the random error. If the loss is due to the network congestion, ssthresh is adjusted to an estimated bandwidth value. Otherwise, the source keeps currently cwnd and ssthresh value. But, in this case, if the same duplicated ACK was received three times consecutively, that is considered the network congestion. Because a constant value N the standard of network condition, is not a reality value but the mathematical value of a measurement. After TIMEOUT occurred, ssthresh is applied bandwidth estimation scheme and cwnd is set 1.

The pseudo code of the proposed algorithm is the following:

```
if (3 duplicate ACKs) then  // 3rd duplicate ACK received
   if(N >= B) then // congestion loss is most likely to have occurred
                   // B = beta
       ssthresh = (BWE * min_RTT)/size
       if(cwnd$>$ssthresh) then
           cwnd = ssthresh
       n=0
       end if

    else if(N < B) then   // random loss is most likely to have occurred
        if(n=3) then // this is not random loss
            ssthresh = (BWE * min_RTT)/size
            n = 0
            if(cwnd  > ssthresh) then
                cwnd = ssthresh
            end if
        else
            n = n+1  // keep currently cwnd and ssthresh
        end if
    end if
end if

if (Timeout) then    // a TIMEOUT expires
    ssthresh = (BWE * min_RTT)/size
    if (ssthresh < 2)
        ssthresh = 2
    end if
    cwnd = 1
    n=0
end if
```

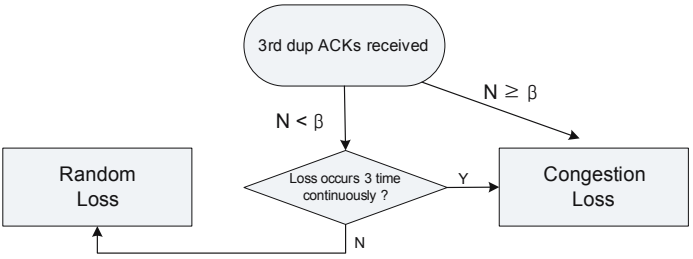


Fig. 3. State diagram of Fast Retransmit/Fast Recovery

4 Simulation Results and Analysis

4.1 Simulation Topology

Our simulation based on hybrid (Wired and Wireless) Networks, the simulation topology is composed of each nodes. A source node is a general fixed node, a destination node is mobile node, and a base station which connects a wired and wireless network is located in the center. All results are obtained using the ns2 simulator [8]. As illustrated in Figure 4, the wired link capacity is 10 Mbps and one-way propagation time of 45ms. Also the wireless network has a very short 2 Mbps wireless link capacity with a propagation time of 0.01ms than wired network. The error model assumed exponential error model.

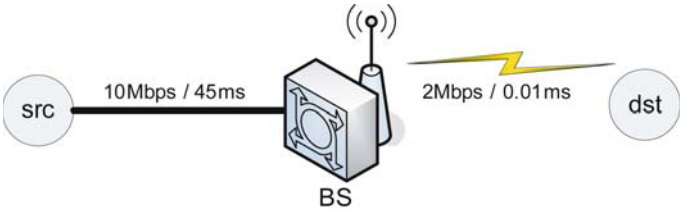


Fig. 4. Simple Simulation Topology (Hybrid Network)

4.2 Simulation Results and Analysis

Figure 5 shows the result of comparison with other TCP schemes when wireless link bandwidth is controlled from 1Mbps to 8Mbps at 0.5% fixed error rate. In spite of large available bandwidth, the protocols Newreno and Reno can use only 0.3Mbps∼0.4Mbps at high BERs because their behavior is unfit for handling such high BER conditions. On the other hand, in the case of TCP Veno and our proposed algorithm, the throughput increases steadily with respect to available bandwidth. In fact the graphs in fig. 5 shows that proposed algorithm's

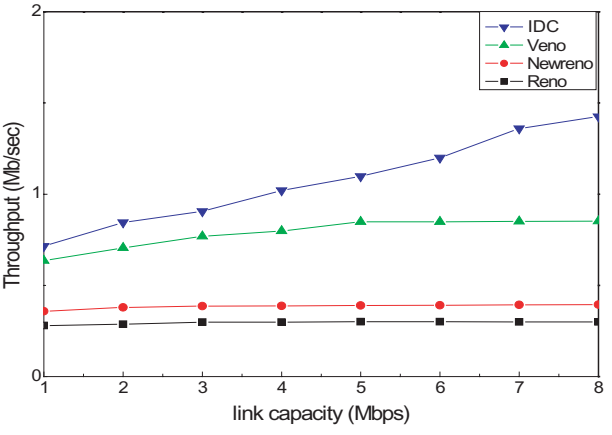


Fig. 5. Throughput vs Error rate

throughput increases almost linearly with the bandwidth as opposed to TCP Veno whose throughput almost saturates at high bandwidth. When the wireless link bandwidth reaches 8Mbps, proposed TCP is 450% better than Reno with respect to throughput.

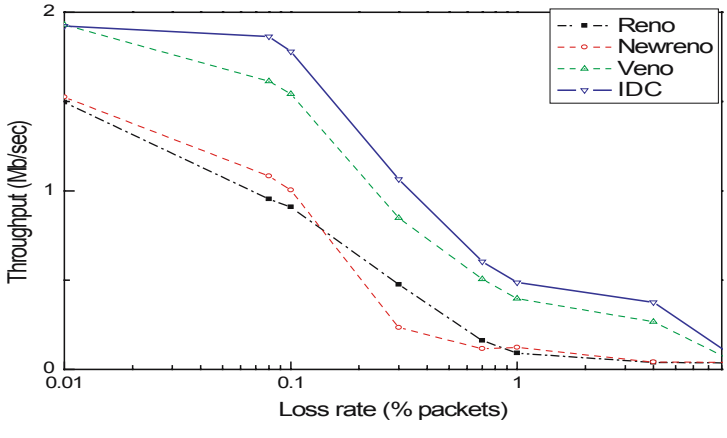
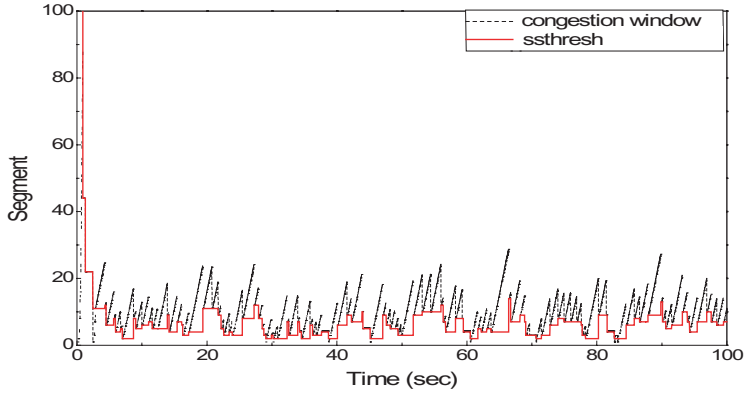
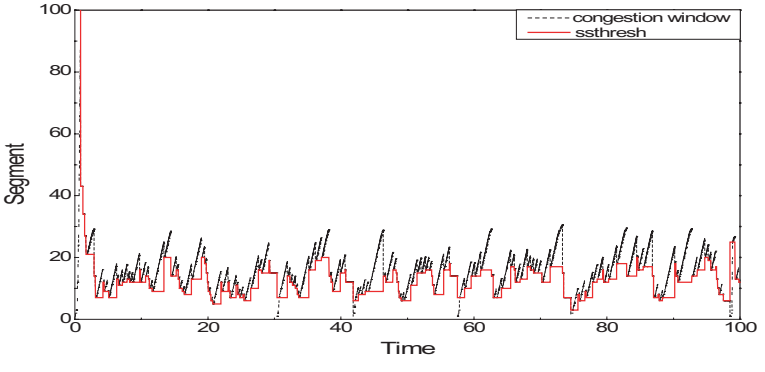


Fig. 6. Throughput vs Loss rate

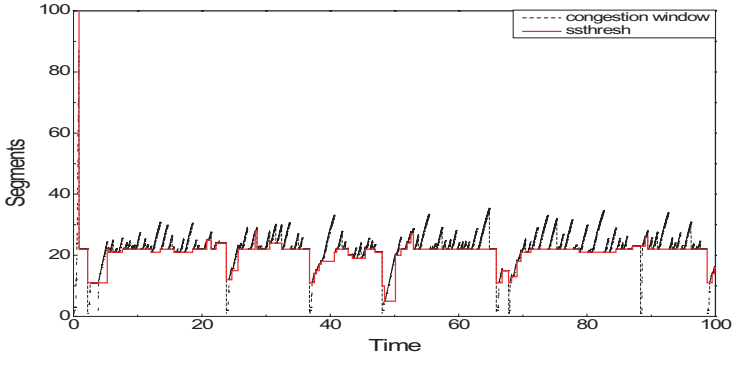
Figure 6 shows a result of throughput measurement as the error rate is varied. We fixed the wireless link bandwidth at 2 Mbps. In a variation of the error rate from 0.01% to 0.1%, the throughput of Reno drops down to 0.5Mbps, but that of proposed TCP drops down to only 0.15Mbps, and that is still able to use the available bandwidth of a link. However the variations of throughput rates of each TCP scheme with respect to error or loss rate are the same.



(a)Reno



(b)Veno



(c)IDC

Fig. 7. Cwnd and ssthreshold vs Time

The figure 7 shows the measurement of cwnd and ssthresh for TCP Reno, Veno, proposed TCP with respect to time. The Bandwidth and error rate were fixed at 2Mbps and 0.1% respectively. We find that both TCP Veno and proposed algorithm are more efficient than Reno from these figures. In the case of our proposed TCP, the reason for unstable performance from 0 to 5 seconds is that the calculation of the beginning of an available bandwidth is not measured accurately. But over the subsequent time, it shows that proposed algorithm maintains higher value of segment size than other TCP schemes in a stable manner. Also it is clear that computing the available bandwidth makes faster recovery possible compared to the other schemes.

5 Conclusions

In wireless network, TCP has some major problems because wireless links are subject to high bit-error-rate and wireless networks may suffer high packet loss rate. Wireless links exhibit much poorer performance than wired and, even more importantly, the effects of this behavior are erroneously interpreted by TCP, which reacts to network congestion every time it detects packet loss, even though the loss itself may have occurred for other reason. In this paper, we discriminate the cause of packet loss and based on that, adapt the control algorithm so that the TCP used in wired networks is improved to perform well in wireless networks. The proposed algorithm distinguishes whether the reason for a packet loss is due to the network congestion or some random error, and accordingly adapts the congestion window and slow starts threshold value. In order to evaluate the performance of proposed algorithm, we compared it with Reno, Newreno, Veno TCP via simulations. In simulation results, proposed TCP provided better throughput than other TCP protocols, such as Reno and Newreno, used in wired network when BER is high.

Acknowledgment. This research was supported by IRC (Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyounggi-Province Regional Research Center designated by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. Cheng Peng Fu, Soung C. Liew : TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks, IEEE JOURNAL, Feb. 2003
2. Saverio Mascolo, Clasudio Casetti, Mario Gerla, M. Y. Sanadidi, Ren Wang : TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, ACM SIGMOBILE, 2001
3. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson : TCP Vegas: New techniques for congestion detection and avoidance., in Proc.SIGCOMM'94, London, U.K., Oct. 1994, pp.24-35

4. V. Jacobson : Congestion Avoidance and Control., in Proc. SIGCOMM'88, Stanford, CA, pp.314-329.
5. V. Jacobson : Modified TCP Congestion Avoidance Algorithm., Apr.30, 1990
6. W. Stevense : TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm, RFC 2001, Jan. 1997, URL: http://www.ietf.org/rfc/rfc3041.txt/
7. Bakre, A.; Badrinath, B.R. : II-TCP: indirect TCP for mobile hosts Distributed Computing Systems, 1995, Proceedings of the 15th International Conference on, 1995, pp. 136-143
8. ns-2 network simulator (ver 2.6). LBL, URL : http://www.isi.edu/nsnam/ns
9. C. P. Fu : TCP Veno: End-to-end Congestion Control Over Heterogemeous Netowkrs, Ph.D. dissertation, The Chinese Univ. Hong Kong, 2001
10. Bakre, A.. Badrinath : I-TCP : indirect TCP for mobile hosts Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on, 1995, pp. 136-143
11. K.Brown and S. Singh : M- TCP : TCP for Cellular Networks, ACM SIGCOMM Computer Communication Review , October 1997, Volume 27 Issue 5
12. M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo : TCP Westwood: Congestion Window Control Using Bandwidth Estimation, In Proceedings of IEEE Globecom 2001, San Antonio, Texas, USA, November 25-29, Volume: 3, pp 1698-170
13. Claudio Casetti, Mario Gerla, Saverio Mascolo, M.Y. Sansadidi, and Ren Wang : TCP Westwood: End-to-End Congestion Control for Wired/Wireless Network" In Wireless Networks Journal 8, 467-479

A Method to Alleviate Unfairness Between HSTCP Flows with Different RTT*

Dong-Chun Ahn[1], Seung-Joon Seok[2], Kyung-Hoe Kim[1], and Chul-Hee Kang[1]

[1] Korea University, Seoul Korea
[2] Kyungnam University, Masan Kyungnam Korea
{a0r0t, kyunghoe, chkang}@widecomm.korea.ac.kr
http:widecomm.korea.ac.kr

Abstract. High-speed TCP (HSTCP) is a protocol that is proposed to take advantage of high capacity bandwidth of backbone network links. HSTCP is able to effectively support the large scale congestion window as compared to Reno TCP. HSTCP is a promising protocol that is getting most attention because it can accommodate existing regular TCP in network and can implement easily by modifying only sender side of TCP transmission protocol. However, HSTCP does not reflect the property of RTT in parameters that increase the congestion window as Reno does. Consequently, the flows experience the serious unfairness when they compete to get the limited resource of network. Especially, the problem appears severely as RTT ratio between the flows grows larger. Therefore, this paper proposes F-HSTCP as a new protocol that reflects the property of RTT as well as current window size in AIMD parameters. This paper also shows that F-HSTCP can resolve the unfairness problem through simulations. F-HSTCP can embrace the difference of RTT.

1 Introduction

With the advance of communication technologies, available bandwidth of network is increased day by day. At the same time, the number of internet users is dramatically increased and also the size of transferred data between users is made larger more and more. So it is required that the transport protocols are able to support the high speed network environment. It is, however, hardly ensured that current transport protocols, such as Reno TCP, get high speed transmission rate over 10Gbps. This is because it theoretically requires less packet loss rate than one caused by physical BER, and because it takes a long time to recover to the window level that the flow was before the congestion. To address this problem, the High-BDP protocols such as High-Speed TCP (HSTCP), Scalable TCP (STCP), FAST TCP, XCP etc. have been proposed so far. These High-BDP (Bandwidth Delay Product) protocols mainly deal with large congestion

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 680–689, 2006.

window problem, because it should be large when TCP flows go through high speed or delayed link. In this paper, we treat only HSTCP among those protocols, because HSTCP has similar response function characteristics to ones of regular TCP and excellent compatibility with regular TCP. Furthermore, it can be implemented easily by modifying only transmission process of sender side. Owing to these merits, HSTCP is a promising protocol that can be adopted in next generation protocol.

However, HSTCP also has a serious unfairness problem to be resolved. There can be various causes, but RTT (Round Trip Time) directly influences on flow's performance. HSTCP is in more serious state rather than regular TCP during same unit time. This is because the increment scale size of HSTCP is larger than a fixed '1' of Reno-TCP. Hereupon, this paper analyzes how the difference of RTT influences on performance of flows, and shows that a just RTT difference can induce a great difference on throughputs between flows.

This paper also proposes a new mechanism which can embrace the difference of RTT. We call it F-HSTCP (Fair-HSTCP). F-HSTCP suggests a new increment value that can respond more reasonably to each ACK. The new increment value includes the delay property as well as current congestion window. This is for the resource to be distributed fairly between competing flows.

The remainder of this paper is as follows. Section 2 overviews representative High-BDP protocols and describes basis characteristics of HSTCP. Section 3 points out the problems of HSTCP and shows the two simulation results. Section 4 suggests F-HSTCP as a new mechanism to utilize more efficiently the BW of network. Section 5 evaluates the performance of F-HSTCP using NS-2. Finally, conclusions are provided in Section 6.

2 Related Research Protocols

A regular TCP has following some problems when it is used to transfer the data of high-capacity. It can sustain high speed rate of 10Gbps only when it guarantee more strict loss rate than theoretical BER, as 10^{-10}. If 1 segment is directly matched with 1 packet and there is a packet loss when current window size is w, it takes $(w/2) \cdot RTT$ to recover to the window level w that it was before loss. So, to solve these problems, high-BDP protocols like as HSTCP, STCP, FAST TCP, XCP are introduced.

HSTCP is a representative high-BDP protocol that is proposed to solve these problems. While regular TCP uses a constant factor of (+1) and (-1/2) with response to ACK and Loss, HSTCP uses a modified factor in proportion to own current window size. Hereby, HSTCP can keep the high speed of 10Gbps if only it maintains 10^{-7} of loss rate [2]. This is valid from following Eq.1

$$\mathrm{w} = \frac{0.12}{\mathrm{p}^{0.835}} \tag{1}$$

Eq.2 is the increment value to be added after receives ACK, and Eq.3 is the decrement value to be reduced after detects a loss [3]. As we know by above

equations, the response parameters of HSTCP are expressed as a function of window like as $a(w)$ and $b(w)$ while those of regular TCP are constant such as a=1, b=1/2 [4][6]. $a(w)$ and $b(w)$ are derived under assumptions that size of packet is 1500Byte and RTT is 100ms

$$\mathrm{a(w)} = \frac{\mathrm{w}^2 \cdot \mathrm{p(w)} \cdot 2 \cdot \mathrm{b(w)}}{2 - \mathrm{b(w)}} \tag{2}$$

$$\mathrm{b(w)} = (B - 0.5) \cdot \frac{\log \mathrm{w} - \log \mathrm{W}_0}{\log \mathrm{W}_1 - \log \mathrm{W}_0} + 0.5 \tag{3}$$

As we can see from Table.1, HSTCP increases more $a(w)$ and decreases less $b(w)$ according as current window becomes larger. Like this, HSTCP controls $a(w)$ and $b(w)$ non-linearly. This algorithm enables HSTCP to sustain the high speed such as 10Gbps. Also, it can shorten the time that taken to return to previous window level.

STCP uses congestion control of MIMD (Multiplicative Increase Multiplicative Decrease) instead of AIMD when cwnd is larger than 16. Then, STCP uses a fixed response parameter as a=0.01, b=0.125. And the other case, it operates in same way with regular TCP [8]. XCP provides very excellent efficiency and equity and stability because it notifies congestion degree explicitly with similar procedure to ER (Explicit Rate) algorithm of ABR (Available Bit Rate) in ATM net. However, there are some problems to be adopted in network because XCP should be applied to the sender side, receiver side, and router at the same time [9].

Table 1. HSTCP AIMD Values

w	a(w)	b(w)
38	1	0.50
118	2	0.44
221	3	0.41
347	4	0.38
495	5	0.37
663	6	0.35
851	7	0.34
1058	8	0.33
1284	9	0.32
1529	10	0.31
1793	11	0.30
2076	12	0.29
...	...	...
10661	30	0.21
11358	31	0.20
12082	32	0.20
...	...	...
84035	71	0.10
89053	72	0.10

Fast-TCP uses congestion control based on Vegas. This protocol uses the DCA (Delay based Congestion Avoidance) mechanism that analogizes buffer state of network with momentary change of RTT like as Vegas. DCA is a mechanism that estimates the current state of network by increasing or declining tendency

of RTT. But, the latest study evolves that a packet loss bears less relation to increasing trend of RTT, so gradual introduction of DCA is difficult [10].

These high-BDP protocols have common characteristic that improves throughput by sending more amount of data than regular TCP for unit time.

3 HSTCP Problems

The bandwidth-delay product (BDP) defines the amount of data a TCP connection should have in flight at any time. The windows value means the amount of transferred data during a round trip time delay and depends on the characteristic of BW and Delay. Here, transmission rate per unit time is referred to as throughput. Thus, throughput of a flow at time t is expressed as follows

$$X(t) = \frac{W}{RTT} : \ throughput\ at\ time\ t \tag{4}$$

Eq.4 means that it takes RTT time for sender to transmit W packets.

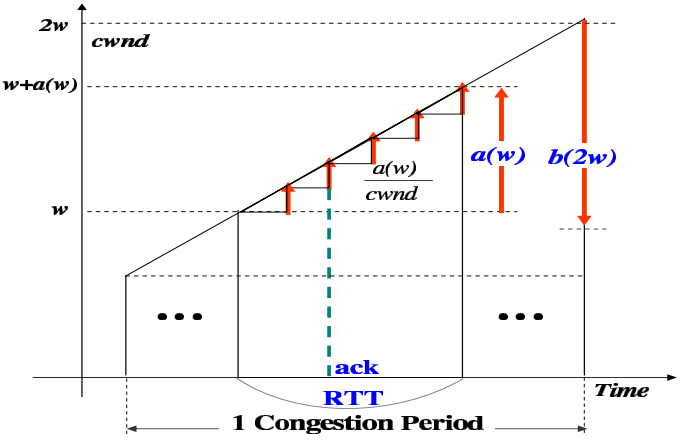


Fig. 1. Dynamic Characteristic of HSTCP

Fig.1 shows the relation between window size and HSTCP AIMD parameters. It adds $a(w)$ and reduces $b(w)$ according to current congestion window. It takes RTT time for the whole a(w) to be applied to congestion window size.

Section.3 points out 2 problems of HSTCP. First, it takes different time for HSTCP flows to reach to the maximum bandwidth of the link if the lengths of transmission link are not equal. It can be considered as link utilization problem because it can not utilize the bandwidth as much as the difference of required time. Second, HSTCP flows undergo the serious unfairness problem when they compete in bottleneck link if they have unequal propagation delay. Here, let's say

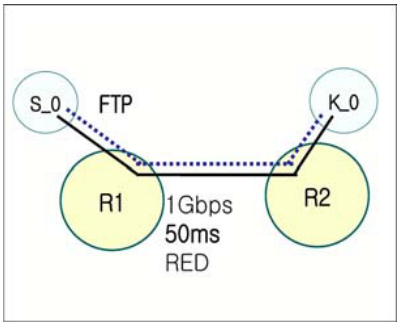


Fig. 2. One flow topology

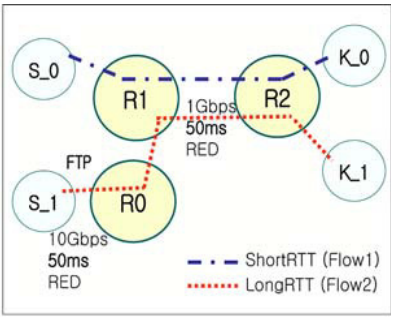


Fig. 3. Competing flows' topology

it is fair when competing flows share evenly the limited resource of network. Since a(w) is added up to current window size in RTT unit, short delayed flow naturally gets an opportunity to obtain more resource relatively than long delayed flow. This causes competing flows to be faced with unfairness problem. It is because they have the same $a(w)$ without consideration for any other conditions if only their window size are concurrent.

These two problems are caused by the property of RTT. In fact, $a(w)$ and $b(w)$ has been formed out of the assumption that the flow's packet size is 1500Byte, and delay is 100ms. From the modeling equations, they regard the delay of all flows as 100ms. So, the problems maybe natural since the delay property would not be reflected when it calculates increment size to be increased during RTT cycle. These problems happen exceedingly when its congestion window is large, because the data to be increased in unit time would be bigger. From Table.1, we can observe that $a(w)$ is gradually bigger as current window size grows larger [1].

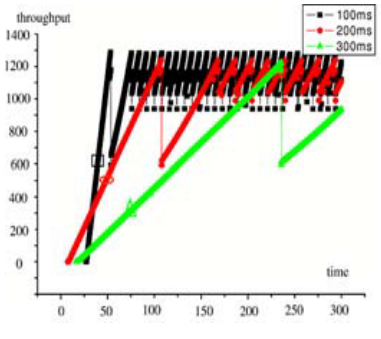


Fig. 4. Flow's throughput by RTT

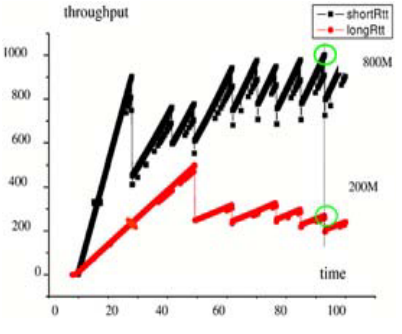


Fig. 5. 2 flows' throughput

Through simulation, this section will confirm 2 issues indicated as problems. For performance evaluation, NS-2(version 2.27) network simulator has been used, and 2 topologies are designed for each performance measure. The bandwidth of router R1-R2 is set to only 1Gbps, so, the link would be a bottleneck and causes losses in network. To avoid synchronous loss situation, router used RED with

threshold level 300 as queue management of bottleneck link. All flows used SACK option and transferred data through FTP. Receiver's window size is set with very large value as much 100,000 so that do not influence in performance.

Fig.2 is a topology to check that how much time is taken for a flow to reach to the maximum throughput of bottleneck link when different RTT is given to the flow. Fig.3 is a topology to observe the bandwidth ratio of two flows when they compete at same bottleneck link. To analyze easily, we used only 2 HSTCP flows. The unique difference between two flows is RTT which meaning the length of transmission path. The flow1 has short RTT (100ms) and flow2 has longer RTT (200ms), so the RTT ratio of two flows is 1:2. All the other conditions are established equally to the two flows. That is a simulation environment to check how two competing HSTCP flows are increasing their resources from the limited bandwidth of R1-R2.

Fig.4 is a simulation result on topology of Fig.2. We ran a flow 3 times changing its delay to be 100ms, 200ms and 300ms. This is for measurement about how much time is taken for the flow to get the maximum bandwidth of the link as it grows its delay. As we can see in Fig.4, when the flow is delayed with 300ms, it can not utilize the maximum bandwidth of the link until it comes to 225 seconds. This fact can be considered as resource wasting as much that time. Like this simulation, the delay of the flow has a critical influence on its throughput performance. The fact will be come out as a serious problem in case the delay is extremely large.

Fig.5 is a simulation result come from Fig.3. This is showing unfairness problem that 2 flows have been experienced. The condition is that they compete for the limited resource of R1-R2 bottleneck link. Run time is 100 seconds.

On the whole, the bandwidth sum of two flows maintains 1Gbps, bottleneck link's BW. Similarly, the bandwidth sum of two flows is also 1G at 92 seconds. But, their bandwidth occupation ratio is 4:1 even though their RTT ratio is 1:2. The flow2 would be a disadvantage up to 4 times in throughput utilization in spite of RTT difference of 2 times. If the execution time is extended as much 1,000 seconds, the bandwidth occupation ratio would be larger. The ratio of bandwidth occupied by each flow becomes larger than just two flows' RTT ratio. Finally, the flows may not converge to any reasonable level or it would take a great time even though they converge. This is an unfairness problem due to solely RTT difference.

4 Proposed F-HSTCP Method

This section introduces F-HSTCP that can distribute fairly the resource. While current HSTCP decides the increment $a(w)$ with only current window size, F-HSTCP decides the increment $a(w)$ to be reflected current window size as well as relevant RTT. HSTCP provided a remarkable mechanism to sustain a high speed such as 10Gbps. But, as we can see two simulation results, it undergoes serious unfairness problem when flows compete for the limited resource of bottleneck link, since they do not reflect the RTT property to increment parameter.

It can be supposed easily that the mentioned problems can not but appear. Because present HSTCP flows would add up the equal $a(w)$ during next RTT unit if they are same window size[7].

To overcome the weakness, F-HSTCP is designed for throughput increment rate to be equal within same interval time. If we differentiate Eq.4 at time t, we can get the equation standing for throughput's increment rate. Because RTT itself value is a constant, throughput's increment rate can be expressed like as following Eq.5.

$$\frac{dX(t)}{\mathrm{dt}} = \frac{d}{dt} \cdot \frac{W(t)}{RTT} = \frac{1}{RTT} \cdot \frac{dW(t)}{dAck} \cdot \frac{dAck}{dt}$$

$$= \frac{1}{RTT} \cdot a(w) \cdot \frac{1}{RTT} = \frac{a(w)}{RTT^2} \tag{5}$$

Eq.5 means the increment rate to be increased when HSTCP receives an ACK. Consequently, Eq.5 also explains that throughput's increment rate is inversely proportional in RTT square. Therefore, we can understand the problems in section.3. The present HSTCP refers only current window size itself when they calculate $a(w)$ without considering RTT. Therefore, F-HSTCP may need a new parameter which reflecting the property of delay. Like as $a(w)$ is the variable for 100ms RTT flow, F-HSTCP need a new parameter which can include delay property of 200ms, 300ms and so on. It is $a_1(w)$ that the flow can increase during its next RTT 100ms RTT, because the delay is supposed to be 100ms. Similarly, it is $a_2(w)$ that should be applied to the flow which has 200ms delay. If its delay is 300ms, then $a_3(w)$ will be needed.

Then, let's match $a_1(w)$ for the flow1, $a_2(w)$ for the flow2 and $a_3(w)$ for the flow3. When they receive the ACK, they will increase its window as much $a_1(w)$/cwnd, $a_2(w)$/cwnd and $a_3(w)$/cwnd. Since the increment rate of throughput should be equal among them, there should be some relation among $a_1(w)$/cwnd, $a_2(w)$/cwnd and $a_3(w)$/cwnd. If there is any equation that satisfies the relationship among them, we can expect that resource will be distributed fairly to the all flows. To distinguish from existent a(w), let's define x(w) as the increment size to be increased during next RTT unit. x(w) is a new concept variable which is reflecting the current congestion window size as well as the property of delay.

Now let's get the x(w). To derive formula simply, let's suppose only two HSTCP flows which are competing at the bottleneck link. They are classified with same number. Namely, the window1 of flow1 has the characteristics of bw1 and RTT_1, and window2 of flow2 would be bw2 and RTT_2. And, let's assume there is a following relation between two flows' RTT.

$$RTT_2 = \alpha_{2|1} \cdot RTT_1 \tag{6}$$

The $\alpha_{2|1}$ means the RTT ratio of two flows. That esxpresses how much RTT2 is different from RTT1. Let's denote $X_1(t)$ and $X_2(t)$ to express each flow's transmission rate. For competing flows to acquire the resource fairly, the throughput's increment rate should be same in some time. Eq.7 shows that condition [5].

$$\frac{dX_1(t)}{dt} = \frac{dX_2(t)}{dt} \tag{7}$$

$$\frac{a_1(w)}{RTT_1^2} = \frac{a_2(w)}{RTT_2^2} \tag{8}$$

$a1(w)$ is a window scale size of flow1 and $a2(w)$ is one of flow2. After substituting RTT1 for RTT2, we can get the following Eq.9. That shows the correlation of two flows' increment scale size.

$$a_2(w) = \alpha_{2|1}^2 \cdot a_1(w) \tag{9}$$

Through Eq.9, flow2 can get its own increment rate by multiplying the square of $\alpha_{2|1}$ and $a_1(w)$. Then, throughput's increment rate of flow2 would be same as flow1.

One flow can get its own increment rate if it knows the RTT property of accompanied flow as Eq.9. To generalize this mechanism, a criterion that all flows should refer to is needed. When $a(w)$ and $b(w)$ is derived for HSTCP to sustain 10Gbps transmission rate, the flow has been 1500byte of packet size, 100ms of RTT. So, F-HSTCP uses 100ms as a reference RTT. That does not influence on HSTCP throughput because F-HSTCP regards HSTCP as a criterion.

$\alpha_{2|1}$ can be replaced with α, because that means how each flow's RTT is different from 100ms. Therefore, F-HSTCP flows increase $a(w) \cdot \alpha^2$ during RTT unit while HSTCP adds just $a(w)$. Consequently, HSTCP adds $\frac{a(w)}{\text{cwnd}}$ whenever it receives ACK, but F-HSTCP does $\frac{a^2 \cdot a(w)}{\text{cwnd}}$.

The modification of NS-2 code for F-HSTCP is expressed in Fig.6.

This paper has explained $x(w)$ as the new increment size to be applied in RTT unit. However, $b(w)$ that decrease current window to be adjusted to the

```
recv_Ack
switch(wind_option) {
                ⋮                    /* windowOption_1 : RenoTCP */
    case windowOption_8 :
        if( cwnd_ ≥ 38) {
            calc a(w);               /* general HSTCP a(w) */
            α <- rtt_ / 100 ;
            x(w) = a(w) · α²;        /* new x(w) for F-HSTCP */
        }
}
answer = x(w) / cwnd_;
return answer;
```

Fig. 6. Code modification for F-HSTCP

network state should be same with standard HSTCP. This is because $b(w)$ is already included in $a(w)$ parameter. So, $b(w)$ value should be one of HSTCP.

5 F-HSTCP Performance Evaluation

In this section, we show that issues mentioned as problems could be resolved in case of use F-HSTCP introduced in section.4. We also used the same environment that used in section.3. The code of Fig.6 has been applied to the real NS-2 simulator code of "ns-allinone-2.27/ns-2.27/tcp/tcp.cc".

Fig.7 is a result when the topology of Fig.2 is executed under F-HSTCP mechanism. The flows' RTT are 100ms, 200ms and 300ms. As mentioned before, the flow does not compete with other flows, but solely use the total bandwidth of 1Gbps in network.If it is compared with Fig.4, we can understand at once that F-HSTCP has superiority in utilization of resource than HSTCP. In Fig.7, The time taken for each flow to utilize the maximum bandwidth of network is almost equal even though their RTT ratio is 1:2:3. The bandwidth that was wasted in HSTCP is utilized in F-HSTCP. Fig.8 shows the bandwidth ratio of flows when 2 flows compete on the bottleneck link. The topology is Fig.3. If it is compared with Fig.5, we can see a considerable improvement of fairness. In Fig.5 and Fig.8, flow1 has same throughput level at around 20 seconds. This means there is no difference between HSTCP and F-HSCTP when flow's RTT is 100ms. And the throughputs of 2 flows are diverging at the around 40 and 75 seconds due to limit of BW. But, they attempt to converge to some reasonable level in some time.

In this way, F-HSTCP reflects each flow's RTT characteristic in throughput's increment rate whenever it receives ACK, and so it can utilize the resource more efficiently than existing HSTCP. Also as we can see from the simulation results, it can alleviate considerably the unfairness problem.

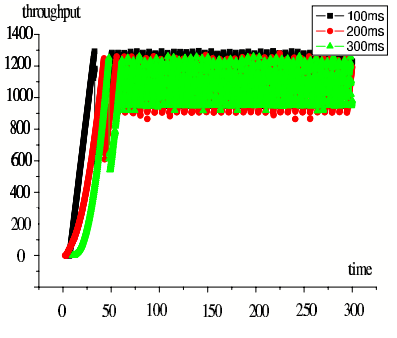


Fig. 7. Individual HSTCP flow throughput

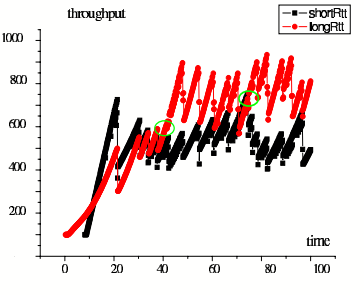


Fig. 8. Competing HSTCP flows' throughput

6 Conclusion

HSTCP enabled a flow to maintain a high speed such as 10Gbps by modifying basic AIMD. However, HSTCP has been faced with a serious unfairness problem

because it calculates the increment scale size with only current window size. Namely, this means that HSTCP regards RTT of all flows as 100ms. By this reason, it may be natural that HSTCP flows experience the unfairness problem. In this paper, the problems are shown in Fig.4 and Fig.5.

Therefore, this paper proposed F-HSTCP as a new protocol that reflects the property of RTT as well as current window size in AIMD parameters. Instead of a uniform $a(w)$, F-HSTCP adopts $a(w) \cdot \alpha^2$ as the increment value to be added up during RTT units. The newly introduced $a(w) \cdot \alpha^2$ enables F-HSTCP to embrace the difference of RTT for itself. This paper also showed that F-HSTCP is able to resolve the unfairness problem through modified mechanism. Fig.7 and Fig.8 are simulation results which are showing those improvements.

All F-HSTCP flows have the same throughput's increment rate. In case of HSTCP, the increment value to be increased during its own RTT period is designed to be equal. But, F-HSTCP has the throughput's increment rate to be equal whenever it receives ACK. From these characteristics, F-HSTCP is able to utilize more efficiently the resource of network irrespective of its own RTT. Also, F-HSTCP can distribute fairly the limited resource of network to the competing flows. And, the operating time is extended as much as the portion added in "ns-allinone-2.27/ns-2.27/tcp/tcp.cc", but it may be negligible because just 2 lines are inserted. Even more, the calculation is executed with existent variables.

References

1. Damien Phillips and Jiankun Fu, "Analytic Models for Highspeed TCP Fairness Analysis", Networks 2004 (ICOIN2004), vol2, pp.725-730, Nov. 2004
2. Sally Floyd, "HighSpeed TCP for Large Congestion Windows", IETF RFC 3649, Dec. 2003
3. Sally Floyd, Sylvia Ratnasamy, and Scott Shenker, "Modifying TCP's Congestion Control for High Speeds", Internet draft, http:// www.icir.org/floyd/notes.html, May. 2002
4. Allman, M. Paxson, V. And W. Stevens, "TCP Congestion Control", IETF RFC 2581, Apr. 1999
5. Eitan Altman, Chadi Barakat and Emmanuel Laborde, "Fairness Analysis of TCP/IP", Decision and Control 2000, vol 1. Pp.61-66, Dec. 2000
6. Richard Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm", IETF RFC 2001, Jan. 1997
7. Evandro de Souza, Deb Agarwal, "A HighSpeed TCP Study: Characteristics and Deployment Issues", LBNL Technical Report LBNL-53215, May. 2003
8. T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks", ACM Computer Communication Review, vol.33 no.2, pp.83-91, Apr. 2003
9. D. Katabi, Mark Handley, and Charles Rohrs, "Internet Congestion Control for High Bandwidth-Delay Product Networks", In Proceedings of the ACM Sigcomm, Aug. 2002
10. C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance", In Proceedings of the IEEE Infocomm, Mar. 2004

Application-Rate Aware Congestion Control Algorithm for Video Streams

Jinyao Yan, Qin Zhang, and Jianzeng Li

Communication University of China, Beijing, P.R. China, 100024
{jyan, zhangqin, jzli}@cuc.edu.cn

Abstract. In this paper, we propose an algorithm called Application-rate Aware and TCP friendly Rate Control Algorithm (AATFRC) for video streams. AATFRC algorithm takes the encoded rate of the video streams into account, while being TCP friendly in the long-term. Experimental results show that our proposed AATFRC algorithm is superior to TCP-friendly Rate Control Algorithm (TFRC) in terms of a) little delay to reach the minimum rate of the application during the slow start; b) the assurance of minimum application quality during the short-term congestions; c) the variation of the sending rate is not bound to the TCP response function but consistent with the encoded application rate.

1 Introduction

IP network has the heterogeneous nature, in particular a) variability of path capacity between server and client for the various clients b) heterogeneity of device capabilities and c) the single best-effort service class offered in Internet. Therefore, multimedia communication over IP network is facing many challenges. To deal with these diverse and changing network conditions for the multimedia communications, congestion control algorithms for streaming applications are used to adapt the sending rate that the current available network resources are neither overloaded nor underutilized. TCP, the dominant congestion protocol in the Internet, uses an Additive-Increase Multiplicative-Decrease (AIMD) mechanism to detect additional available bandwidth and to react to congestions. TCP congestion control is suitable and efficient for bulk data transfers. However, it is not well suited for the growing number of audio/video streaming applications. Without congestion control, non-TCP traffic can cause starvation or even congestion collapse to TCP traffic, if both types of traffic compete for resources at a congested FIFO queue [1]. Therefore, new TCP-compatible features have been integrated into congestion control algorithms ([2],[3],[9]) for audio/video-streaming applications in order to handle competing dominant TCP flows in a fair manner. These features have two important characteristics in common: (i) slow responsiveness to smooth data throughput; and (ii) "TCP friendliness". By definition [1], a flow is called TCP-friendly, if and only if it uses in the long term no more bandwidth than a conforming TCP flow would use under comparable conditions in the steady state.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 690–700, 2006.

An example of the class of TCP friendly congestion control algorithms is the TCP-Friendly Rate Control algorithm (TFRC) [2]. TFRC is an equation-based congestion control mechanism that uses the TCP throughput function presented in [4] to calculate the actual available rate for a media stream. Previous work ([5], [6]) on TCP-friendly congestion control protocols showed that TFRC offers better performance than other TCP-friendly congestion control algorithms. Even though TFRC tries to smooth the rate variability, TFRC ensures only the friendliness or the compatibility to the existing TCP protocols without fully consideration of the friendliness to applications ([10],[11]). In [10], authors showed the impact of TCP-friendliness to application friendliness and tried to develop a joint media- and TCP-friendly congestion control algorithm. We summarize the deficiencies of TFRC concerning application friendliness as follows:

Firstly, in its initial phase, TFRC uses the slow start method to guarantee that currently available network resources are not overloaded. It gradually increases its rate starting from 1 packet/second until available path capacity is reached, thus it introduces delay to the live media applications like videoconference.

Secondly, in the congestion avoidance phase of TFRC, even if media flows are not the cause of congestions, they could still suffer from short-term congestion incidents [8]. Subsequently, the quality of rendered streams is worsened. More over, even there are short-term congestions, media applications must be transported higher than the minimum rate (for example, the encoded rate of the base layer of scalable video streams). Otherwise, the quality of application would be harmed dramatically [14].

Thirdly, the TFRC's sending rate is not consistent with the rate application encoded. TFRC employs the TCP response function with the average loss event rate so that it increases sending rate by at most 0.14 packets/RTT to smooth the sending rate [2]. However, the variation of the encoded bit-rate can be very high when the encode mode changes between intra-frame-encode and inter-frame-encode modes or when stream increases the encode layers for scalable video streams. On the other hand, live media applications have some natural limits on transmission rates, it cannot be transmitted faster than the high rate that media was encoded (for example, the encoded rate of the base layer and all enhancement layers of scalable video streams), and thus media applications are not aggressive as bulked data applications.

Therefore, the challenge is *to develop a congestion control algorithm that is TCP-friendly and "application-rate aware" at the same time*. We propose an algorithm called Application-rate Aware and TCP friendly Rate Control Algorithm (AATFRC) in this paper.

The contributions of our proposed algorithm in this paper are a) little delay to reach the minimum rate of application during the slow start; b) the assurance of minimum application quality during the short-term congestions; c) the variation of the sending rate is consistent with the rate application encoded in short term, but it is not bound to the TCP response function.

The rest of this paper is organized as follows. Firstly we briefly take look at the TFRC algorithm and the overlaid media steam traces in section 2. Then, we present our proposed AATFRC algorithm in section 3. Section 4 reports the simulation results and analysis. We conclude the paper in section 5.

2 TCP Friendly Rate Control Algorithm and Media Streaming

2.1 TCP Friendly Rate Control Algorithm

The TCP-Friendly Rate Control algorithm is a representative example of the TCP compatible congestion control algorithms. TFRC algorithm has two phases namely the first phase-slow start and the second phase-congestion avoidance. In the slow start phase, TFRC increases the rate starting from 1 packet/second in order to enter the network cautiously. In congestion avoidance phase, TFRC uses the TCP throughput function (1) presented in [4] to calculate the actual available rate for a media stream.

$$R(p) = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}}) * p(1 + 32p^2)} \tag{1}$$

$R(p)$, the throughput of TFRC or TCP, depends mainly on the round-trip time t_{RTT}, the retransmission timeout value t_{RTO}, the segment size s, and the packet loss rate p. Detailed design of TFRC can be found in [2].

A major advantage of TFRC is that it has a smooth sending rate for media streams, while maintaining roughly the same throughput as that of a conforming TCP flow during the congestion avoidance phase under comparable conditions.

2.2 Rate Variation of Media Streaming

Fig.1 is an example of encoded video rate trace. We plot the encoded length of each frame from frame 15000 to frame 17000 of From Dusk Till Dawn [15]. The video coding approach used is MPEG-4 with 25 frames per second.

As shown in Fig.1, the rate variability of the video stream is small in a large timescale, namely in the average of one second. However, the encoded bit rate shows significant rate variability from frame to frame, namely in the timescale of 40ms (25 frames per second). Clearly this is due to that the intra-frame-encoded frames are encoded with much more bits than that of the inter-frame-encoded frames.

TFRC algorithm smoothes the data rate as the only goal for media streams. However, the smooth sending rate does not match the encoded rate of video streams in a small timescale. Thus, this deficiency, the third deficiency of TFRC mentioned in section 1, worsens the quality of rendered streams.

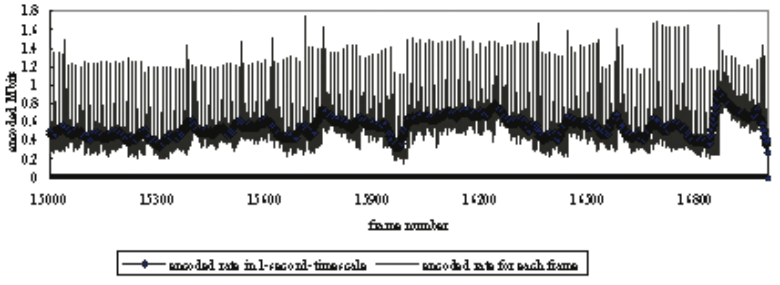


Fig. 1. encoded length of each frame from frame 15000 to frame 17000 of From Dusk Till Dawn

3 Application-Rate Aware and TCP Friendly Rate Control Algorithm

3.1 State Transition Diagram of AATFRC Algorithm

As we discussed in the first section, TCP friendly rate control algorithm has three deficiencies in terms of media friendliness. We present an Application-rate Aware and TCP friendly Rate Control Algorithm (AATFRC) in this section. Without loss of generality we can assume the media stream is encoded in two bit-rates: the low bit rate when only the base layer of the stream is encoded, and the high bit rate when both the base layer and enhancement layer of the stream are encoded. Following is the state transition diagram of AATFRC algorithm:

R_L: average Low-sending Rate (the sending rate in the R_L state is equal to the encoded rate of the base layer)

R_H:average High-sending Rate (the sending rate in the R_H state is equal to the encoded rate of the base layer plus enhancement layer)

AA: Application Minimum rate Aware state.

R: the available ending rate estimated with equation (1).

3.2 Description of AATFRC Algorithm

As shown in Fig.2, after the connection start, AATFRC goes to the Slow Start phase. In the Slow Start phase, the sending rate increases up to the value of the low rate of video stream immediately. Receivers can present the video streams timely with a basic application quality as the arriving rate of the video streams increases up to the low rate of video streams without any delay. If the receiver reports no packet loss event, the sender increases the sending rate up to the high rate of video streams. It is not necessary and impossible to send the bits faster than the high rate at which live streams is being encoded. Note that we assume there are no sender buffers and only small receiver buffers since buffers would introduce delays to the real-time streaming system.

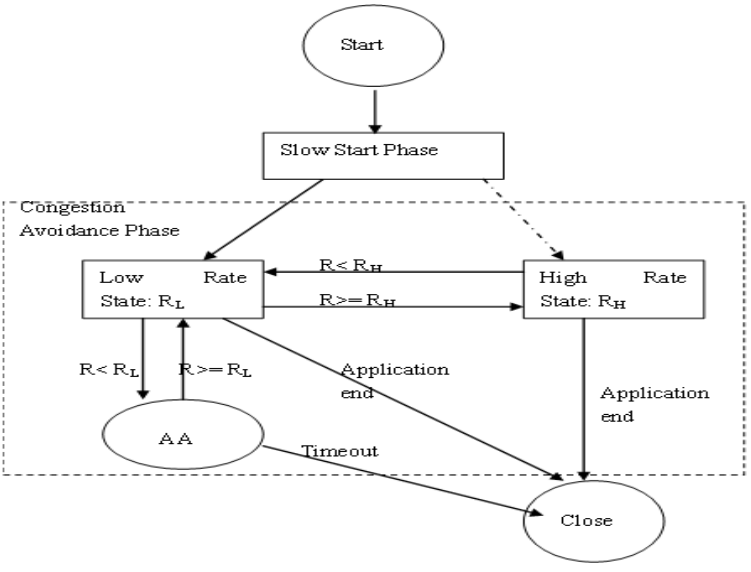


Fig. 2. State transition diagram of AATFRC algorithm

Connections enter the congestion avoidance phase from the slow start phase when a packet loss event is reported. In this phase, our algorithm uses the TCP throughput function (1) to calculate the actual available rate R for the media stream so that our algorithm is TCP friendly to the competing TCP streams. If the estimated available throughput is higher than the high rate of video streams for a certain period of time, the sender adjusts the sending rate to the high rate that video is encoded. If the estimated available throughput is lower than the high rate of video streams for a certain time, the sender adjusts the sending rate to the low rate that video is encoded. To avoid the oscillation of the sending rate, we set a timer here, which is larger than the timescale of short congestions, to trigger the adjustment of the sending rate.

In the congestion avoidance phase, we propose a state called AA (Application Minimum Rate Aware state) in AATFRC as shown in Fig.2. When the estimated available throughput is lower than the low rate of video streams, connections enter the AA state, and another timer is triggered. Connections come back to low rate state only if the estimated available throughput is high than the low rate and the timer is not expired. Connections would be closed down when the timer is expired, since we consider the network enters into a long-term and severer congestion state and application quality would be damaged if the available sending rate R were lower than R_L.

The method of calculating the end-to-end loss rate in AATFRC is the same Weighted Average Loss Interval (WALI) method in TFRC.

In practice, neither the encoded low rate nor the encoded high rate of video streams is strictly constant. In our AATFRC, the low rate R_L is the average of encoded rate of base layer while the high rate R_H is the average of encoded

rate of base layer and enhancement layers. The sending rate of AATFRC is exactly the rate of video streams is being encoded during R_L, R_H and AA states. The available sending rate R estimated with TCP response function is the criterion for transitions between the states. Thus, AATFRC algorithm takes the application rate into account and assures the sending rate matches the encoded rate while maintaining the TCP friendliness in long-term.

4 Simulations and Results Analysis

4.1 Simulation Configurations

We use ns-2 [7] for our simulation and performance analysis of AATFRC. In the simulations, we use the "dumbbell" topology, which has a single bottleneck link whose capacity is 2Mbps and link delay is 60ms as shown in Fig.3. All other links have the capacity of 10 Mbps and the link delay is 10ms (representing access links or LAN connections).

A video-streaming server (node 2) and a video-streaming client (node 5) are added as shown in Fig.3. Nodes 3 and 6 are added as the TCP sender and receiver. To simulate background traffic, a flow is generated between nodes 4 and 7 by superposing 100 ON/OFF sources with the rate of 10kbps that have Pareto distribution [12] during the entire simulation time in experiment 2, whereas from 50s to 100s in experiment 1.

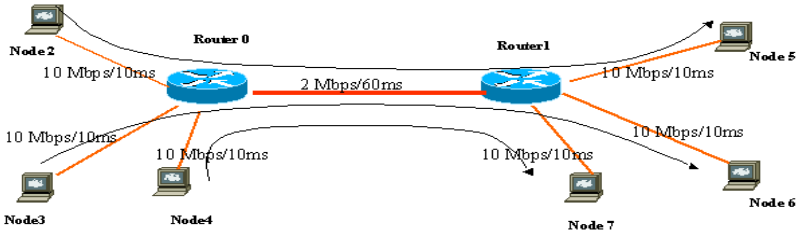


Fig. 3. Simulation topology

4.2 Simulation Results of AATFRC Algorithm

In this section, we present experimental results obtained from some simulations we conducted to evaluate AATFRC algorithm. Subsequently, we conduct experiments on TFRC using the identical simulation setting and compare the observed bit rates traces against those achieved by AATFRC.

We define three metrics to compare the properties of algorithms under study:

1) Slow start delay: the period of time between the start of the connection and the transmission rate reach to the encoded bit rate of video streams.

2) Responsiveness to short-term congestions: how the algorithm adjusts the sending rate and how the media quality is affected if network congestion occurs.

3) Difference between the sending-rate and application rate: the larger average difference between the sending rate and application-encoded rate, the more delay and jitter introduced to playback at the client side.

Experiment 1: Simulation results with scalable video streams (modeled as two-CBR sources)

Given the encoded layers of video streams, the encoded rate is usually steady in large timescales. To examine the properties of the algorithms, we model the overlying video streams as CBR sources with 2 kinds of rates i.e. a low rate and a high rate. Specifically, we use a low rate with 400kbps and a high rate with 800kbps in our simulations presented in this paper. Indeed, we have also conducted simulations with different low-rates and high-rates of video streams and the results are similar.

The bit rate traces obtained during our simulation are shown in Fig.4 and Fig.5.

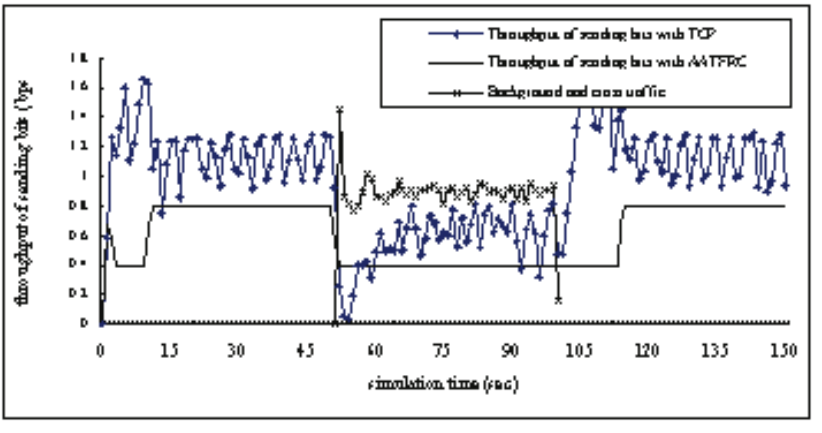


Fig. 4. Sending rate traces with AATFRC

The first observation is that the slow start delay of AATFRC and TFRC are respectively less than 1 second and about 10 second. TFRC stream enters the network so cautiously that it introduces longer delay and hurts the live communication. AATFRC stream enters the network with the application low rate to meet the application friendliness, and then AATFRC increases the sending rate or cuts the connection depending on the congestion conditions.

Please note, if the network goes to severe congestion or has not enough bandwidth capacity at all, the AATFRC connections will be cut in a certain length of time depending on the timers set in the algorithm. In this case, the connections go through states: Start->Slow Start->Low Rate->AA->Close. We omit the experiments conducted for this case.

The second observation is that AATFRC is less responsive to short-term congestions, thus keeps the streaming application robust. In this experiment, the

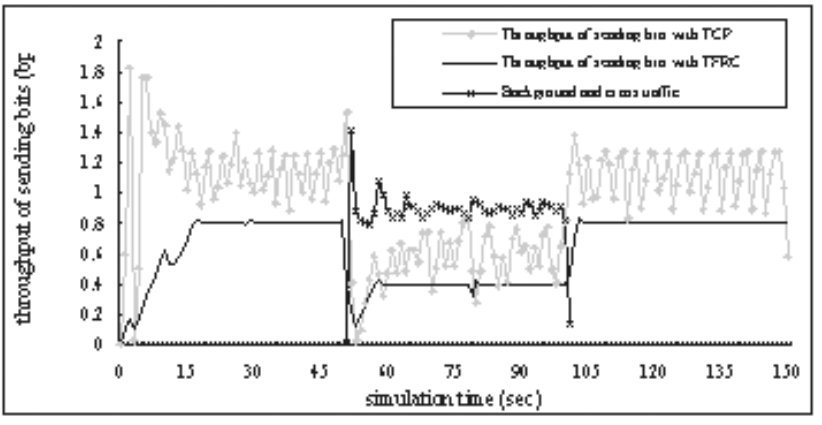


Fig. 5. Sending rate traces with TFRC

background traffic flow is generated between nodes 4 and 7 during the simulation time from 50s to 100s.As shown in Fig.4, AATFRC sends the stream with the low rate after the 50th second when a short-term congestion happens. Whereas, TFRC decreases the sending rate less than the low rate after the 50th second when a short-term congestion happens, so the streaming quality is harmed dramatically and the displaying of the stream at receiver side is stuck. The similar case happens around 75th -80th second where a short-term congestion happens.

The third observation is that both AATFRC algorithm and TFRC algorithm are TCP friendly in long term since they use in the long term no more bandwidth than the conforming TCP flows use under comparable conditions. During the short-term and severe congestions, AATFRC might use more bandwidth than that of conforming TCP flows to assure the application quality. And it is also clear that media streaming applications have self-limited sending rates, therefore, streaming applications are not the causes of short-term congestions.

Experiment 2: Simulation results with video traces

We conduct experiments on AATFRC with single layer video traces to exam the difference between the sending-rate and application rate. The application rate trace generated by the video-streaming server (node 2) is the encoded trace from frame 15000 to frame 17000 of From Dusk Till Dawn as shown in Fig.1. In this experiment, some background traffics are generated between nodes 4 and 7 during the entire simulation time. We get the sending rate of TFRC and AATFRC in 1-second-timescale in Fig.6, while we exam the difference of sending and encoded video rate in 40-milliSecond-timescale in Fig.7. We omit presenting the rate trace of TCP connection.

From Fig.6 and Fig.7, we learn the difference between the sending rate of TFRC and encoded video rate is much larger than that of AATFRC, especially in the small timescale cases. The larger difference stands for the more delay and jitter for the streaming and displaying.

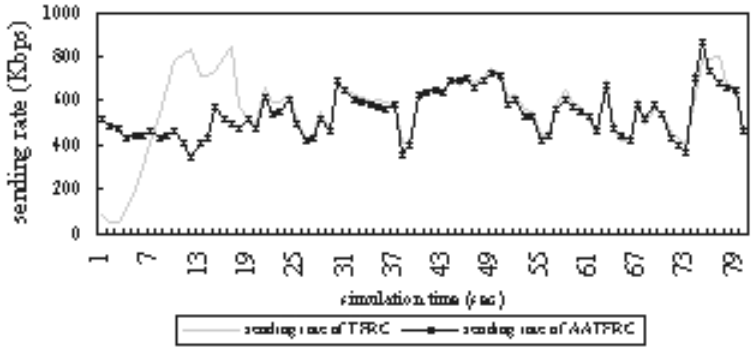


Fig. 6. Sending rate trace of TFRC and AATFRC in 1-second-timescale

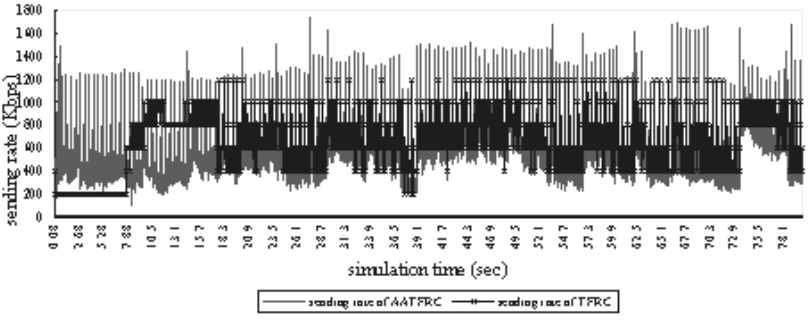


Fig. 7. Sending rate trace of TFRC and AATFRC in 40-milliSecond-timescale

Apparently, again in this experiment, AATFRC algorithm introduces little startup latency while TFRC algorithm introduces about 10 seconds startup latency due to its slow-start.

4.3 Discussion and Future Work

Where to be implemented. The AATFRC can be implemented into UDP-based real time transport protocol RTP [16] with interactions with streaming application since AATFRC needs to know the bit-rate at which application is encoded. Alternatively, the AATFRC can be implemented directly by the media streaming applications while using UDP as its transport protocol.

The setting of the timer. In [13] authors investigated the stationarity of the Internet. Experimental results show that packet loss rate is stable on time scales of a few seconds to minutes. Since the throughput of a transport protocol is mainly determined by delay and loss rate, we can expect that the stationarity of the throughput is maintained on time scales of a few seconds to minutes.

Therefore, we consider congestions less than a few seconds as short-term congestions. In our algorithm for simulation we set a short timer (e.g. $10*RTT$) when the connection come to the AA state for the first time to avoid overloading the network since there is little capacity information of the connection path at this moment, and then, we set a long timer (e.g. $100*RTT$) after that. If the timer in the state AA were very long, the connection can lead to severer and long-term congestions; if the timer were very short, the media stream would not be robust enough to short-term congestions. We will set the timer more adaptively in our future work.

5 Conclusion

In this paper, we propose the application-rate aware and TCP friendly rate control algorithm (AATFRC) to assure the streaming application quality, while being TCP friendly in the long-term. Experimental results show that our AATFRC algorithm is superior to TCP-friendly rate control algorithm (TFRC) in terms of a) little delay to reach the minimum rate of application during the slow start; b) the assurance of minimum application quality during the short-term congestions; c) the variation of the sending rate is consistent with the rate application encoded in small timescales, but it is not bound to the TCP response function.

We will continue our work on the extensive simulation and conduct real video streaming with AATFRC algorithm on the Internet as our future work.

Acknowledgements

This work is supported in part by the NSFC under Grant 60502015 and 60432030. The first author would like to acknowledge Prof. Bernhard Plattner, Dr. Martin May and Kostas Katrinis for their collaboration on the previous work at ETH Zurich, Switzerland.

References

1. Sally Floyd, Kevin Fall, "Promoting the Use of End-to-End Congestion Control in the Internet ", IEEE/ACM Transactions on Networking,1999.
2. S. Floyd, M. Handley, and J. Padhye, "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM, September 2000.
3. D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," Proc. of IEEE INFOCOM '01, pp. 631-640, 2001.
4. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation". SIGCOMM Symposium on Communications Architectures and Protocols, Aug. 1998.
5. D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," Proc. of ACM SIGCOMM '01, Aug. 2001.

6. R. Yang, M. S. Kim, and S. S. Lam, "Transient behaviors of TCP-friendly congestion control protocols," Proc. of IEEE INFOCOM'01, pp. 1716-1725, 2001.
7. ns-2 network simulator, http://www.isi.edu/nsnam/ns/
8. Zhiheng Wang, Sujta Banerjee, Sugih Jamin, "Media-friendliness of a slowly-responsive congestion control protocol" NOSSDAV'03,June 2004, Cork, Ireland
9. Jinyao Yan, Martin May, Kostas Katrinis, Bernhard Plattner, "A New TCP Friendly Rate Control Algorithm for Scalable Video Streams" in Lecture Notes in Computer Science 3462, IFIP Networking 2005, Waterloo, Canada, May 2-6, 2005.
10. Jinyao Yan, Kostas Katrinis, Martin May, Bernhard Plattner, "Media- and TCP-Friendly Congestion Control for Scalable Video Streams" IEEE trans. on Multimedia,Vol.8,NO.2,Apr.2006.
11. T. Phelan, "Datagram Congestion Control Protocol (DCCP) User Guide :draft-ietf-dccp-user-guide-03.txt", Internet Draft , April 2005
12. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Selfsimilarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level," Proceedings of ACM SIGCOMM '95, Cambridge,MA, Aug. 1995.
13. Y. Zhang, V. Paxson, and S. Shenker, "The stationarity of internet path properties: Routing, loss, and throughput," ACIRI Technical report, May 2000.
14. W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Trans.on Circuits and Systems for Video Technology, Vol.11, No.3, March 2001
15. http://trace.eas.asu.edu/TRACE/ltvt.html
16. RFC 1889 - RTP: A Transport Protocol for Real-Time Applications, January 1996. (Updated in RFC 3550, July 2003).

Network Security

An Efficient Key Tree Management Algorithm for LKH Group Key Management

Deuk-Whee Kwak[1], SeungJoo Lee[2], JongWon Kim[2], and Eunjin Jung[3]

[1] Backbone Network Research Team, Telecommunication Network Laboratory, KT, Daejeon, Korea
dhkwak@kt.co.kr
[2] Networked Media Lab., Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
{sjlee, jongwon}@gist.ac.kr
[3] Department of Computer Sciences, The University of Texas at Austin, Texas, USA
ejung@cs.utexas.edu

Abstract. The efficiency and security of secure group communication are dependent on the group key manager (GKM), which manages the group keys and membership. Although a GKM can employ any kind of group key management algorithm, we adopt the logical key hierarchy (LKH) in this paper for its efficiency and scalability. LKH is a tree-based group key management algorithm and it is more efficient when the key tree is balanced. However, only a few papers or documents have dealt with practical tree balancing techniques so far. In this paper, we propose *LKHTreeManager*, an LKH key tree management algorithm that is efficient for a large and highly dynamic secure group. The proposed technique efficiently manages the key tree by combining LKH and AVL (Adelson-Velskii and Landis) tree. We show that *LKHTreeManager* reduces the membership processing time, as well as the number of key messages.

1 Introduction

In a secure group communication that provides confidentiality and integrity, only eligible members are allowed to send or read the data that are encrypted and/or hashed with shared group keys [1]. The group keys and membership are managed by the group key manager (GKM). The GKM receives the join or leave request, processes the request, generates new group keys, and distributes them to group members. Therefore, the efficiency and security of the secure group communication heavily depend on the efficiency and security of the GKM's group key and membership management techniques.

Group key management algorithms specify how the group key should be managed [1, 2]. Although the GKM can employ any kind of group key management algorithm, we adopt the logical key hierarchy (LKH) in this paper for its efficiency and scalability.

LKH [2, 3], as one of the RFCs of IETF, manages the group key with a hierarchical structure called the key tree. A key tree is a data structure where

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 703–712, 2006.

a leaf node represents an individual key of each member and the root node is the group key. Each member stores a set of keys that are on the path from the member's individual key to the group key. When the group key needs to be changed, the cost of updating the group keys is proportional to the height of the node[1] in the key tree to be added or deleted[2]. Therefore, to reduce the group key management cost we need to reduce the average height of a node to be added or deleted. A height balanced tree has a minimum tree height on the average. Therefore, a group key management algorithm should address the issue of key tree balancing. However, only a few papers or documents have deals with practical key tree-balancing techniques.

In [2], the authors discuss how to manage the group keys for a multicast group that employs LKH, in the balanced tree, but they do not discuss how to keep the LKH balanced. In [3], the authors discuss the techniques for establishing and maintaining a binary tree based LKH and propose the use of a hash function for finding member location, and the distance[3] value for moving the highest node to another location. However, the proposal does not provide any specific algorithm or experimental results but only enumerates several balancing rules. In [4], the authors propose the use of a normal search tree or B^+ search tree for constructing and managing the LKH and advocate that the latter is more efficient than the former. The authors who propose the use of a key tree for group key management in [5] also propose a group key management framework, called *Keystone* in [6], to make use of the proposed group key management techniques. However, they do not specify how to keep the tree balanced while managing the group membership. The LKH tree-balancing technique in [7] is a decentralized and fault-tolerant algorithm, and thus distributes the load of the key server, but it is less efficient than a centralized algorithm.

In this paper, we propose a practical and efficient LKH tree management algorithm that reduces the membership-processing time and the number of key messages. The remaining sections of this paper are organized as follows: In Section 2, we introduce the related works on key tree management algorithms. An efficient algorithm for LKH group key management is proposed in Section 3, and in Section 4, we compare the proposed algorithm with a simple key tree algorithm in terms of membership-processing time, and with a B^+-tree-based LKH management algorithm in terms of the number of key messages. Lastly, we wrap up this paper with Section 5.

2 Related Work

2.1 Search Trees for Group Membership Management

In order to efficiently manage membership in a large group, a systematic management technique and information structure that is efficient in all the cases

[1] The terms member and node are used interchangeably according to the context.
[2] The terms 'add and join' and 'delete and leave' are used interchangeably according to the context, respectively.
[3] Path length from the root to the node.

of node insertion, retrieval, and deletion are required. For this purpose, we adopted the AVL (Adelson-Velskii and Landis) search tree [8]. An AVL is a binary tree that is either empty or consists of two AVL sub-trees, T_L and T_R, whose heights differ by no more than one. The worst case time complexity for the addition/retrieval/deletion of a node from a tree consisting of N nodes is $O(\log_2 N)$ and storage usage is $O(\log N)$. An AVL is not only a balanced but also a sorted tree in that the values of the left side nodes are always smaller than the values of those on the right side. However, the tree is efficient when it is loaded in memory as much as possible. Because of this, as well as the fast search time and memory efficiency, it is commonly used for small-sized search trees.

2.2 LKH Group Key Management Techniques

LKH is a group key management algorithm that is scalable since the number of messages for rekeying is $O(\log N)$ when the key tree structure is balanced [2, 5]. It is a rooted tree where each leaf node corresponds to each user. The leaf nodes of the LKH are secret keys between the GKM and each group member, and all the internal nodes except the root are the key encryption keys (KEKs) that should be shared among the sub-group members. The roor node is the group key. Therefore, each member should hold all the keys on the path from the leaf to the root. If a member joins the group, all the keys on the path should be changed to guarantee the backward security (BS) [9]. If a member leaves the group, the key set on the path should be changed to guarantee the forward security (FS).

2.3 LKH Key Tree Balancing Techniques

LKH is efficient when the structure is well balanced. Therefore, we need some specific techniques by which we can efficiently construct the LKH tree and keep the tree balanced while operating it. In [4], the authors propose the use of a B^+-tree for LKH implementation. Since it makes use of B^+-tree management techniques for operating LKH, while it does not need an additional tree-balancing phase, it requires, in normal times, a higher tree maintenance cost to make the tree balanced. In [3], two balancing techniques for a binary tree-based LKH tree are introduced. The first is that when a node is deleted, the highest leaf in the tree is searched and moved to the location of the node to be deleted. The second is that when a node is deleted, the tree is checked to see if a certain threshold value is out of the acceptable range. If it is, the highest node is searched and moved to the shallowest point until the threshold value returns to the acceptable range. This proposal, however, does not define any specific thresholds or specifies balancing algorithms, but only briefly enumerates some balancing rules. The technique proposed in [7] is to rebuild the key tree based on an AVL tree-building technique whenever a member is added or deleted. When a member is deleted, some keys related to the member are also removed, and many AVL sub-trees remain. These sub-trees are rebuilt to satisfy the AVL tree properties. When a member is added, the node is considered as a sub-tree and the same algorithm is

applied recursively until it satisfies the AVL tree properties. Although this decentralized algorithm is fault-tolerant and can solve the concentrated load problem of a single key server, it is inferior to [3] or [4] in terms of time complexity.

3 *LKHTreeManager*: The Proposed Key Tree Manager

In this section, we describe *LKHTreeManager*, our LKH key tree management algorithm. *LKHTreeManager* consists of three parts: an AVL search tree for membership management, a breadth first traversal (BFT) for insertion of a node, and a key tree balancing algorithm.

Structure of a Key Tree Node. The data structures of a key node and member information are shown in Fig. 1. The double links between a parent and a child make the tree traversal easy and the algorithm simple. We define a subtree T_i as follow:

Definition 1 (*Subtree* T_i). *A subtree* T_i *of a binary tree* T *is a tree that consists of a root* N_i*, the left subtree* T_{iL}*, and the right subtree* T_{iR}*.*

The variables x and y of T_i are initialized by 0 and defined as follows:

- $T_i.x = H(T_i) = \text{MAX}(H(T_{iL}.x), H(T_{iR}.x)) + 1$.
- $T_i.y = H(T_{iL}) - H(T_{iR})$.

In other words, x is the height of the tree T while y is the difference in the heights of the two sub-trees.

3.1 AVL Search Tree for Membership Management

When the group membership changes, the key management algorithm should minimize the location search time for the node and the number of key messages.

```
typedef struct node{
    node        *left;        //pointer to left node
    node        *up;          //pointer to up node
    character   keyValue;
    integer     x;            //height
    integer     y;            //height difference
    node        *right;       //pointer to right node
    membInfo    *member;      //pointer to the member
} Node;

typedef struct membInfo {
    character   nodeID        //ID of node
    character   name          //member name


}MembInfo;
```

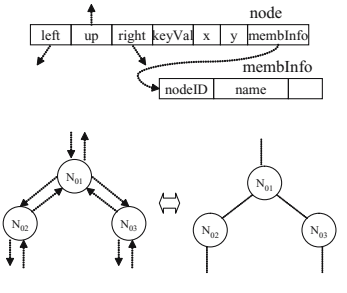


Fig. 1. Data structure of key tree node and member information

The ID of a group member has nothing to do with the location in the LKH key tree. Because a simple search technique requires $O(N)$ time, a large and highly dynamic group requires an efficient membership management technique in terms of both member insertion/retrieval/deletion time and storage requirement. We adopt an AVL search tree for membership management. It is efficient since its worst time complexity for insertion/retrieval/deletion operations are all the same to $O(\log_2 N)$ and storage usage is $O(N)$.

The structure of an LKH key tree combined with an AVL search tree is shown in Fig.2. *nodeID* is used for the search key and the algorithms for node addition and deletion are almost the same as general AVL search tree management algorithms. When a member is added to or deleted from a group, the AVL search tree should also be managed at the same time. In the case of delete operations, the AVL search tree greatly helps to locate the node in the key tree. However, in the case of add operations, it causes some burden since the addition location is determined by the member addition algorithm and the AVL search tree is notified for storing the location. Nevertheless, in Section 4, we show through simulation that the search tree greatly improves the efficiency of key tree management performance.

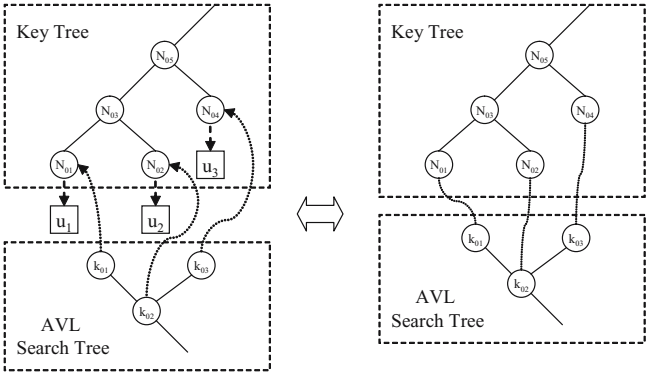


Fig. 2. Structure of the LKH key tree combined with the AVL search tree

3.2 BFT for Efficient Node Insertion

The management cost of the key tree is proportional to the number of key messages to be delivered, and this fact should be considered first in the design of group key management algorithm. Therefore, when a node is inserted, the algorithm should minimize the search time for the insertion location and the number of key messages. We propose the adoption of the BFT algorithm for finding the proper insertion location.

Node insertion at random may unbalance tree. However, if there is no functional relationship between a member ID and a location in the key tree, the node insertion operation can be adjusted to contribute the key tree to be more

balanced. For this purpose, the *LKHTreeManager* makes use of the BFT tree traversal technique, which visits the nodes in the key tree from top to down and from left to right, trying to find a node that has no child. The first node without a child is the node that becomes the sibling of the new node. Fig. 3(a) describes the algorithm in detail and Fig. 3(b) shows an example of a member addition.

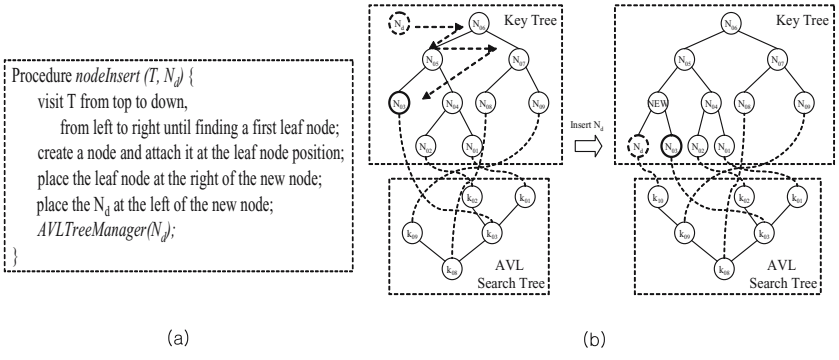


Fig. 3. Member insertion: (a) Algorithm; (b) Example

3.3 Efficient LKH Key Tree Balancing After Node Deletion

In a general binary tree, both the operations of node insertion and deletion can unbalance the tree. In particular, frequent delete operations may make the key tree heavily unbalanced.

Fig. 4(a) describes the proposed node deletion algorithm, *nodeDelete*, while Fig. 4(b) shows an example of a deletion that applies this algorithm. In order to delete a node in the key tree, the node is first searched in the AVL search tree, and the search tree provides the location information of the node in the key tree, and then the node is deleted from both trees. After deleting a node, *nodeDelete* checks the balancing factor, bf, to see if the tree needs to call the balancing algorithm, *keyTreeBalancer*. bf is defined as follow:

- $bf = 2^{H(T)-1}$.

In other words, bf is defined as the half of the maximum nodes the tree T can hold. If *threshold* is set as the number of the nodes in T, then the balancing is performed when bf is more than double of *threshold*.

Fig. 5(a) specifies the tree balancing algorithm, *keyTreeBalancer*. This balancing procedure lowers the average height of the key tree by transforming the tree into a complete structure. As we can see in Fig. 5, which that shows the procedure for applying *keyTreeBalancer*, before applying the algorithm, the average path length of (b) is 3.4, but after applying the algorithm, the value of (e) is 3.0. For tree traversing, *keyTreeBalancer* follows the depth first traversal (DFT) post-order technique.

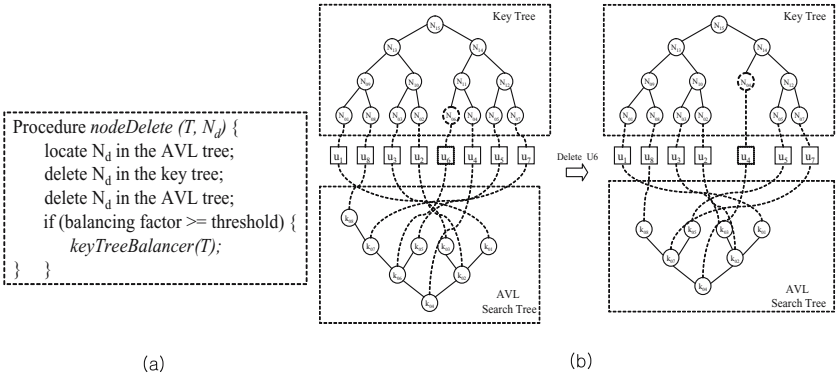


Fig. 4. Member deletion: (a) Algorithm; (b) Example

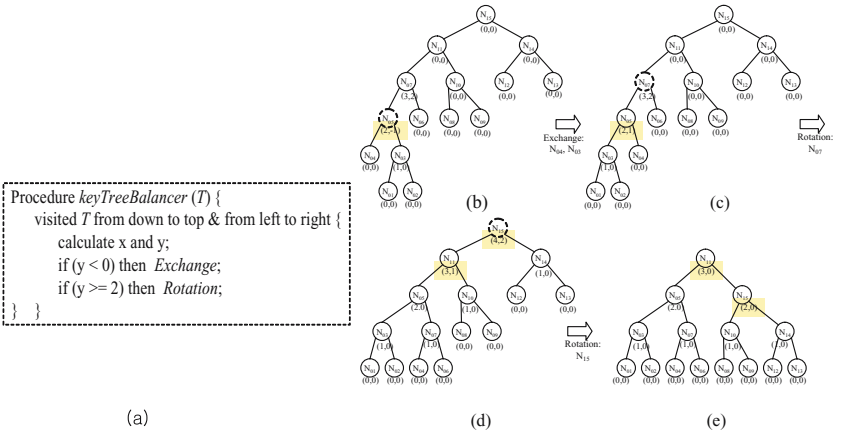


Fig. 5. The tree-balancing: (a) Algorithm; (b) Example

The two main subalgorithms in *keyTreeBalancer* are *Exchange* and *Rotation*. *Exchange* exchanges the positions of left and right sub-trees, which occurs when the height of right sub-tree is greater than that of left sub-tree. *Exchange* does not affect the group and sub-group keys, because the path from the root node of the subtree to the root of the key tree does not change. Because of *Exchange*, the *Rotation* operation can always be performed in a clockwise direction. This makes the *Rotation* algorithm simple and easy. Because *Rotation* changes the path, some keys should be changed. The number of additional key messages needed for a rotation is a maximum of three.

4 Simulation

In this section, we show the efficiency of *LKHTreeManager* (LTM for short) through simulations. The simulations are performed as follows: In the

initialization phase, a client sends a great number of consecutive join requests to the GKM. Receiving the requests, the GKM constructs an LKH key tree combined with an AVL search tree. In the next phase, the client randomly sends a great number of join or leave requests to the GKM. The performance is measured in terms of total execution time and the number of key messages to compare the efficiency of membership processing and the performance of the balancing algorithms respectively.

Comparison of Membership Processing Efficiency. In this simulation, we showed the effect of the AVL search tree on membership processing. The performance was measured by the total execution time for processing 1,000 addition or deletion requests that are randomly made by a client when the group size is 2,000, 4,000, 6,000, 8,000 and 10,000, respectively. The time unit is millisecond. Fig. 6 compares the results. BIN is the case of a normal binary key tree and AVL is the case of the binary key tree with the AVL search tree. We apply the same insertion and deletion algorithms that we propose in this paper to both BIN and AVL to show the effect of the AVL search tree. Fig. 6 shows that AVL processes the requests faster and requires less additional time for an increased group size compared to BIN. This means that the time spent on managing the AVL search tree and searching for the location quickly is much less than the time spent on searching for the location in the key tree without the search tree. Thus, we can conclude that in a disordered key tree, a key tree with an AVL search tree greatly improves membership-processing time.

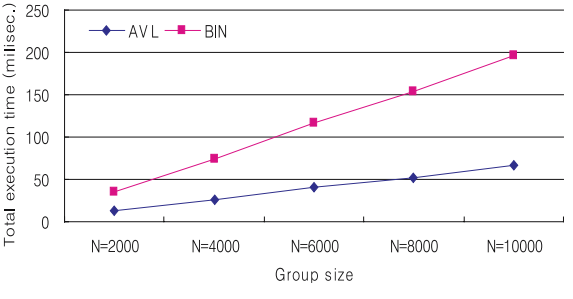


Fig. 6. Comparison of the effect of the AVL search tree in the key tree (Membership processing time of GKM)

Comparison of the Number of Key Messages. In this simulation, we compare the performance of LTM, the algorithm that we propose in this paper, with B^+-LKH, which is arguably the most efficient at present in many aspects [2], in terms of the number of key messages required. Fig. 7(a) shows the increase rate in the number of key messages when the initial group size is 1,000, the degree of B^+-LKH is 3, and the $\rho = \delta/\lambda = 1$, where δ is the number of joins and λ is the number of leaves. The dynamicity, which is defined by $\delta + \lambda$ during a unit time, begins with 100/100 and increases by 100 respectively until it becomes

600/600. Because $\rho = 1$, the group size does not change and is 1,000 on the average. According to the simulation, regardless of the degree of dynamicity, LTM requires less number of key messages than B$^+$-LKH does and is more efficient.

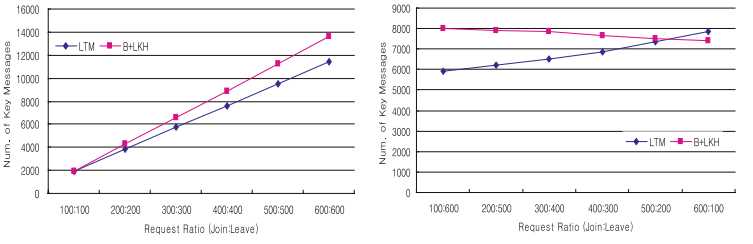


Fig. 7. (a) Comparison of the increase in the number of key messages by increasing dynamicity with fixed $\rho = 1$; (b) Comparison of change in the number of key messages by increasing ρ with fixed dynamicity

In the following simulation, we set the group size at 1,000, the same as in the previous experiment, but with changes in ρ. ρ begins with 100/600 and the leave rate decreases by 100 and the join rate increases by 100 until it becomes 600/100. Therefore, although the dynamicity does not change at all during the experiment, because ρ increases, the group size increases. Fig. 7(b) shows that LTM is more advantageous as ρ decreases, and B$^+$-LKH is advantageous as ρ increases. This is because when a node is added, in the case of LTM, the height of the node is immediately increased by one because of the characteristics of the binary tree, and it also affects the total height of the tree. However, in the case of B$^+$-LKH, the height is increased by a log function of k. When a node is deleted, in the case of LTM, the height of the node is immediately decreased. In the case of B$^+$-LKH, however, the height is decreased not only by a log function of k, but the number of key messages is also proportional to k. On the average, LTM is superior to B$^+$-LKH.

5 Conclusion and Future Work

In this paper, we proposed a group membership management technique that processes membership fast and an efficient LKH key tree-balancing algorithm that reduces the number of group key messages. The proposed algorithm is simulated to show that it processes the membership information about three times faster when the LKH key tree is combined with the AVL search tree and that it reduces the number of key messages by about 14% compared to B$^+$-LKH. Although the algorithm is designed mainly for very dynamic secure group communication systems, we also expect that the algorithm would be useful for general secure group communication systems. In the future, we need to apply

the proposed algorithm to a real multicasting system to show that it is really a very practical one.

Acknowledgements

This research is supported by Korea Research Foundation (KRF-2004-041-D00463).

References

1. M. Moyer, J. Rao, and P. Rohatgi, "A survey of security issues in multicast communications," in *IEEE Network*, vol.13, Nov./Dec. 1999.
2. D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architecture," IETF RFC 2627, June 1999.
3. M. Moyer, J. Rao, and P. Rohatgi, "Maintaining balanced key tress for secure multicast," draft-irtf-smug-key-tree-balance-00.txt, June 1999.
4. S. Ghanem and H. Abdel-Wahab, "A secure group key management framework: Design and rekey issues," *IEEE Computer Communication*, 2003.
5. C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. on Networking*, Feb. 2000.
6. C. Wong and S. Lam, "Keystone: A group key management service," *Proc. International Conference on Telecommunications*, May 2000.
7. O. Rodeh, K. Birman, and D. Dolev, "Using AVL trees for fault tolerant group key management," Hebrew University, Computer Science TR 2000-45, Nov. 2000
8. E. Horowitz, S. Sahni, and S. Anderson-Freed, *Fundamentals of Data Structures in C*, Computer Science Press, 1992.
9. H. Krawczyk, "SKEME: A versatile secure key exchange mechanism for Internet," *Proc. of the 1996 Symposium on Network and Distributed System Security (SNDSS '96)IEEE Symposium on Security and Privacy*, 1996.

Proposal for a Practical Cipher Communication Protocol That Can Coexist with NAT and Firewalls

Shinya Masuda[1], Hidekazu Suzuki[1], Naonobu Okazaki[2], and Akira Watanabe[1]

[1] Graduate School of Science and Technology, Meijo University 1-501 Shiogamaguchi, Tenpaku-ku, Nagoya-shi, Aichi, 468-8502, Japan
{m0432038, m0432022}@ccmailg.meijo-u.ac.jp,
wtnbakr@ccmfs.meijo-u.ac.jp
[2] Faculty of Computer Science and Systems Engineering, University of Miyazaki 1-1 Gakuen Kihanadai, Miyazaki-shi, Miyazaki, 889-2192, Japan
oka@cs.miyazaki-u.ac.jp

Abstract. Threats to network security have become a serious problem, and encryption technologies for communications are an important issue these days. Although the security of IPsec ESP (, that is a typical existing cipher communication technology) is strong, it has such problems that it can not be used in the environment where it coexists with NAT and firewalls, and that there also exists some degradation of throughput. For such reasons, ESP is used only for some limited applications such as VPN (Virtual Private Network). In this paper, we propose a new cipher communication protocol, called *PCCOM (Practical Cipher COMmunication)*, that can verify the identity of the corresponding counterpart and assure the integrity of packets in the environment where it coexists with NAT and firewalls, without changing the format of the original packets. To confirm the effectiveness of PCCOM, we installed a trial system in FreeBSD, and confirmed the coexistibility with NAT and firewalls. We also measured its throughput, and good performance was confirmed, which is attributable to "no change" of the packet format.

1 Introduction

Threats to network security have become a serious problem these days, and the importance of security technologies is increasing. In particular, network security technologies to ensure the security of a network by encryption of packets in the IP layer such as IPsec ESP [1]-[3] are expected to be the basic security technology for network. Actually, however, IPsec ESP is not so widely spread because its use is restricted only to the environment without NAT/NAPT (NAT hereinafter) and firewalls. For this reason, there is demand for different technologies that can coexist with NAT and firewalls. But, good security and practicality conflict with each other and it is difficult for one technology to meet both requirements. For future security technologies, therefore, it will be important to choose appropriate technologies depending on the situation.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 713 – 722, 2006.

ESP provides such functions as encrypting packets to prevent eavesdropping, verifying the identity of the correspondent to prevent masquerades and assuring the integrity of packets to prevent their manipulation and so on. ESP is available in two modes; i.e. transport mode and tunnel mode. The former is used for End-to-End communications and the latter for Gateway-to-Gateway or Host-to-Gateway communications. In reality, however, it is not so widely used except when the tunnel mode is used in Gateway-to-Gateway as a means of constructing the Internet VPN (Virtual Private Network). This is considered to be attributable to the fact that ESP can not pass through NAT and firewalls established by packet encryption and integrity assurance.

On the other hand, there is a technology that encrypts a particular range without changing the packet format to pass through firewalls, and to reduce the overhead of relay performance (hereinafter *Replacement Method*) [4]. Replacement Method, however, can not pass through NAT, and does not realize identity verification and packet integrity assurance.

In this paper, we propose a new cipher communication protocol, called *PCCOM (Practical Cipher COMmunication)*. PCCOM succeeds to the merits of Replacement Method that it does not change packet format. PCCOM realizes the verification of identities and assurance of the integrity of packets by recalculations of TCP/UDP checksum [5]-[7] using *Pseudo Data* generated with a common secret key and contents of the packet. With this method, packets converted by PCCOM can pass through NAT, and a high throughput can be achieved because the packet format is not changed. The encryption range of PCCOM is only the user data portion to pass through a firewall, but the necessary minimum level of security is maintained because it realizes integrity assurance for the entire packet. Premises of PCCOM are that the common secret key must be shared between both terminals in advance and the process information table describing the process of packets must already be built correctly.

In order to confirm the effectiveness of PCCOM, we have developed a trial system. As PCCOM is a method that processes without changing the packet format, implementation is easy and it has performance advantages. As the result of our evaluation, we confirmed that a high throughput can be achieved.

This paper is composed as follows. Section 2 describes existing technologies and their constraints, Section 3 proposes PCCOM, Section 4 describes the implementation of PCCOM, Section 5 is a performance evaluation of PCCOM. Finally Section 6 is a conclusion.

2 Existing Cipher Communication Technologies and Their Constraints

Fig. 1 shows the packet formats of the transport mode and the tunnel mode of IPsec ESP. In the case of transport mode, ESP header is inserted between the IP header and its payload, and the payload portion of the original IP packet is encrypted. ESP trailer adjusts data size to the block size of encrypted data. ICV (Integrity Check Value) is calculated and added as ESP authentication value to the end of the packet to assure the integrity from the ESP header to the ESP trailer. In the case of tunnel mode, the

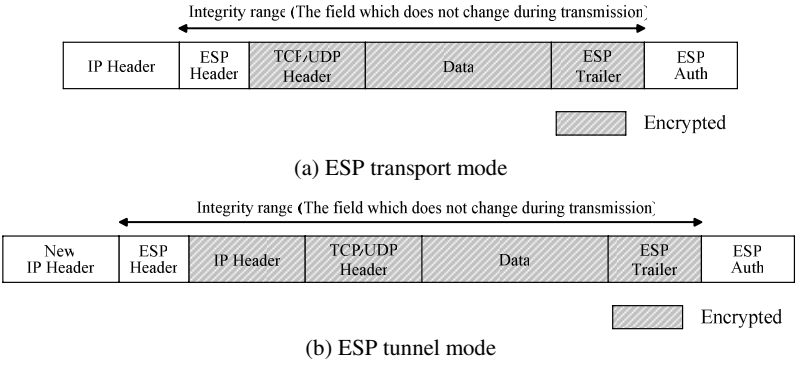


Fig. 1. Packet format of IPsec ESP

encapsulating is done with a new IP header having IP address of the security gateway, thus assuring the integrity of the data from the ESP header to the ESP trailer.

In both modes, the port number field is in the encryption range, and so firewall can not judge the purpose for which the packet is used. As a result, the firewall often prohibits passing of all IPsec packets. Since the TCP/UDP checksum field is in the encryption range/integrity assuring range, when the packet passes NAT, it is taken as a forged packet and is discarded by IPsec process because NAT translates IP address and recalculates the TCP/UDP checksum field. The problem is that the TCP/IP does not have a good hierarchical structure, and the IP address is included in the checksum calculation range. To cope with this situation, a method to pass through the NAT by encapsulating ESP with UDP header is proposed [8], but the method can not include the encapsulating header portion in the range of integrity assurance and increases overheads due to the header addition.

Although the security strength of the IPsec is strong enough, it is necessary to consider the affinity with the existing systems such as NAT and firewalls and also the throughput degradation.

Fig. 2 shows the packet format of the Replacement Method, which is the basis for PCCOM. The packet format is not changed from the original format and the plaintext and ciphertext are replaced as they are. The encryption range covers all the portions after the TCP/UDP checksum field so that the firewall can identify the port number and prevent the ciphertext from being guessed from the TCP/UDP checksum. This method is effective in the intranet, but it has a constraint that it can not pass through NAT accompanied by recalculations of the TCP/UDP checksum. Another constraint is that there is a possibility of spoofing and manipulation because it does not realize the identity confirmation and integrity assurance of the packet.

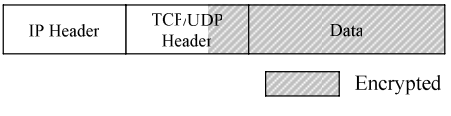


Fig. 2. Packet format of the replacement method

3 Proposal of PCCOM

The functions provided by PCCOM are to secure confidentiality by encryption, realize identity confirmation and integrity assurance of a packet, and it can coexist with NAT and firewalls, and since it does not change the packet format, it can realize a high throughput. IP addresses and port numbers are not included in the range of integrity assurance because they are translated by NAT. In this respect, IP addresses and port numbers can be assured by a table search process of a process information table describing the process contents of packets.

3.1 Principle of PCCOM

Fig. 3 shows the packet format of PCCOM. PCCOM applies its unique calculation to the TCP/UDP checksum using a special value, called *Pseudo Data* generated by the common secret key and the contents of the packet, thus realizing the identity confirmation and integrity assurance of the packet. Its principle is shown as below.

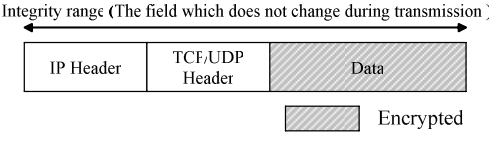


Fig. 3. Packet format of PCCOM

With PCCOM, a checksum base value called *CB (Checksum Base)* is first defined to realize the identity confirmation and integrity assurance. CB is the hash value of the common secret key shared in secret in advance, and parts of header contents which do not change during transmission in IP header and TCP/UDP header (gray portions in Fig. 4). Since as the seeds of the CB, a common secret key and a sequence number, which differs for every packet are included, it is very difficult for a third party to guess the CB value. This CB is the key data to realize the identity confirmation and packet integrity assurance as shown below.

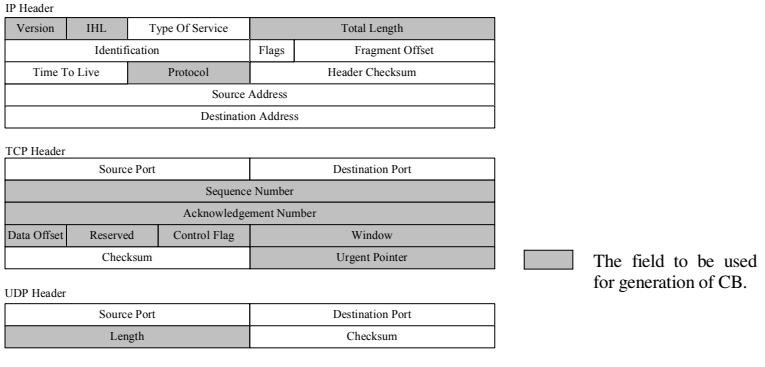


Fig. 4. The field to be used for generation of CB

Fig. 5 shows a difference in calculation range of TCP/UDP checksum between general communication and PCCOM. The dotted lines in the figure indicate pseudo information created when the checksum was calculated. In normal communication, TCP/UDP checksum is calculated from TCP/UDP header, TCP/UDP pseudo header, and user data. TCP/UDP pseudo header includes IP addresses. In the case of PCCOM, it is calculated from TCP/UDP header, TCP/UDP pseudo header, and Pseudo Data. Pseudo Data is the hash value of encrypted data and CB described as above.

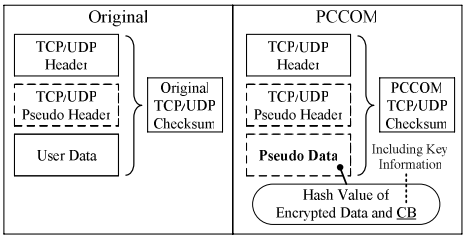


Fig. 5. Calculation range of checksum

The flow of the integrity assurance is described below. Sending terminal recalculates TCP/UDP checksum with Pseudo Data after data encryption. Receiving terminal verifies TCP/UDP checksum with Pseudo Data generated by the same method before data decryption. If the verification result is correct, it decrypts the user data and recalculates the original checksum, and gives the packet to the upper layer (TCP/UDP). With this method, it is possible to assure the integrity of the packet and also to realize the identity confirmation. Even if manipulators try to manipulate part of the packet and recalculate TCP/UDP checksum to conceal the manipulation, they can not calculate correctly because they do not know the content of Pseudo Data. Besides, IP addresses and port numbers are not included in the range of CB generation because they are translated by NAT when it is coexisted. The assurance of IP addresses and port numbers are realized by the method described in the following paragraph.

With the above calculation method, even if the IP address, port number, and checksum are renewed due to the existence of NAT on the communication path, the ideas of the integrity assurance and identity confirmation are maintained. That is, the verification of the checksum on the destination terminals is not affected, because recalculation of checksum in NAT is only to calculate the difference of the translated portion [9]. With the PCCOM, the encryption range of the packet is after the user data portion, but since the identity confirmation and packet integrity assurance are applied, it is possible to prevent attacks such as TCP session hijacking. Since the checksum field is only 16 bit length, crackers may succeed in the manipulation of a packet in the probability of $1/2^{16}$, but even if they could succeed in the manipulation, they would not be able to send intentional data because the user data is encrypted. With PCCOM, as firewall can use filtering functions by checking the contents of TCP/UDP header, practical merit of this method is considered large.

For the encryption algorithm, CFB mode of block cipher which is capable of any length data size is adopted. Therefore, we do not need to worry about the occurrence of fragment because the packet length is not changed.

3.2 Assurance of IP Addresses and Port Numbers

With the PCCOM, IP addresses and port numbers are not included in the range of CB generation because their values are changed when passing through NAT. The integrity of these portions is assured by a table search process of a process information table describing the process of packets. Fig. 6 shows the process of the table search. The process information table consists of IP addresses, port numbers, protocol number, process of packets such as encryption/decryption, relay transparently and discard a pointer of the common secret key, and a cipher algorithm. This table is searched with IP addresses, port numbers, and protocol number of the receiving packet. After the table search, IP addresses, port numbers, and protocol number are rechecked from the contents of the table, and if the information of the relevant packet correctly exists in the table, it is assured that IP addresses and port numbers in the packet are correct.

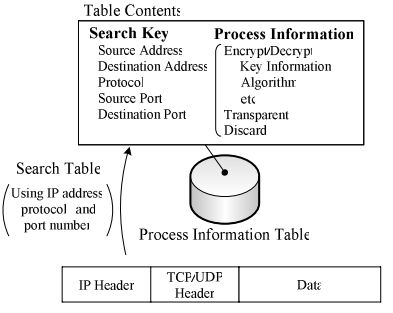


Fig. 6. Table search process

This method is based on the condition that a table of correct contents is generated in advance. As a method to assure the generation of a correct table, such existing technologies as IKE (Internet Key Exchange) [3] can be used.

4 Implementation

We have developed a trial system of PCCOM and verified its operation. In this chapter, the implementation method of the trial system, specifications, construction, and its operation are described.

4.1 Implementation Method

The trial system is installed in the kernel of FreeBSD (5.1 Release). Fig. 7 shows the implementation method of the trial system. This system does not modify the existing process in the IP layer. The process of the PCCOM module is simply called from ip_input() and ip_output() (kernel space function), and returned after the processing is completed. The reason why this implementation is possible is that PCCOM does not change the packet format. In case of IPsec, the process must be changed over the entire IP layer because the packet format is changed (for instance, by the addition of a header). Therefore, PCCOM has an advantage of a high throughput.

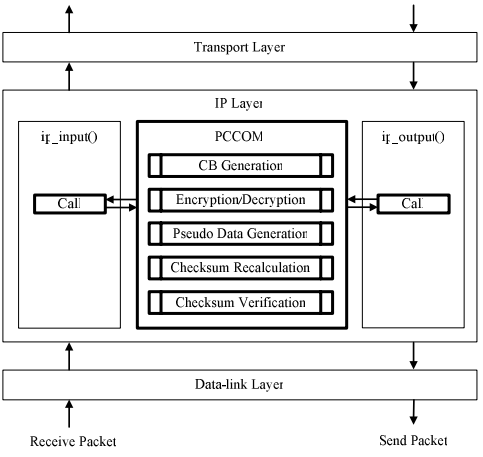


Fig. 7. Implementation method of the trial system

4.2 Specifications and Construction of the System and Its Operation

Table 1 shows the specifications of the trial system. The process information table is implemented as a hash table. For a cipher algorithm, AES (key length of 128 bits) is adopted, and MD5 is adopted for hash function. As a cipher library, OpenSSL (openssl-0.9.7d) is used.

Table 1. Specification of the trial system

Items	Contents
Table search method	Hashing
Cipher algorithm	AES (CFB mode)
Key length	128 bit
Hash function	MD5

PCCOM consists of CB generation module, encryption/decryption module, Pseudo Data generation module, checksum recalculation module, and checksum verification module. PCCOM executes pre-determined operation to the sending or receiving packet according to the process information table. The process information table contains IP addresses, port numbers, and protocol number, and corresponding process operation such as encryption/decryption, relay transparently, or discard. First, PCCOM calculates the hash value from IP addresses, port numbers, and protocol number in the packet and retrieves the table and checks that the correct IP addresses, port numbers, and protocol number exist in the table. Second, it executes the relevant process according to the process operation described in the table.

Using the trial system, we have confirmed that the communication can be performed via firewall of packet filtering type and NAT, and also have confirmed that a packet can be detected as illegal when the contents of the packets are manipulated.

5 Performance Evaluation of the Trial System

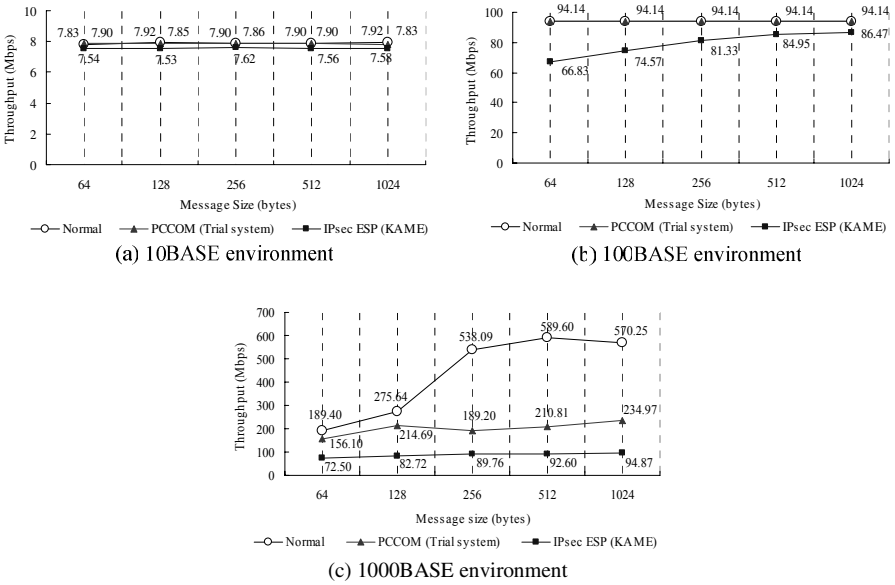
We have measured communication performance between two terminals implemented with the trial system. For reference, we have also measured the ones implemented with IPsec ESP (KAME). We also measured the process time in PCCOM for each module and clarified the portion forming a bottleneck for processing. Table 2 shows the specifications of the terminals used for the tests. As parameters for ESP, operation mode is transport mode, encryption algorithm is AES (key length of 128 bits), authentication algorithm is HMAC-MD5 and Replay prevention is disabled so that the conditions would be same as the specifications of PCCOM trial system.

Table 2. Specification of the terminals

Items	Contents
CPU	Pentium4 2.4GHz
Memory	256MB
NIC	10BASE-T,100BASE-TX, 1000BASE-TX
OS	FreeBSD (5.1 Release)

5.1 Measurement of Communication Performance

Fig. 8 shows the relationship between the IP packet size and the throughput among the cases of non-ciphering (Normal), PCCOM, and IPsec ESP in each Ethernet type of



(a) 10BASE environment

(b) 100BASE environment

(c) 1000BASE environment

Fig. 8. Measurement results of throughput

10BASE, 100BASE, and 1000BASE. For measuring the throughput, the network bench mark software *Netperf* [10] is used. The value in the figure is the average of 10 trials.

In the environment of 10BASE, some degradation in performance is detected with the ESP, but since the number of packets processed is not so large, the processing overhead does not form a bottleneck for both PCCOM and ESP. In the environment of 100BASE, Normal and PCCOM show the upper-limit performance of NIC, and no degradation in performance is detected with PCCOM. With ESP, on the other hand, the performance degrades about 8.2% for a packet of 1024 byte length (long packet) and about 29% for a packet of 64 byte length (short packet) in comparison with Normal. In the environment of 1000BASE, PCCOM degrades about 58.8% in performance for the long packet and ESP about 83.4% in comparison with Normal. In case of the short packet, PCCOM degrades about 17.6% in performance and ESP about 61.7% in comparison with Normal.

The shorter the packet size is, the lower the throughput is, because the number of packets to be processed increases. Especially for the short packet of ESP, processing bottleneck other than the encryption such as the header addition appeared remarkable.

Fig. 9 shows the download time of a 500MB file with FTP in the environment of 1000BASE. The measurement results are the average value of 10 trials. While PCCOM requires about 145.1% of time in comparison with Normal, ESP requires about 311.6% of time.

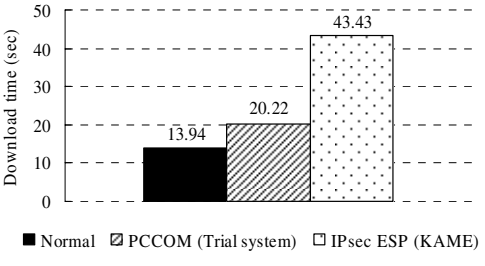


Fig. 9. Download time of a 500MB file using FTP

5.2 Processing Cost in PCCOM

In order to estimate the processing cost in PCCOM, we have measured the internal processing time of PCCOM for each module. The internal processing time is calculated with the CPU clock counter values before and after the processing, using the RDTSC (ReaD Time Stamp Counter).

Table 3 shows the processing time of each module and the ratio. The measurement result is the average value of the results of packets of 1460 byte length during the communication of FTP. From Table 3, it is shown that the encryption/ decryption takes most of the processing on both transmission-side and receiving-side. A large reduction in processing time can be expected by using a dedicated hardware cipher engine, and it is considered that the performance closer to that of Normal can be achieved.

Table 3. Processing time of the modules and their ratios

	Modules	Processing time (μs)	Ratio (%)
Sending-Side	CB Generation	0.868	3
	Encryption	26.043	90
	Pseudo Data Generation	1.704	6
	Checksum Recalculation (Original)	0.294	1
Receiving-Side	CB Generation	0.890	3
	Pseudo Data Generation	2.863	9
	Checksum Verification (Original)	0.281	1
	Decryption	25.547	83
	Checksum Recalculation (Normal)	1.286	4

6 Conclusion

We have proposed PCCOM that can realize identity confirmation and integrity assurance of the entire packet and that can coexist with NAT and firewalls without changing the format of the original packet. PCCOM realizes the functions by recalculating TCP/UDP checksum using Pseudo Data generated with a common secret key and contents of the packet. To confirm the effectiveness of PCCOM, we have developed a trial system. As the result of our performance evaluation, we have confirmed that PCCOM can achieve a high throughput.

References

1. S. Kent and R. Atkinson “Security Architecture for the Internet Protocol”, RFC2401, Aug. 1998.
2. R. Atkinson, “IP Encapsulation Security Payload (ESP)”, RFC2406, Dec. 1998.
3. D. Harkins and D. Carrel, “The internet key exchange (IKE)”, RFC2409, Dec. 1998.
4. A. Watanabe, Y. Koui, T. Ideguchi, Y. Yokoyama and S. Seno, “Realization Method of Secure Communication Groups Using Encryptions and Its Implementation”, Trans. IPS Japan, vol.38, no.4, pp.904-914, Apr 1997.
5. R. Braden, D. Borman, and C. Partridge, “Computing the Internet Checksum”, RFC1071, Sep. 1988.
6. T. Mallory and A. Kullberg, “Incremental Updating of the Internet Checksum”, RFC1141, Jan. 1990.
7. A. Rijsinghani, “Computation of the Internet Checksum via Incremental Update”, RFC1624, May. 1994.
8. A. Huttunen, B. Swander, V. Volpe, L. Diburro, and M. Stenberg, “UDP Encapsulation of IPsec Packets”, RFC3948, Jan. 2005.
9. K. Egevang and P. Francis, “The IP Network Address Translator (NAT)”, RFC1631 May. 1994”.
10. Netperf, http://www.netperf.org

An Integrated Scheme for Intrusion Detection in WLAN

Dong Phil Kim, Seok Joo Koh, and Sang Wook Kim

Department of Computer Science, Kyungpook National University
1370 Sankyuk-dong Buk-gu, Daegu, 702-701, Korea
{dpkim, sjkoh, swkim}@cs.knu.ac.kr

Abstract. Wireless Local Area Network (WLAN) is susceptible to security provisioning in spite of the solutions such as the Wired Equivalent Protocol (WEP) or IEEE 802.1x. This paper proposes an integrated scheme for intrusion detection in WLAN systems. The proposed scheme operates with one or more Gathering Agents (GAs) and a Master Server (MS). Each GA is used to get security information by collecting the frame packets in WLAN, whereas the MS is purposed to detect and prevent the various attacks by analyzing the packets in the WLAN systems. A detection engine contained in the MS employs OUI list matching for detection of MAC spoofing attacks, sequence number analysis for man-in-the-middle attacks, and Finite State Machine (FSM) analysis for Denial-of-Service (DoS) attacks. By experiments, it is shown that the proposed scheme could effectively detect and prevent the various attacks that could possibly be done in the WLAN systems.

1 Introduction

Wireless Local Area Network (WLAN) is one of the key wireless access technologies and has been rapidly spread out in the world-wide markets. One of the challenging issues on WLAN is the security problem. In particular, the security issue has so far been studied by many researchers, but the WLAN is still vulnerable to the promising attacks [1, 2, 3].

The WLAN basically has the broadcast nature in the radio transmissions, and thus anyone in the same network coverage may access to all the transmitted packets. It implies that the WLAN could be highly susceptible to security attacks. Furthermore, the security problem of WLAN has still been one of the key issues for commercial deployment, in spite of the existing solutions contained in the firewall or Enterprises Security Management (ESM). Some solutions have so far been proposed for the WLAN security management. Such security mechanisms include the Wired Equivalent Protocol (WEP) and IEEE 802.1x, and etc. The WEP and 802.1x cannot ensure to provide the complete detection/protection against a variety of promising attacks.

In this paper, we propose an integrated scheme for intrusion detection in the WLAN systems. We describe the architecture of the integrated mechanisms for intrusion detection in WLAN and show how the proposed scheme could

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 723–732, 2006.

detect the various promising attacks, and then discuss some experiments over the test networks. This paper is organized as follows. Section 2 describes the existing WLAN security solutions. In Section 3, we describe the proposed scheme for intrusion detection. Section 4 describes how to detect the various attacks by the proposed scheme. Section 5 discusses some experimental results for the proposed scheme over the test-bed networks. Finally, Section 6 concludes this paper.

2 Existing Security Solutions for WLAN

This section briefly reviews the security scheme defined in the 802.11 standard and the existing WLAN security solutions: Wired Equivalent Protocol (WEP) and IEEE 802.1x.

The open system authentication is the default authentication mechanism for 802.11. It operates in the simple two-step process. First, the station who wants to authenticate with another station sends an authentication management frame containing the identifier of the sending station. The receiving station then responds with a frame indicating whether it can recognize the identity of the sending station. The open system authentication is too much simple and thus rarely provides a high-level security [4]. That is, it is easier for an attacker to connect to the network and to launch the attacks.

On the other hand, it is assumed in the shared key authentication that each station has received a secret shared key through a secure channel, which is independent of the 802.11 networks. Stations perform the authentication based on the shared secret key. Use of the shared key here requires an implementation of the Wired Equivalent Privacy (WEP) algorithm [2, 6]. The WEP was designed to provide confidentiality for network traffic using 802.11. The details of the algorithm used for WEP are beyond the scope of this paper. Yet, the WEP has been reported that it is still vulnerable to security owing to the relatively short initial vectors and statically managed keys [2]. This leads to the attacker decrypting some portion of the 802.11 frames [7, 8].

The IEEE 802.1x [5] has been proposed for the port-based network access control, which is used to provide the network security for WLAN. It provides the centralized authentication of wireless clients with authentication servers such as RADIUS or DIAMETER. Since the 802.1x is used for denying unauthenticated network access, we can prevent the misuse of network resource from illegal users. However, 802.1x is still vulnerable to attacks, which take the availability of network resource, such as Denial-of-Service [7]. It cannot authenticate all the packets. Accordingly, it is possible for an attacker to place a hub between an 802.1x authenticating switch and a legitimate user for physical access to the wires [9].

To enhance the security of the WLAN, this paper proposes an integrated scheme for intrusion detection, which could be used to detect abnormal behaviors of the prospective malicious users by monitoring and analyzing all the wireless packets on the WLAN. The proposed detection scheme is designed to

characterize the various patterns of intrusions/attacks that could be done in the WLAN system.

3 The Proposed Scheme

In this section, we describe the proposed integrated scheme for intrusion detection. Figure 1 shows an overall architecture of the WLAN security system based on the proposed scheme. The system consists of gathering agent (GA) and master server (MS). The GA supports the real-time monitoring of the packets flowing in the WLAN system. It collects and analyzes the wireless packets for detecting the prospective attacks. In particular, the GA is used to monitor the current status on the stations by analyzing the 802.11 MAC frames.

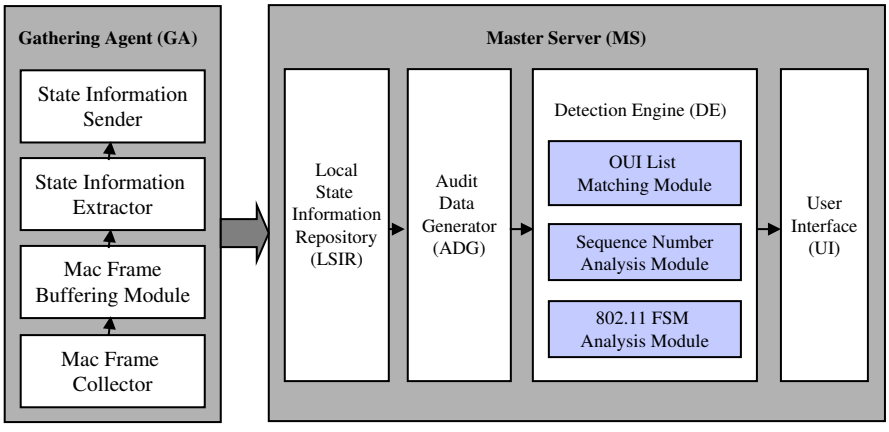


Fig. 1. Architecture of the proposed intrusion detection system for WLAN

With the analysis of those wireless packets, the MS will determine whether or not any attacks are intended in the WLAN systems. For this purpose, the MS provides the following four functions: Local State Information Repository (LSIR), Audit Data Generator (ADG), Detection Engine (DE), and User Interface (UI).

The LSIR stores the status information for the stations in the WLAN system by receiving the relevant frames from the GAs, and then forwards each data to the ADG. The ADG will analyze the packets to generate some audit data, and forward them to the DE. The DE consists of the following three detection modules: 1) OUI list matching, 2) sequence number analysis, and 3) 802.11 FSM analysis. Each module could be executed in the sequential order so as to detect the MAC spoofing, man-inthe- middle and Denial-of-Service attacks. As such, the intrusion detection will be performed in the integrated manner. If an attack is detected by the DE, it is informed to the UI, and further notified to the security officer.

The GA in the proposed scheme is used to support status monitoring between the wireless client terminals and an AP. The GA collects the MAC frames by using the MAC Frame Collector (with a wireless NIC) supporting RFMON mode, and then delivers the collected packets to the MAC Buffering Frame module. The MAC Buffering Frame module will take only the frames associated with management. It is noted that the management frames are purposed to establish communications between stations and AP, and thus to provide services for association establishment and authentication. Accordingly, a GA could extract the state information from those management frames.

The state information for AP is taken from Beacon and Association Response frames, the information for wireless client terminal could be taken from Probe Request, Authentication and Association Response frames. The state information for an AP includes the MAC address, SSID, available channels, transmission rates, and the number of the wireless client terminals associated with the AP. On the other hand, the state information for wireless client terminal includes the MAC address, SSID, available channel, encryption scheme, the delivered packet count and the current connection state. The GA will forward such state information to the MS.

3.1 Intrusion Detection

The master server (MS) determines whether or not any attack is being intended in the WLAN system. The received state information is stored in the Local State Information Repository (LSIR). Audit Data Generator (ADG) then generates the audit data for deciding whether or not the attack is being progressed. The ADG uses the MAC address and current connection state and transmitted packet counts from the state information for each station. Such the audit data will be forwarded into the Detection Engine (DE). The DE consists of the following three modules: OUI list matching, sequence number analysis, and 802.11 FSM analysis modules.

3.1.1 OUI List Matching

The OUI list matching module can be used to detect the conventional MAC spoofing attack. The stations on WLAN communicate each other by using the MAC addresses. It is noted that the MAC addresses could be used as a unique layer 2 identifier for the station in the WLAN. In the proposed scheme, the OUI list matching function uses such an OUI list so as to evaluate all source MAC addresses on the network. In the scheme, the detection of using a wrong prefix (which has not been allocated yet by the IEEE) can be reported as an anomalous activity.

3.1.2 Sequence Number Analysis

The sequence number analysis module is used to provide the protection against the man-in-the-middle attack, in which a rogue client may try to steal a real client MAC address and then to associates with the access point. The sequence number analysis module can be used to detect this kind of attack.

It is noted that the sequence number field in the MAC frame is a sequential counter that is incremented by one for each frame, starting at 0with a modulo of 4,096. Thus, it is unlikely that the MAC frames from the two different stations have the same sequence number [9]. By monitoring the sequence numbers in frames, we can detect the man-in-the-middle attack that is subject to the faked frames or illegally injected frames.

In the proposed scheme, the sequence number analysis function will compare the sequence numbers of the recently received audit data frames. If the gap of the sequence numbers between two consecutive frames is greater than two, the current audit data might be reported as an anomalous frame by the man-in-the-middle attack. Otherwise, the frame will be passed to the next step, the FSM analysis module, as described below.

3.1.3 FSM Analysis

The Finite State Machine (FSM) analysis is already employed to keep track of the status of a station in the 802.11 standard, in which the three states are defined for a station: listen, authentication, and association. In this paper, we design the FSM analysis module by extending the three states into the seven ones. The proposed FSM module can be used to provide the protection against the Denial-of-Service (DoS) attack, in which an attacker may try to exhaust most resources of the host or network, and thus render them unavailable to the legitimate users.

In general, an FSM progresses a system through a sequence of pre-defined states by transitions from one state to another. A transition occurs in response to events. In this paper, we redefine the FSM of 802.11 in terms of the connection states and generated events, as described in Table 1. As described in Table 1, we define the seven states for the proposed scheme. The LISTEN state is the starting point for the connection between AP and client terminals. Both AP and client terminals perform the authentication steps so as to verify each other. At this time, the states enter the AUTHENTICATION_REQUESTED and AUTHENTICATION_ESTABLISHED. They then try to establish an association by using the management frames. At this time, the states enter the ASSOCIATION_REQUESTED and ASSOCIATION_ESTABLISHED. If the connection is terminated gracefully, the state goes into CLOSED. Otherwise, the connection is abnormally terminated, the state will be in FAILUED.

The seven events used for transition of states in the FSM model are as follows:

P_0 : this event is generated when the AP broadcasts Beacon frames;
P_1 : this is generated when the station sends Probe Request frames, and receives Probe Responses from the AP;
P_2 : this is generated when the AP replies with Authentication frames for authentication;
P_3 : this is generated when the client terminal requests an association to the AP by sending an Association Request frame;
P_4 : this is generated when the AP agrees to open a connection for the terminal by sending an Association Response frame;

Table 1. Description of states for the proposed FSM analysis

Symbol	State	Description
L	LISTEN	All connections must start in a Listen state.
T1	AUTHENTICATION_REQUESTED	When the first Probe Request and Probe Response are sent, the connection is in the AUTHENTICATION_REQUESTED.
T2	AUTHENTICATION_ESTABLISHED	If the Authentication frame is sent and AP authenticates station,the connection is in the AUTHENTICATED_ ESTABLISHED.
H	ASSOCIATION_REQUESTED	When the Association Request is sent, the connection is in the ASSOCIATION_REQUESTED.
A	ASSOCIATION_ESTABLISED	When the Association Response is received, connection enters ASSOCIATION_ESTABLISED.
C	CLOSED	When the Deauthentication or Diassociation is sent, the connection is in CLOSED.
F	FAILED	When the Deauthentication/ Diassociation/Probe Request are sent more than Threshold, the connection is in FAILED.

P_5 : this is generated when the connection fails to authenticate by a Deau then authentication frame;
P_6 : this is generated when the connection enters the FAILURE state;
P_7 : this is generated when the number of frames transmitted is greater than the prespecified threshold in the fixed time interval;
P_8 : this is generated when the connection is terminated and goes into the LISTEN state.

Figure 2 illustrates the expected state transition diagram by the proposed FSM model, which is based on the states and events described above.

4 Detection of Attacks by the Proposed Scheme

This section describes how to detect any attacks with the help of the proposed schemes

4.1 MAC Address Spoofing

The OUI list matching module has already been equipped for some of the commercial products to detect the MAC spoofing attack. A malicious attacker may perform the MAC spoofing attacks by randomly generating the MAC addresses. The system may keep the OUI list for state monitoring, and the OUI list matching algorithm works well against the MAC spoofing attack.

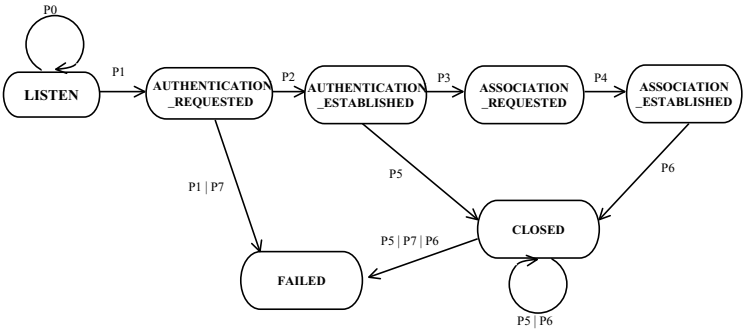


Fig. 2. Finite state transition for WLAN

4.2 Man-In-the-Middle Attacks

The man-in-the-middle attack will be done by a malicious user to inject invalid traffic during an association between an authenticated user and AP. To detect the man-inthe- middle attack, the sequence numbers of the MAC frames are analyzed. It is noted that the sequence number appears out of order when a man-in-the-middle attack is being made from injection of invalid traffic. For example, if an AP is assumed to start an association with a client station with 2,091. The next sequence number would be 2,092. The sequence number analysis module is invoked to detect the difference of the sequence numbers between the frames transmitted. Once an attack is initiated, the gap of the consecutive frame sequence numbers would be grater than two.

4.3 Denial-of-Service Attacks

A malicious node can deplete the resource of the network by transmitting a large of packets. The proposed system will detect this attack by measuring the total number of packets received from each node. If this count (total number) exceeds a pre-specified threshold, then an alert for the DoS attack is signaled. The threshold for the DoS attack may be configured from the experimentation. In the proposed FSM analysis, the FAILED state is considered as anomaly activity. For an example, the DoS attack based on the Disassociation or Deauthentication frames could be triggered by the P_6 or P_7 event for one station. For example, a system administrator might set the threshold for P_7 to be 40 frames per minute. In this case, if the number of the frames transmitted by the same source is more than 40 over one minute, then the P_7 event will be triggered.

5 Experimental Results

In this section, we describe some experimental results of the proposed scheme for intrusion detection. To perform the experimentation, we have implemented the gathering agent (GA) for frame monitoring and master server (MS) for detection of attacks.

5.1 Test Scenario

To experiment the proposed scheme for intrusion detection, we construct a small testbed, as shown in Figure 3, which consists of an AP and two mobile stations. For the test purpose, we have also employed GA, MS and three kinds of attackers on the testbed. For the attacker 1, Wellenreiter is used for MAC spoofing attack, and for the attacker 2 the WLAN-JACK is used for the man-in-the-middle attack, and the NetStumbler is used for the DoS attack.

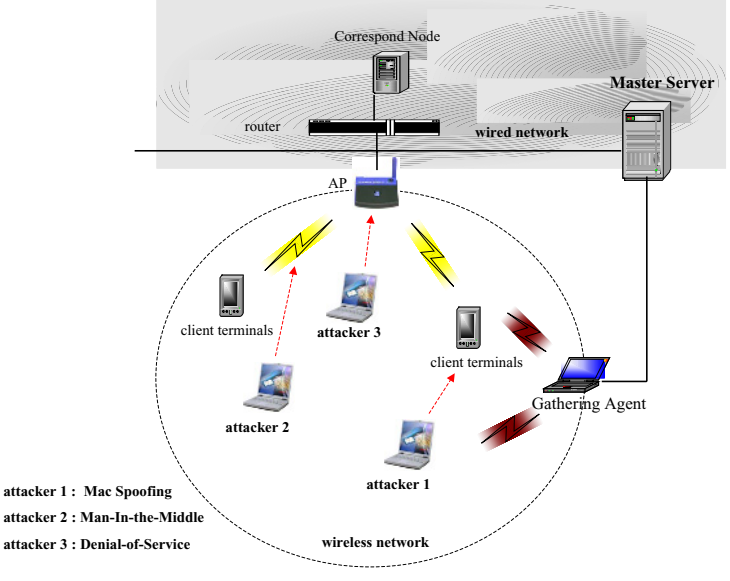


Fig. 3. Test environment for experiments

In Figure 3, the attacker 1 using Wellenreiter generates the random MAC address values ranged between 0x000000 and 0x00FFFF for the OUI portion (3bites) and then prepends a prefix value of 0x00 so as to avoid generating the MAC addresses conflicted with the reserved and multicast address space. In WLAN-JACK, it is assumed that attacker 2 monitors a pattern of legitimate sequence numbers. Attacker 2 identifies the Deauthenticate frames and then sends some spoofed Deauthetnicate frames over the broadcast address. Attacker 3 using NetStumbler sends Probe Request frames in order to identify AP, and then launches the DoS attacks.

5.2 Results and Discussion

To evaluate the proposed system, we measured the following performance metrics:

1) Hit Ratio (HR)

Hit Ratio is defined as the percentage of attacks correctly detected by the system over the total number of attacks down.

2) False Positive Ratio (FPR)

False Positive Ratio is defined as the percentage of the false positives (incorrectly) detected over the total number of attacks detected.

The HR can be used as a measure to see how effectively the proposed system could detect the various attacks. The FPR can be used to determine how much incorrectly the proposed system is performed.

Figure 4 and 5 show the results of HR and FPR, as the number of attacks taken increases for the three kinds of attacks, respectively. In the figures, the non-zero FPR is observed because the gathering agent cannot collect all the transmitted packets and thus the detection engine cannot process them. Overall, it is shown in the figures that the proposed system can effectively detect all kinds of attacks with a high Hit Ratio and a low False Positive Ratio.

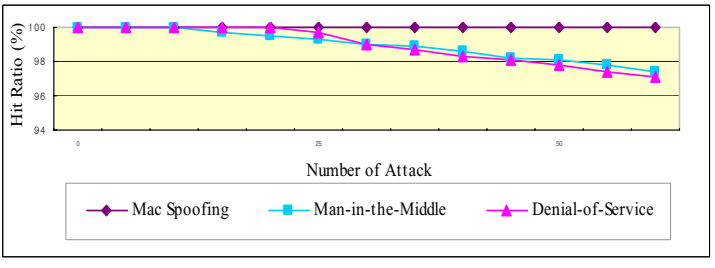


Fig. 4. Hit Ratio by the proposed scheme

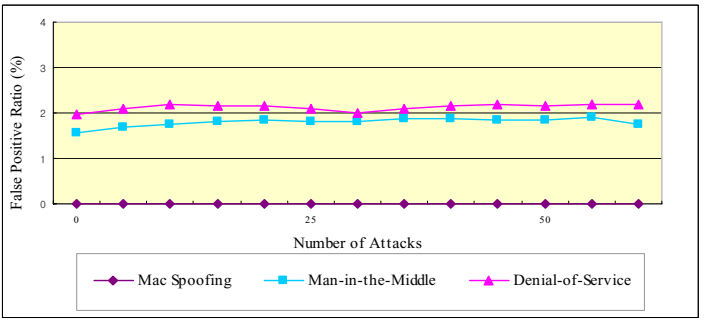


Fig. 5. False Positive Ratio by the proposed scheme

6 Conclusion

In this paper, we proposed a new integration scheme for intrusion detection for the WLAN system. The proposed scheme can be used to detect the various types of attacks such as the MAC spoofing, Man-In-the-Middle, and DoS attacks. From the experimental results, it is shown that the proposed scheme can effectively detect the promising attacks with high Hit Ratio and low False Positive Ratio.

Acknowledgment

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

1. IEEE 802.11 Standard, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1997.
2. Fluhrer, S., Mantin, I., and Shamir. A., Weakness in the Key scheduling Algorithm of RC4, Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography, August 2001.
3. Lim, Y., Schmoyer, T., Levine, J., and Henry, L., Wireless Intrusion Detection an Response, Proceedings of the IEEE Workshop on Information Assurance, 2003.
4. IEEE Draft P802.1X/D11. Standards for Local and Metropolitan Area Networks: Standard for Port based Network Access Control, March 2001.
5. Arbaugh, W., Shankar, N., Y. Wan, An initial Security Analysis of the IEEE 802.1X Standard, Technical Report, Department of Computer Science, University of Maryland, 2002.
6. Wright, J., Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection, Available from http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf
7. IEEE OUI and Company_ID Assignments, Available from http://Stanard.ieee.org/regauth/oui/oui.txt
8. Wright, J., Detecting Wireless LAN MAC Address Spoofing, Available from http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf
9. Hennie, H., Finite-State Models for Logical Machines, John Wiley & Son.

Topology-Aware Key Management Scheme for Secure Overlay Multicast

Jong-Hyuk Roh[1], Seunghun Jin[1], and Kyoon-Ha Lee[2]

[1] Information Security Research Division, ETRI, Korea
[2] Dept. of Computer Science and Engineering, Inha University, Korea

Abstract. Recently, the research focus of multicast has been put on overlay multicast. In overlay multicast, while the multicast routing, data replication and group management have been extensively studied, an important but less studied problem is security. In particular, adding confidentiality to overlay multicast is needed. To achieve confidentiality, data encryption keys are shared among the multicast group members. There is a need for key distribution scheme to solve the rekeying overhead. We introduce the key management solution called KTOM. And, we propose the use of periodic batch rekeying in KTOM.

1 Introduction

As expectations for the Internet to support multimedia applications grow, new services need to be deployed. One of them is the group communication service. There is more than a decade of important research and development efforts. However, the deployment of multicast routing in the Internet is far behind expectations, because of technical and marketing reasons [4]. Therefore, overlay multicast schemes have been proposed as the alternative group communication solution.

Many of the group communication services require data confidentiality for information protection and for charging purpose. Providing confidentiality in IP multicast has been extensively studied. However, they cannot be directly applied in overlay multicast, mainly due to the fundamental difference in data forwarding.

To offer data confidentiality in overlay multicast, there are two straightforward basic methods, host-to-host encryption and whole group encryption. However, these methods may not perform satisfactorily given a certain data rate and group dynamics. To solve these problems, SOT(Secure Overlay Tree) is proposed. Group member are divided into several clusters. Instead of sharing a group key among all members, members in a cluster share a *cluster key*. However, multicast packets need to be re-encrypted when the cross the boundary of clusters.

The performance of overlay multicast is lower than that of native multicast routing protocols because data forwarding at the end host is necessarily less efficient than using multicast routers in the backbone[6,7]. So, the delay of data forwarding is the critical issue in the security overlay multicast.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 733–742, 2006.

In this paper, we proposed the key management scheme called KTOM (Key Tree in Overlay Multicast) which uses the key tree mechanism to reduce the rekey overhead. And, we propose the use of periodic batch rekeying in KTOM.

2 Related Works

2.1 Basic Schemes

Host-to-Host Encryption. Each pair of peers that are neighbors in the multicast distribution tree shares a symmetric key. Upon receiving a packet from a parent node, a member decrypts the packet using the key shared with parent. Then it re-encrypts the packet using the key shared with child node and forwards to child node. By this scheme, when the membership changes, only its parent and children needed to be rekeyed. However, this scheme requires per-packet processing on every node re-encryption. Therefore, the nodal processing overhead is expected to be high for high-bandwidth applications [2].

Whole Group Encryption. Sender encrypts the data using a universal group key k_g. When a member receives the data packet, it simply delays the packet to its child nodes and decrypts the packet using a k_g. Therefore, this scheme has good performance in the data transmission. However, whenever one of the group member joins or leaves, the group key has to be changed. This incurs $O(N)$ re-key messages to all the existing N members, who are required to process the rekey messages. Clearly, the overhead of rekey is expected to be high for dynamic group [2].

2.2 SOT

Group members are divided into non-overlapping clusters of size m. Instead of sharing a group key among all members, members in a cluster share a cluster key. When the membership changes, rekey messages are only delivered within a cluster. Only $O(m)$ rekey messages are processed for each join/leave. SOT loosely maintains its cluster size by splitting and merging. Every cluster has a cluster leader, which manage the cluster for coordinating operations such as rekeying, merging and splitting. Packets are re-encypted only when they cross the boundary of clusters, and only take place at the ingress and egress nodes of a cluster. In other words, SOT uses "whole cluster encryption" within clusters and "host-to-host encryption" between clusters [2]. Either host-to-host encryption or whole group encryption may not perform satisfactorily given a certain data rate and group dynamics. SOT reduces the disadvantages in the two basic schemes using a hybrid scheme where the group members are divided into clusters. However, SOT has worse performance than whole group encryption in the multicast message transmission, because messages are re-encrypted when they cross the boundary of clusters. And the rekey overhead of SOT is more than host-to-host encryption.

3 Key Tree in Overlay Multicast

In this section, we describe KTOM scheme in detail. To quickly transmit the multicast data, KTOM uses a single group key. And to reduce the rekeying overhead, the key tree mechanism is employed in KTOM.

3.1 Key Tree

Logical key hierarchical (LKH) is often used to offer data confidentiality in IP multicast [1,5]. In LKH, one universal group key is used to transmit multicast data as whole group encryption. The main purpose of LKH is to reduce rekeying overhead. However, LKH cannot be directly applied in overlay multicast, because there is fundamental difference between IP multicast and overlay multicast in the data transmission [2,3]. In IP multicast, all interior nodes are routers and all end hosts are at the leaf positions. In contrast, in overlay multicast end hosts take all interior and leaf positions. Nevertheless, we employ the key tree in the secure overlay multicast.

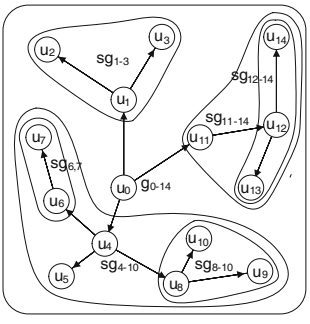


Fig. 1. Subgroup in overlay multicast

Fig. 1 shows the subgroup in overlay multicast. Group members are divided into subgroups to build the key tree. Subgroups are composed hierarchically. This rule is that if the node has one or more child node, it makes the subgroup that includes all its children and itself. In Fig. 1, the sender u_0 that send multicast data has child nodes, u_1, u_4, and u_{11}. Then, group g_{0-14} that includes all group members is generated. Each u_1, u_4, and u_{11} has the child node, then sg_{1-3}, sg_{4-10}, and sg_{11-14} is generated. Also, each u_6, u_8, and u_{12} has the child node, subgroups $sg_{6,7}$, sg_{8-10}, and sg_{12-14} are generated in the subgroup.

KTOM has a trusted key server responsible for generating and securely distributing keys to users in the group. And the key server manages the key tree that changed whenever the user joins or leaves. Fig. 2 shows the key tree that includes subgroups and members in Fig. 1. The key tree is composed with two

types of nodes: u-nodes representing users and k-nodes representing keys. There are three types of keys. The first type is a *group key*, used to encrypt/decrypt multicast data; the second type is a *subgroup key*, used to encrypt/decrypt other keys instead of the actual data; the last type is the *individual key*. Each member holds the keys along the path from u-node itself all the way to the root. Therefore, for the case of user u_6, u_6 holds k_6, $k_{6,7}$, k_{4-10}, and k_{0-14}. Each subtree in the entire key tree is a subgroup and each member is assigned to more than one subgroup. For example, member u_6 belongs to groups $sg_{6,7}$, sg_{4-10}, and g_{0-14}.

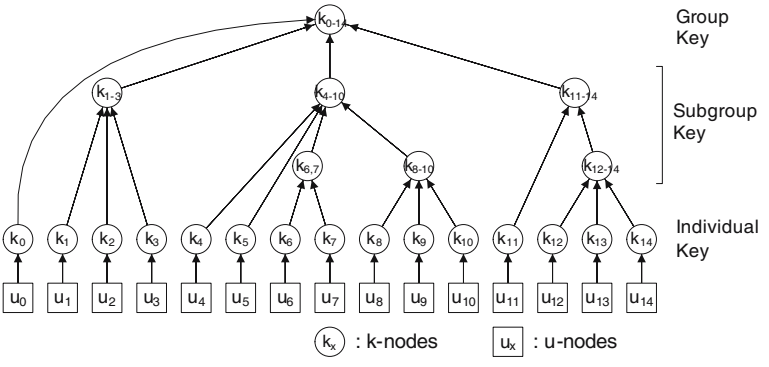


Fig. 2. Key tree

3.2 Member Joining

A new member u_x contacts the key server s to join the multicast group. For a join request from user u_x, server s performs the authentication and authorization process. If the join request is granted, the individual key k_x of u_x is generated and is shared by u_x and s.

A new member u_x finds a node that called the joining point in the overlay multicast tree. u_x first contacts the root of the tree, chooses the best node among the root's children, and repeat this top-down-process until it finds an appropriate parent. After finding the joining point, the key tree is modified. The modifying method of key tree is different according to the position of joining point.

The Joining Point Is Not the Leaf Node. If a new member u_x is attached to the root node or the interior node, u_x joins the existing subgroup that is composed with joining point and its children. According to this situation, server s modifies the key tree. Server s creates a new u-node for user u_x and a new k-node for its individual key k_x. The new k-node is attached to the k-node of joining subgroup. And, to guarantee backward secrecy, all keys along the path from joining point to the root node need to be changed.

For example, if a new member u_x is attached to u_1, k-node k_x is attached to k-node k_{1-3} in the key tree. To guarantee backward secrecy, the key of this

k-node is changed to $k_{1-3,x}$. Moreover, the group at the root is changed from k_{0-14} to $k_{0-14,x}$. User u_4, ..., u_{14} only need the new group key $k_{0-14,x}$. User u_1, u_2, u_3 and u_x need not only new group key but also the new subgroup key $k_{1-3,x}$. Server s creates and sends the following three rekey messages:

$$s \rightarrow u_4, ..., u_{14} : [k_{0-14,x}]_{k_{0-14}}$$
$$s \rightarrow u_1, u_2, u_3 : [k_{0-14,x}, k_{1-3,x}]_{k_{1-3}}$$
$$s \rightarrow u_x : [k_{0-14,x}, k_{1-3,x}]_{k_x}$$

Note the first message. There is no single key that is shared only by u_4, ..., u_{14}. The old group key k_{0-14} can be used to encrypt the new group key because u_x does not know this key k_{0-14}. At the second message, subgroup key k_{1-3} that is shared only by u_1, u_2, and u_3 is used for encryption.

The Joining Point Is the Leaf Node. If a new member u_x is attached to leaf node, the new subgroup that includes u_x and joining point is generated. Server s creates not only a new u-node and a new k-node for u_x, but also new k-node for new subgroup. The k-node for u_x is attached to the k-node for new subgroup.

For example, a new member u_x is attached to u_3, server s creates the new subgroup $sg_{3,x}$ and the k-node $k_{3,x}$. The k-node $k_{3,x}$ is attached to k_{1-3} and k_x is attached to $k_{3,x}$. To guarantee backward secrecy, key k_{0-14} and k_{1-3} must be changed. Server s creates and sends the following four rekey messages:

$$s \rightarrow u_4, ..., u_{14} : [k_{0-14,x}]_{k_{0-14}}$$
$$s \rightarrow u_1, u_2 : [k_{0-14,x}, k_{1-3,x}]_{k_{1-3}}$$
$$s \rightarrow u_3 : [k_{0-14,x}, k_{1-3,x}, k_{3,x}]_{k_3}$$
$$s \rightarrow u_x : [k_{0-14,x}, k_{1-3,x}, k_{3,x}]_{k_x}$$

3.3 Member Leaving

After granting a leave request from user u_x, the data transmission tree and key tree are updated. The modifying method of key tree is different according to the position of u_x.

The Departing Node Is Not the Leaf Node. If the departing user u_x is the root node or the interior node in the data transmission tree, the existing user u_y among the child nodes of u_x replaces u_x to transmit multicast data. In the overlay multicast, to reduce the transmission overhead of each node, there is the maximum number of children node, called maximum degree d_{max}. When u_y replaces u_x, children of u_x become the children of u_y. If the sum of new children and existing children exceeds the maximum degree d_{max}, the user u_z among the existing children of u_y replaces the old position of u_y. This process can be repeated.

For example, the maximum degree d_{max} in the overlay multicast of Fig. 1 is 3. When the departing user is u_4, let's assume that the case u_8 replaces u_4 is the optimal choice to provide the maximal throughput. In this case, the sum of children of u_8 exceeds d_{max}. Therefore, u_9 replaces the old position of u_8. (See Fig. 3.)

Server s modifies the key tree according to the above situation. The subgroup sg_{5-10} takes the place of sg_{4-10} and the $sg_{9,10}$ takes the place of sg_{8-10}. To guarantee forward secrecy, the universal group key k_{0-14} must be changed to $k_{0-3,5-14}$. And, the old k-node k_{4-10} and k_{8-10} is changed to the new k-node k_{5-10} and $k_{9,10}$. Server s creates and sends the following five rekey messages:

$$
\begin{aligned}
&s \rightarrow u_1, u_2, u_3 : [k_{0-3,5-14}]_{k_{1-3}} \\
&s \rightarrow u_{11}, ..., u_{14} : [k_{0-3,5-14}]_{k_{11-14}} \\
&s \rightarrow u_5 : [k_{0-3,5-14}, k_{5-10}]_{k_5} \\
&s \rightarrow u_6, u_7 : [k_{0-3,5-14}, k_{5-10}]_{k_{6,7}} \\
&s \rightarrow u_8, u_9, u_{10} : [k_{0-3,5-14}, k_{5-10}, [k_{9,10}]_{k_9}, [k_{9,10}]_{k_{10}}]_{k_{8-10}}
\end{aligned}
$$

When the rekey messages are encrypted, the key k_{0-14} and k_{4-10} must not be used. Because the departing node u_4 knows that keys.

The Departing Node Is the Leaf Node. In the case that the departing user u_x is the leaf node in the data transmission tree, the process is simple. The only user u_x is removed from the transmission tree. According to this situation, server s modifies the key tree. If u_x has sibling node, the u-node and k-node of u_x are only removed from the key tree. Unless, the subgroup includes u_x and its parent node is removed. In the key tree, u-node and k-node of u_x and k-node of subgroup is removed from the key tree. And, the k-node of u_x' parent is attached to the k-node of upper subgroup.

For example, if the departing user is u_2, only u-node and k-node of u_2 are removed. If the departing user is u_7, the subgroup $sg_{6,7}$ is removed. In the key tree, u-node and k-node of u_7 and k-node of $sg_{6,7}$ are removed. In the case that u_7 leaves the group, to guarantee forward secrecy, the group key k_{0-14} and subgroup key k_{4-10} must be changed. Server s creates and sends the following five rekey messages:

$$
\begin{aligned}
&s \rightarrow u_1, u_2, u_3 : [k_{0-6,8-14}]_{k_{1-3}} \\
&s \rightarrow u_{11}, ..., u_{14} : [k_{0-6,8-14}]_{k_{11-14}} \\
&s \rightarrow u_5 : [k_{0-6,8-14}, k_{5-6,8-10}]_{k_5} \\
&s \rightarrow u_8, u_9, u_{10} : [k_{0-6,8-14}, k_{5-6,8-10}]_{k_{8-10}} \\
&s \rightarrow u_6 : [k_{0-6,8-14}, k_{5-6,8-10}]_{k_6}
\end{aligned}
$$

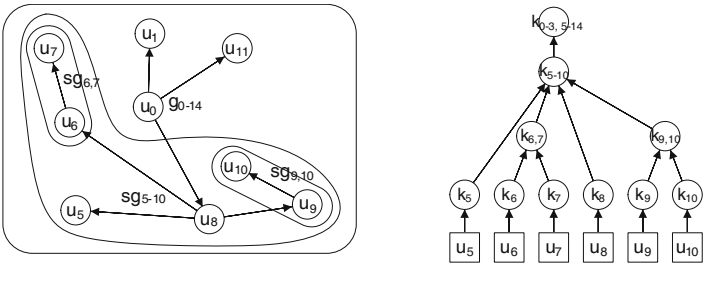


Fig. 3. Member leaving

4 Batch Rekeying

Ideally, a departed user should be expelled from the group, and a new user be accepted to the group, as early as possible. Thus, the key server should rekey immediately after receiving a join or leave request. This is called individual rekeying. However, individual rekeying has two problems: inefficiency and an out-of-sync problem between keys and data. To solve the problems, batch rekeying is proposed. In batch rekeying, the key server waits for a period of time, called a rekey interval, collects all the join and leave requests during the interval, generates new keys, constructs a rekey message and multicasts the rekey message [8]. We apply the batch rekeying to whole group encryption, SOT, and KTOM. In host-to-host encryption, batch rekeying is not necessary, because the group key is not used. In whole group encryption, the key server collects join and leave requests during a rekey interval. If there is a leave request, the key server cost (the number of encryption) is the number of group member. In SOT, each cluster leader individually manages the batch rekeying. If there is a leave request in cluster, the cluster leader cost is the number of cluster member.

4.1 KTOM Marking Algorithm

In [8], the marking algorithm is proposed for the key server to process a batch of requests. The key server modifies the key tree to satisfy the leave and join requests. The u-nodes for departed users are removed or replaced by u-nodes for newly joined users. This algorithm manages the balance of key tree. However, KTOM can not use this algorithm, because the position of joining or leaving members is decided according to the network topology. In KTOM, the following algorithm is used.

1. During interval, leave node is marked LEAVE.
2. During interval, joining member find one more joining point, evaluate the cost value, and send the value to key server.
3. At the end of interval, the key server modifies the key tree according to the following rules.
   (a) Replaces leave node by join node that found this location as the joining point. If there are one more candidates, the member has the smallest cost value is chosen.
   (b) All joining points are listed from the root of key tree. The member has the smallest cost value between itself and joining point is chosen. And this member is attached to the joining point.

5 Performance Evaluation

In our simulation, we compare the performance of KTOM with the two basic schemes and SOT. The simulation parameters are listed in Table 1.

Table 1. Parameters used in simulation

Parameter	Value
Group size	$100 \sim 1000$
Degree of multicast tree	4
Link delay	$1 \sim 3$ ms
Data packet size	10000 bytes
Cluster size in SOT scheme	50, 100, 200
Delay of encryption/decryption	0.2 ms
Delay of Key Agreement	1 ms

Fig. 4 shows the average time of multicast data transmission for different group sizes. This elapsed time is from the sender encrypts the message to all receivers decrypt the message. KTOM and whole group encryption show the better performance than SOT and host-to-host encryption, because the re-encryption is not needed in two schemes.

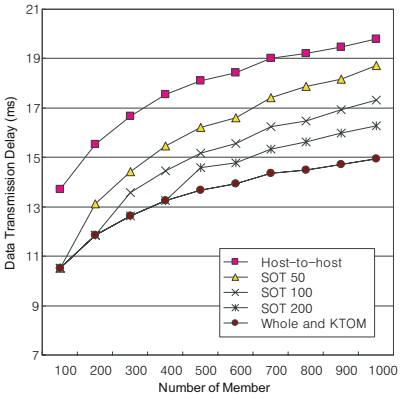


Fig. 4. Data transmission time

The average time of join processing for different group sizes is shown in Fig. 5. As the group size increases, the processing time of KTOM and whole group encryption increases. Because the universal group key is used, the entire group member need rekey process in the two schemes. However, the processing time of host-to-host encryption does not increases and after cluster split, SOT scheme does not increases, because the members need to be rekeyed are limited.

Fig. 5 shows the leave processing time. The average leave time of KTOM is slightly higher than the average join time. The average leave time of SOT scheme has less good performance than the join processing time. In SOT, when member leaving, the cluster leader generates the new cluster key k, it multicast k within the cluster along the overlay tree using host-to-host encryption, i.e., members re-encrypt k using neighbor keys before forwarding it [2]. Therefore, when the

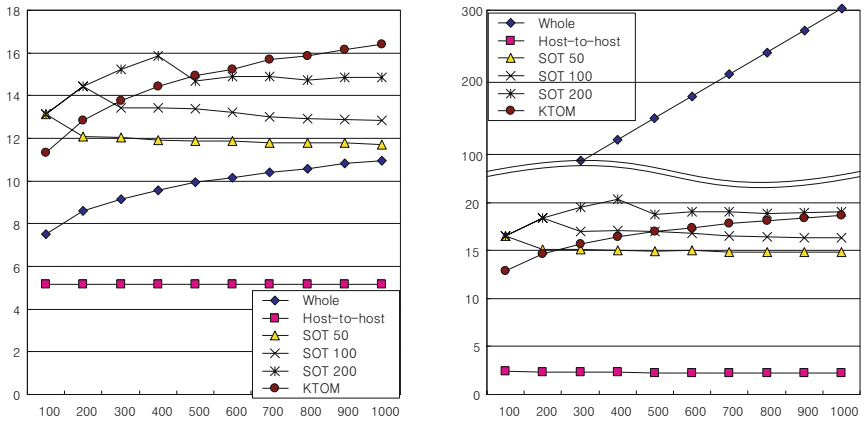


Fig. 5. Join/Leave processing time

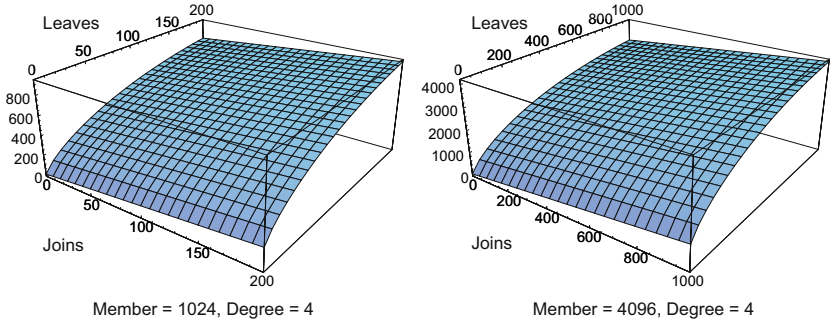


Fig. 6. Batch Rekeying

group size is small, KTOM has better performance than SOT. However, KTOM continuously increases, while SOT scheme does not increase after cluster split. In the whole group encryption, this incurs $O(N)$ rekey messages to all the group N members.

To summarize, in the data transmission, whole group encryption and KTOM scheme are significantly faster. In the join/leave processing, host-to-host encryption and SOT scheme is better. The choice of best key management scheme depends on the application needs: minimizing rekeying latency or minimizing data multicasting latency. However, the delay of data forwarding is the critical weakness of the overlay multicast. Therefore, we believe that minimizing data forwarding latency is better choice in the secure overlay multicast.

Fig. 6 shows the server cost in batch rekeying. During the rekey interval, the number of encryption in key server is evaluated. Fig. 6 shows the comparison for the number of member 1024 and 4096. We observe that KTOM is not better than whole group encryption when the number of joins and leaves is large and the number of leaves is zero.

6 Conclusion

In this paper, we describe a protocol called KTOM to provide data security in overlay multicast. KTOM is based on the key graph of secure group communications. KTOM uses a whole group key to reduce the delay of multicast data transmission. And, to reduce the rekeying overhead, KTOM employs the key tree mechanism. We compare the performance of KTOM with the SOT scheme and two basic schemes. According to the simulation results, KTOM can achieve much better performance than other schemes in the data transmission. And, key tree mechanism reduces the rekeying overhead. And, we propose the use of periodic batch rekeying in KTOM.

References

1. C. K. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Transactions on Networking, vol. 8, pp. 16-29, Feb. 2000.
2. W.-P. K. Yiu and S.-H. G. Chan, "SOT: Secure Overlay Tree for Application Layer Multicast," IEEE International Conference on Communications, vol. 3, Jun. 2004.
3. A. Ganjam and H. Zhang, "Internet multicast video delivery," in Proceedings of the IEEE, vol. 93, pp. 159-170, Jan. 2005.
4. A. El-Sayed, V. Roca, and L. Mathy, "A Survey of Proposals for an Alternative Group Communication Service," IEEE Network, vol. 17, pp. 46-51, Jan.-Feb. 2003.
5. K. Chan and S.-H. G. Chan, "Key Management Approaches to Offer Data Confidentiality for Secure Multicast," IEEE Network, vol. 17, pp. 30-39, Sep.-Oct. 2003.
6. P. Francis, "Yoid: Extending the Internet Multicast Architecture," Technical Report, ACIRI, Apr. 2000.
7. B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," INFOCOM, vol. 3, pp. 1366-1375, Jun. 2003.
8. Xiaozhou Steve Li, Yang Richard Yang, Mohamed G. Gouda, Simon S. Lam, "Batch Rekeying for Secure Group Communications," in Proceedings of the tenth international World Wide Web conference on World Wide Web, May. 2001.

Password-Based User Authentication Protocol for Mobile Environment*

Sung-Won Moon[1], Young-Gab Kim[2],
Chang-Joo Moon[3], and Doo-Kwon Baik[2,**]

[1] Mobile handset R&D Center, Mobile Communications Company, LG Electronics,
219-24, Kasan-dong, Kumchon-gu, Seoul, 153-801, Korea
kdunkman@hanafos.com
[2] Software System Lab. Dept.of Computer Science & Engineering,
Korea University 1, 5-ga, Anam-dong, Seongbuk-gu, Seoul, 136-701, Korea
{always, baikdk}@korea.ac.kr
[3] Department of Computer Science, Konkuk University,
322 Danwol-dong, Chungju-si, Chungcheongbuk-do,380-701, Korea
cjmoon@kku.ac.kr

Abstract. As mobile technologies evolve, mobile services tend to continuously expand and diversify. Therefore, developing security services appropriate for mobile environments is indispensable. This paper concentrates on how password-based user authentication protocols are applied to mobile environment, proposing the Password-based Authentication using Group Servers (PAGS) protocol. This protocol is able to provide authentication services relevant to mobile equipments to reduce complicated client processes in existing protocols. PAGS has the same security as protocols in [4,9], however this protocol is more appropriate for mobile equipments.

1 Introduction

As mobile technologies evolve, mobile services continuously expand and diversify, e.g. Internet Banking systems, mobile network games. Therefore, it is indispensable to provide security services relevant to mobile equipments. Many Internet service systems have introduced the password-authenticated key exchange protocol because of its convenience and inexpensiveness. This protocol can also operate within a mobile environment. Two conditions must be satisfied in order to apply password authentication protocol to mobile equipments: password security and proper usage of limited system resources.

A prominent weakness is use of a password as a cryptographic key. The password is vulnerable to an off-line dictionary attack, because of its regularity and small size. Numerous approaches with the aim of resolving the weakness, have been proposed since Bellovin and Merritt [1]'s research. Those can be categorized into two methods: The first method [5,6,7,11] allows a middle server to

* This work was supported by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.
** The corresponding author.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 743–753, 2006.

store a refined or pure password. However, an attacker compromising the server can easily obtain an original password with an exhaustive dictionary attack. As the second method, protocols dependent on multiple servers have been recently proposed, in order to solve the vulnerability of passwords in protocols relying only on a single server. These protocols never store password-related information on the servers [4,9,10,13]. Therefore, an attacker can not obtain the password from the system.

Although those protocols almost completely guarantee the security of users' passwords, it is impossible to apply them instantly to mobile equipment because the computation required of the client is excessive. In existing protocols, the client, based on the multiple server technique must communicate with n roaming-servers; therefore, the client must wait for n network channels and require n times exponent computation. The heavy computation on the client-side is fatal in applying the password-based protocol for mobile equipments because the system resource of the mobile phone is extremely restricted. In this paper, based on Ford & Kaliski's protocol, a new password-based authentication protocol called Password-based Authentication using Group Servers (PAGS) is presented, in order to increase the performance of the client. To achieve this objective, the agent server, representative of distributed roaming servers, is introduced.

This paper is organized into five chapters. In Chapter 2, two previous protocols are reviewed, and their client-side inefficiency is indicated. Chapter 3 describes the PAGS protocol, when solving the client's existing protocol inefficiency. In Chapter 4, the security and performance of PAGS, is analyzed. In Chapter 5, this paper finishes with a conclusion, and future work.

2 Related Work

Ford & Kaliski [4] proposed the first password-based authentication protocol based on multiple servers, to provide resilience against the middle server compromise caused from protocols based only on a single server.

Jablon [9] indicated the inefficiency of Ford & Kaliski's protocol, due to a secure channel, and developed a new protocol, which can operate without a secure channel. In this section, these two protocols are reviewed, and irrelevancy in applying these protocols to mobile environments is explained.

Ford & Kaliski Protocol consists of enrollment and roaming procedures. The enrollment procedure is executed when users request key-roaming services to providers for the first time. Information of users and the secrets for authenticating them are registered to the database of each roaming-server. In the roaming procedure, each roaming server authenticates users, and generates the hardened key from the password.

The protocol is executed on a subgroup whose order is q, where $p = (2q+1)$ is a secure prime number. The enrollment procedure is omitted because it is similar to the roaming procedure. Fig. 1. describes the roaming procedure of Ford & Kaliski's protocol. Initially, the client generates $W = f(PWD)$, where PWD is a

user's password, and f is Mask-Generation-Function. The client randomly selects a blinding factor a, and computes a roaming request message, $M = W^a modp$. The client transfers $M \| ID$ to roaming servers. The roaming servers search the previously registered b_i with ID, and compute a roaming-response message $C_i = (M)^{b_i} modp$, returning the C_i to the client. The client computes a K_i through $C_i^{1/a} modp$ and generates a strong key $K = KDF(K_1, \ldots, K_n)$. Finally, the client generates $V_i' = h(K, id_i)$, and transfers the V_i' to each roaming server. This procedure is completed, after all servers authenticate the user by comparing V_i stored during key-enrollment and V_i'. If V_i' and V_i are different to each other, the protocol will not provide any additional services because this means the authentication failed.

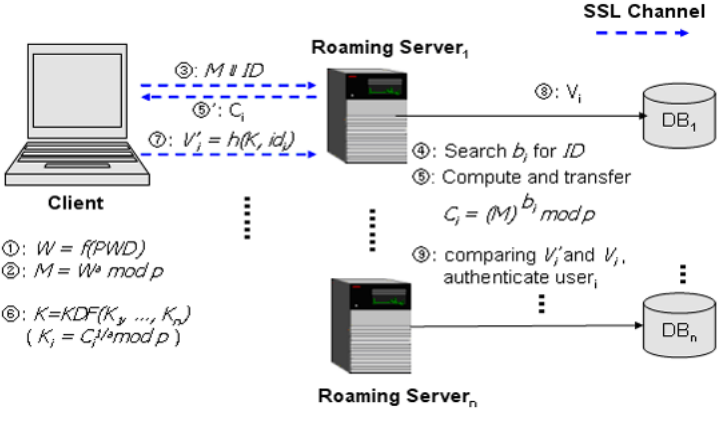


Fig. 1. Roaming procedure of Ford & Kaliski protocol

Ford & Kaliski's protocol can exchange the strong key K and authenticate the user with his password, resolving the vulnerability of the password stored in the single server. However, the messages in 3, 5', 7th procedures of Fig. 1 must be transferred through a SSL channel, because the protocol does not have any mechanism of authenticating each roaming server. It is almost impossible for a client to establish a SSL connection with n roaming servers, because establishing a SSL channel is a very complex process. Moreover, the client must be able to manage the certificate, which each roaming server issues; therefore, this protocol is executed on Public Key Infrastructure (PKI).

Jablon Protocol also uses enrollment and roaming processes. Fig. 2 presents the roaming process. The client computes g_{PWD} by computing $h(PWD)^{2r}$ and selects a random blinding factor a. The client creates a roaming-request message M by $g_{PWD}^a(modp)$ and then transfers $\{\mathbf{request}, ID, M\}$ to each roaming server. The roaming servers search $\{ID, b_i, U_K, proof_{PK_m}\}$, and compute a response message, $C_i = M^{b_i} modp$, transferring $\{\mathbf{reply}, C_i, U_K, proof_{PK_m}\}$ to the client. The client computes $K_i = C_i^{1/a} modp$ for each roaming server and creates a master-key, $K' = h(K_1 \| \ldots \| K_n) mod 2^j$. The client authenticates the roaming

servers by comparing $proof_{PK_m}$ and $h(K'\|g)$. If these two values are different, this protocol does not advance to the next step. If authentication for the servers is valid, the client application obtains a private key U through $D_{K_m}\{U_K\}$ and transmits $\{\mathbf{conform}, Q', \{Q'\}_U\}$ to each roaming server. The roaming servers authenticate the user through procedure 10.

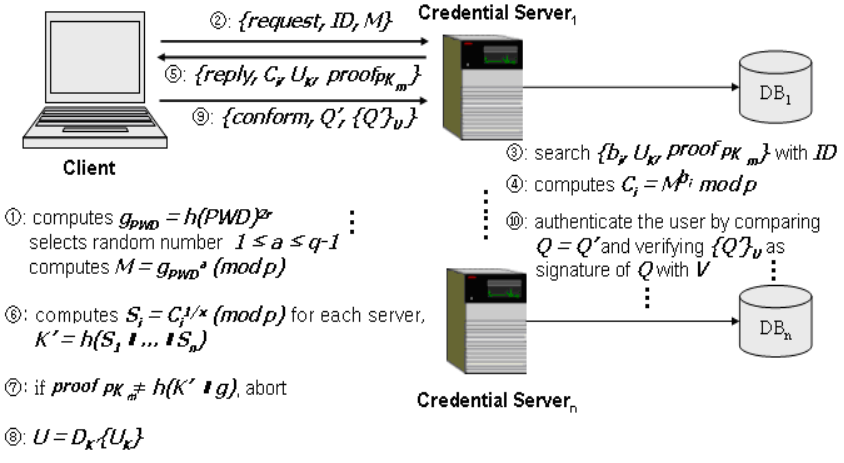


Fig. 2. Key roaming procedure of Jablon protocol

The Jablon protocol is similar to the Ford & Kaliski protocol, from the point where it also uses multiple servers to authenticate users and generate a strong key from the password; however this protocol does not use an additional secure channel. Instead, it uses $proof_{PK_m}$ to authenticate the server-side. As a result, this protocol could increase performance, in contrast to Ford & Kaliski's protocol. However, computation on the client is still extreme, because the Jablon protocol requires that the client communicate with multiple servers.

3 Password-Based Authentication Using Group Server

On the basis of the review in Chapter 2, a new key roaming protocol called PAGS is introduced in this chapter. The principal goal is to minimize complexity of the client-side process required. The entire model is first shown, and then the key enrollment and roaming procedure of the PAGS protocol is explained in detail.

Preliminary. The existing key-roaming protocols are not suitable for mobile equipment because the mobile node is required to conduct excessive computation. In this paper, an agent-server is introduced, in order to reduce the computation of the client. As a result, the client only communicates with the agent server. In addition, the agent server replaces some work the client had previously executed. In Fig 3, the differences between the old protocols and the PAGS protocol are presented.

The agent server receives a roaming-request message from the client instead of all the roaming servers. Similarly, the agent server receives a roaming-response message generated in each roaming server, instead of the client. After all, the agent server processes work of the client and roaming servers. In this model, the entire protocol's performance of the PAGS protocol depends on the agent server, therefore, a bottleneck on the agent server exists. However, it is more efficient to delegate client's work to the agent server, rather than let the client execute this work, which is inefficient due to restricted system resources.

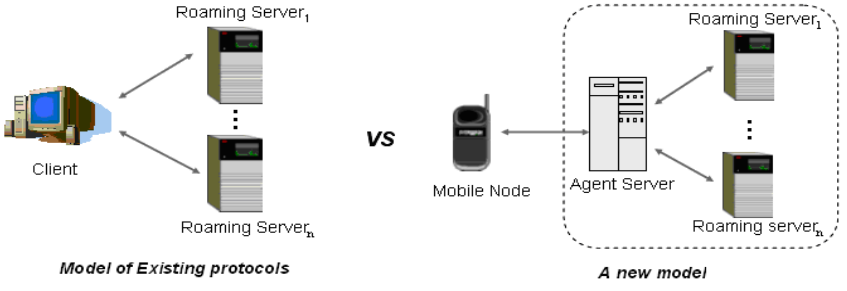


Fig. 3. Comparison between established protocols and PAGS

The PAGS protocol assumes three points; the first is that the communication between the agent server and roaming servers is executed through a secure channel. Therefore, the security between the client and the agent server can be concentrated on. Secondly, the agent server must have sufficient system resources to process all messages from the mobile client and roaming servers. Thirdly, it is assumed that the key enrollment procedure is executed securely in Ford & Kaliski's protocol, the Jablon protocol, and the PAGS protocol.

Table 1. Parameter for PAGS

$1 \leq a \leq q-1$	Blinding factor (random number which is selected by a user)
PWD	Password
$W = f(PWD)$	f is Mask-Generation-Function; e.g. $f(PWD) = g^{PWD} modp$
M	Roaming request information, $W^a modp$
$1 < b_i < q-1$	The random number selected by the ith roaming server. $(1 < i \leq n)$
C_i	Roaming response information, $(= f(PWD)^{ab_i} modp)$
$KDF(K_1, \ldots, K_n)$	Key Derivation Function
$h(K, id_i)$	One Way Function
V	User authentication information
g, q	Random number for mutual authentication
ID_{Agent}	Identifier for an agent server
$AuthServer$	Information to authenticate an agent server
C	Group response message

Table 1 presents the protocol parameters for the PAGS protocol. This protocol also executes on the subgroup whose order is q, where $p(= 2q + 1)$ is a secure

prime number, identical to Ford & Kaliski's protocol. Parameters for the agent server are also introduced.

Enrollment. First, for roaming services, the user who has never provided private information must register authentication and secret information with each roaming server. If a client first accesses the roaming server, the enrollment procedure is executed as shown in Fig. 4.

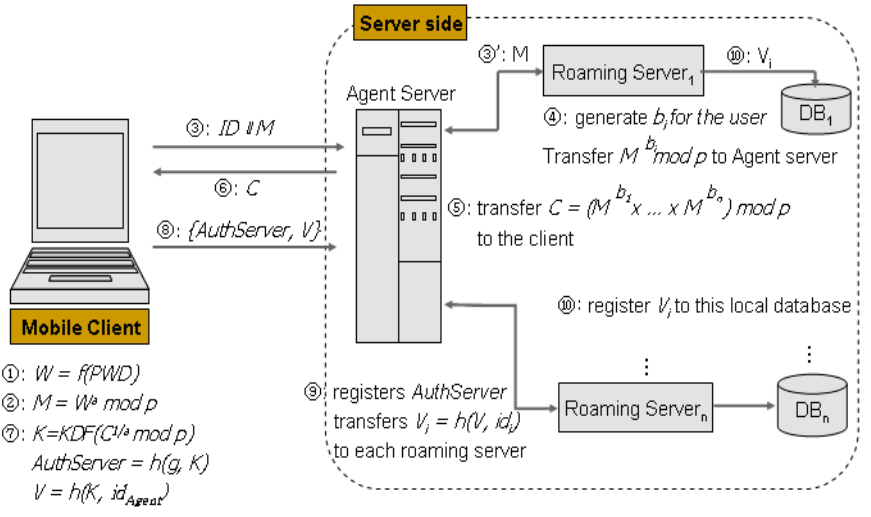


Fig. 4. Enrollment procedure of the PAGS protocol

1. The client receives ID and a password PWD from a user, generates W from the password PWD, using Mask-Generation-Function, f.
2. The client generates a blinding factor a, and computes a roaming request message, M, by computing $W^a modp$.
3. The client transfers $ID \| M$ to an agent server. The agent server transmits the roaming request message, M, to each roaming server.
4. The roaming servers generate their private secrets b_i, and register ID and b_i in their own databases. In addition, the roaming servers create a roaming response message C_i from M and b_i. The roaming server transmits the C_i to the agent server.
5. The agent server computes a group response message, C with the roaming response message C_i, coming from each roaming server using the following equations. **Group response message:** $C = (M^{b_1} \times \ldots \times M^{b_n}) modp$
6. The agent server transmits the group response message, C, to the client.
7. The client creates a strong key from group response message, C, transmitted from process 6. Similar to the following.
   $KDF(C^{1/a} modp) \rightarrow KDF\ \{(M^{b_1} \times \ldots \times M^{b_n})^{1/a} modp\}$
   $\rightarrow KDF\ \{(W^{ab_1} \times \ldots \times W^{ab_n})^{1/a} modp\}$
   $\rightarrow KDF\ \{(W^{b_1} \times \cdots \times W^{b_n})^{1/a} modp\}$

8. In addition, the client creates $AuthServer = h(g, K)$ to authenticate the agent server and $V = h(K, ID_{Agent})$ to validate the user during the roaming procedure.
9. The client transmits $AuthServer$ and V to the agent server.
10. The agent server registers $AuthServer$ into its local database, and transfers user authentication information $V_i = h(id_i, V)$ to each roaming server. This protocol is completed when each roaming registers V_i into their local database.

Through this procedure, the agent server stores information, which it can use to authenticate itself with the client application. Each roaming server registers the information used to authenticate the client application, namely the user.

Key Roaming. Fig. 5. describes the key-roaming procedure of the PAGS protocol. This procedure is used to authenticate whether the user is real, generating the strong key from the password. The roaming server finally passes secret information encrypted with the hardened key created through this procedure.

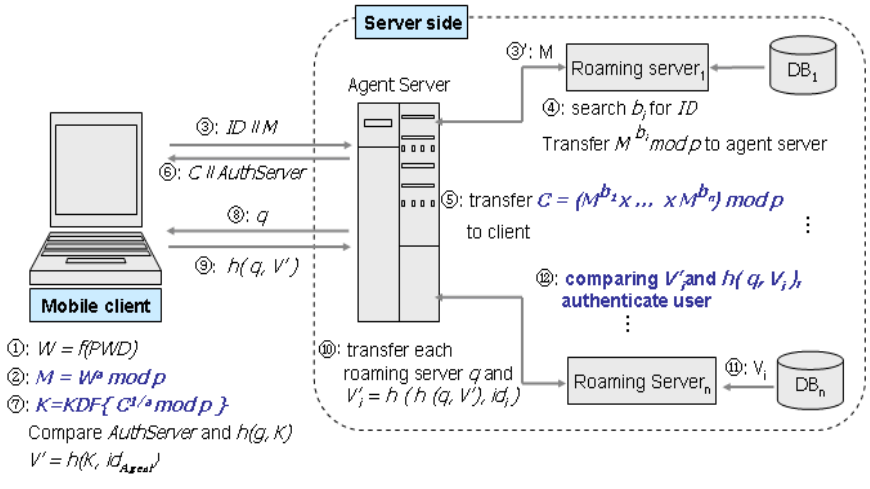


Fig. 5. Roaming procedure of PAGS protocol

The roaming procedure of the PAGS is as follows.

1. The beginning of the protocol is generating W from password, PWD, by using Mask-Generation-Function, f.
2. The client selects a random number as a blinding factor a, computing a roaming-request message $M = W^a mod p$.
3. The client transfers $M \| ID$ created in the second process to the agent server. The agent server, which received $M \| ID$ from the client, transmits this to each roaming server.

4. Each roaming server searches b_i from its database with the user identifier, *ID*, computing $C_i = M^{b_i} modp$. The roaming server transmits C_i to the agent server.
5. The agent server computes group-response message, C, with C_i from each roaming server using the following equation.
   Group response message: $C = (M^{b_1} \times \ldots \times M^{b_n}) modp$
6. The agent server transmits *AuthServer* and C generated in process 5 to the client.
7. The client creates a hardened key by computing KDF $(C^{1/a} modp)$, and generates *AuthServer'*. The client authenticates the agent server by comparing *AuthServer* and *AuthServer'*. If the value is different, this protocol does not proceed to the next step. Again, the client creates *V'* for authenticating the user by $h(K, ID_{Agent})$.
8. The agent server asks user-authentication information by transmitting challenge q to the client.
9. The client transmits $h(q, V')$ to the agent server.
10. The agent server computes $V_i' = h(h(q, V'), id_i)$ and transmits V_i' and q to the each roaming server.
11. ∼12. Each roaming server searches V_i registered in the enrollment procedure with the user's ID. Finally, this protocol is finished by authenticating the user by comparing V_i' and $h(q, V_i)$.

Through the process of Fig. 5, the PAGS protocol creates hardened key, K, and authenticates a user by comparing the authentication information created in the roaming procedure and that registered in the enrollment procedure. The most important characteristic of the PAGS protocol is that an agent server between a client and n roaming servers is introduced. The agent server does the work for the client so that this protocol can reduce client complexity.

4 Analysis of PAGS Protocol

Security Analysis. The security of the PAGS protocol is analyzed on the basis of several attack methods. First, consider a dictionary attack. The PAGS protocol does not store related information with a password. Therefore, there is no possibility of performing a dictionary attack because b_i is not related with a user's password. Consider the case that an attacker conducts a dictionary attack with a password dictionary and communication records. The M and C are password related information. However, these values are encrypted with a and roaming servers' b_i. The a is always a newly generated random number, therefore, the attacker cannot obtain a. The b_i is a secret stored in n roaming servers. This protocol is secure from dictionary attack as far as only one server of all roaming servers is secure.

In the case of a replay attack, where an attacker illegally progresses using an early exchanged message, all the messages exchanged in the PAGS protocol are always newly generated values because blinding factor, a, is always a newly created random number. Therefore, it is impossible to attack the PAGS protocol

with previous M and C. Although an attacker obtains V' from procedure 8 in Fig. 5, it is impossible to obtain the strong-key, K because V' is one-way functioned value of K and q. Consider a man in the middle attack. Ford and Kaliski's protocol used a secure channel to authenticate roaming servers. This protocol required a method where the agent server authenticates the client, because the PAGS did not use a secure channel. It is possible by comparing $AuthServer'$ computed from roaming procedure and previously registered $AuthServer$.

Efficiency Analysis. The PAGS protocol has the problem that a bottleneck exists on the agent server. But, the clients in other protocols are more seriously inefficient because they must also execute the agent server's work. Table 2 presents PAGS's increased client performance by a comparison with Ford & Kaliski and Jablon protocol.

First, the event of a round-trip is explained. The key enrollment procedure in the Ford & Kaliski's protocol requires a similar round trip with other protocols. However, this protocol requires creating SSL during the roaming procedure. As a result, the round trip increases, and there is also a serious overhead on the client. The Jablon protocol introduced a mechanism to authenticate a server-side without a SSL channel; therefore, the round trip is reduced rather than that of Ford & Kaliski's protocol.

Table 2. Efficiency comparison between PAGS and existing protocols

Element \ Protocol		Ford & Kaliski	Jablon	PAGS
Round	Enrollment	3	2	3
	Roaming	8	3	3
Required-Channel		n	n	1
Comp.	Hard Key	$K_1 = C_1^{1/a} \bmod p$ ... $K_n = C_n^{1/a} \bmod p$ $K = KDF(K_1, \ldots, K_n)$	$K_1 = C_1^{1/x} \bmod p$ ... $K_n = C_n^{1/x} \bmod p$ $K = h(K_1 \,\|\| \ldots \,\|\| K_n)$	$K = KDF(C^{1/a} \bmod p)$
		Exponent n	Exponent n	Exponent 1
	User auth.	Hash n	Signature n	Hash 2

Second, the channel number for which the client must wait is compared. In other protocols, the client must communicate with n roaming servers. This is very difficult because mobile nodes have restricted resources and short battery life. The PAGS protocol placed the agent server between the client and roaming servers, and relieved the client from serious complication. Although PAGS is similar with the Jablon protocol in the round trip, the client's performance was dramatically improved by the reduced work.

Third, the amount of computation required on the client-side is considered. Ford & Kaliski's protocol required n times exponent to create hardened key, K. Moreover, because this protocol requires a client to create a secure-channel

with each roaming server, the computation of the client becomes extreme. The Jablon protocol attempted to eliminate an additional secure channel so that no additional computation is required for creating the secure-channel. PAGS can generate strong key K with only 1 exponent.

Finally, the PAGS protocol also demonstrates advanced efficiency in computing user authentication information. Ford and Kaliski's protocol required an n times hash function to create user authentication information. The Jablon protocol requires n times attaching of a digital signature. However, the PAGS protocol requires only 2 times hash functions, because the client is not required to communicate with all roaming servers. Therefore, PAGS protocol can be expected to increase the client's performance.

5 Conclusion and Future Work

As mobile technologies develop, security-services for mobile environments also become increasingly important. In this paper, the PAGS protocol is proposed, based on Ford and Kaliski's protocol. It is possible that the client is emancipated from the burden that it has to communicate with n servers, by introducing an agent server. As a result, the process of the client becomes efficient. It is expected that this new protocol can efficiently operate in mobile environments where the bandwidth and system resources are restricted.

In future work a full theoretical treatment for validating the security of the PAGS protocol, is required, and a more specific structure for applying it to mobile environment is required to be designed.

References

1. S. M. Bellovin and M. Merritt, Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, Proceedings of the I.E.E.E. Symposium on Research in Security and Privacy, Oakland, May 1992.
2. S. Bellovin and M. Merritt, Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password-file compromise, ACM Conference on Computer and Communications Security, 1993.
3. D. Jablon, Strong password-only authenticated key exchange, ACM Computer Communications Review, October 1996.
4. W.Ford and B. Kaliski, Server-Assisted Generation of a Strong Secret from a Password, Proc. 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, June 14-16, 2000.
5. M. Bellare, D. Pointcheval, P. Rogaway, Authenticated Key Exchange Secure Against Dictionary Attacks, Eurocrypt, 2000.
6. L. Gong, T.M.A. Lomas, R.M. Needham, and J.H. Saltzer, Protecting Poorly Chosen Secrets from Guessing Attacks, IEEE Journal on Selected Areas in Communications, vol.11, no.5, June 1993, pp. 648-656.
7. R. Perlman and C. Kaufman, Secure Password-Based Protocol for Downloading a Private Key, Proc. 01999 Network and Distributed System Security Symposium, Internet Society, January 1999.

8. P. C. van Oorschot, M. J. Wiener, On Diffie-Hellman Key Agreement with Short Exponents, Proceedings of Eurocrypt '96, Springer-Verlag, May 1996.
9. David P. Jablon, Password Authentication Using Multiple Servers, The Cryptographers' Track at RSA Conference 2001 San Francisco, CA, USA, April 8-12, 2001.
10. P. Mackenzie, T. Shrimpton, and M. Jakobsson, Threshold Password-Authenticated Key Exchange, In M. Yung, editor, CRYPTO 2002, pages 385400, Springer-Verlag, 2002, LNCS no.2442.
11. P. Mackenzie, S. Patel and R. Swaminathan, Password-authenticated key exchange based on RSA, ASIACRYPT, 2000.
12. V. Boyko, P. MacKenzie, S. Patel, Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman, Eurocrypt, 2000
13. Mario Di Raimondo, Rosario Gennaro, Provably Secure Threshold Password-Authenticated Key Exchange Extended Abstract, Eurocrypt 2003, LNCS 2656, pp. 507-523, 2003.

SVM Based Packet Marking Technique for Traceback on Malicious DDoS Traffic

Hyung-Woo Lee

Div. of Computer, Information and Software, Hanshin University,
Osan, Gyunggi, 447-791, Korea
hwlee@hs.ac.kr
http://netsec.hs.ac.kr

Abstract. Distributed Denial-of-Service(*DDoS*) attack can be done by generating a large volume of traffic through spoofing the IP address of DoS attacker. The e-mail based attack is also similar with existing DDoS attack in network traffic status. In response to such attacks, IP traceback technology has been proposed. For example, the method identifies the source of a spoofed e-mail attack and restructures the path on the network through which the attacking packet has been transmitted. This study proposed an improved marking technique that identifies DDoS traffics with TTL information at routers by applying the SVM module for malicious traffic control and cope with DDoS attack packets efficiently. According to the result of experiments, the proposed technique reduced network load and improved filter/traceback performance.[1]

1 Introduction

The current TCP/IP system such as SMTP based e-mail system is vulnerable to *DoS (Denial of Service)* attacks such as TCP/UDP flooding[1], there have been researches on how to cope with hacking on networks and the Internet[2]. As for techniques to cope with hacking attacks, firewall systems that adopt access control are passive to hacking attacks. IDS(Intrusion Detection System) provides the functions of detecting and blocking abnormal traffic that has reached the victim system, so it is also passive to hacking.

Thus currently available technologies do not provide active functions to cope with hacking such as tracing and confirming the source of spam mail sender. It is because most DoS flooding like spam attacks are carried out by spoofing the IP address of the source system. Thus it is necessary to develop a technology to cope actively with such hacking attacks. Even if the trace-route technique is applied to identify the source address, the technique cannot identify and trace the actual address because the address included in DDoS(Distributed Denial of Service) is spoofed.

Methods of defeating hacking like DDoS attacks are largely divided into passive ones such as vaccines, intrusion detection and tolerance technology, and

[1] This work was supported by Korea Research Foundation Grant (KRF-2005-202-D00487).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 754–763, 2006.

active ones such as traceback of the origin of attacks. Active methods are again divided into proactive traceback and reactive traceback according to how to detect the origin of hacking attacks.

If a victim system is hacked, it identifies the spoofed source of the hacking attacks using the generated and collected traceback path information. PPM (probabilistic packet marking)[5] and iTrace(ICMP traceback)[7] are this type of traceback methods. A recently proposed SVM[3,6,8] method provides both a non-linear classification function for input data when a DDoS attack happens. The existing SVM method provides a control function for DDoS attack traffic but does not provide the function of trace back the source of the attack. It only provides a classification function for packets among the diverse packet transmissions, so *filters* the malicious packet for controling the network transmission on DDoS packets.

Thus this study proposes a technique to trace back the source IP of spoofed DDoS packets by combining the existing SVM method, which provide a filter and control function against DDoS attacks, with a traceback function. Therefore, a router performs the functions of *identifying/controlling traffic using the SVM technique*, and when a DDoS attack happens it sends packet to its next hop router by marking router's information on the header. Compared to existing traceback techniques, the proposed technique reduced management system / network load and improved traceback performance.

Chapter II reviewed the weaknesses of existing technologies for tracing back the source of hacking attacks and directions for improvement, and Chapter III reviewed the advantage of SVM-based filter/control technique for traceback. Chapter IV and V proposed a new packet marking technique that adopted a SVM technology to classify and efficiently trace back the source of DDoS attacks, and Chapter VI compared and evaluated the performance of the proposed technique.

2 Related Works

2.1 Traceback Mechanisms

Because an attacker can carry out fatal DDoS attacks to victim systems by controlling a large number of servers where attacking tools are installed, such a method can be abused by hackers who mean to disturb the Internet. Up to now, when hacking attacks occur in the Internet, they have been defeated passively using firewall, IDS, scanning and trusted OS-based system security, etc. In particular, existing methods cannot restrict or prevent an attempt at hacking itself, so they are often useless and powerless against attacks paralyzing the Internet. To solve such a problem, active hacking prevention methods were proposed.

Traceback : *an essential technology to cope with hacking and virus actively. Traceback technology traces back the source of hacking attacks real-time and resultantly suppresses hacking attacks fundamentally.*

Existing IP Traceback methods can be categorized as proactive or reactive tracing. Proactive tracing (such as packet marking and messaging) prepares information for tracing when packets are in transit. Reactive tracing starts tracing after an attack is detected.

In PPM mechanism[5], a router, an important component of a network, inserts information on packets transmitted through the router into IP packets in order to find the packet transmission route for spoofed packets.

That is, for packets transmitted through the Internet, a router routes them by checking packet header information centering on the IP layer. At that time, the router inserts information on the router address into a writable field of the IP header and sends the packet to the adjacent router.

2.2 E-Mail Sender IP Traceback for Preventing Spoofed Sender

Internet mail suffers from the fact that much unwanted mail is sent using spoofed addresses. "spoofed" in this case means the address is used without the permission of the domain owner. So diverse approach were proposed for preventing this attack.

TCP Attacks on e-mail protocol is designed to be used in conjunction with SMTP over TCP. A sufficiently resourceful attacker might be able to MTA Authentication Records in DNS send TCP packets with forged from-addresses, and thus execute an entire SMTP session that appears to come from somewhere other than its true origin. Such an attack requires guessing what TCP sequence numbers an SMTP server will use. Attacks of this sort can be ameliorated if IP gateways refuse to forward packets when the source address is clearly bogus.

Forged Resent-From Attack chooses a purported responsible address from one of a number of message headers, and then uses that address for validation. A message with a true Resent-From header (for example), but a forged From header will be accepted. Since many MUAs do not display all of the headers of received messages, the message will appear to be forged when displayed. In order to avoid this attack, MUAs will need to start displaying at least the header that was verified. For advanced approach, e-mail attack prevention mechanisms such as IP traceback are required to lessen the e-mail spam sending transactions.

3 SVM Mechanism

3.1 SVM for Classification

When it formulates the boundary between classes, it determine whether the input is useless or not in order to find optimal boundary. For selected input, we call it *Support Vector*. So. it makes optimal boundary between classes. The goal of my research was, primarily, how well packet classification module could be applied to the Traceback System for automatically separate IDS data into normal or anomalous distributions[3,8].

SVM: *Inputs are converted into a high dimensional feature spaces, which enable to separate non-linear separable spaces into a proper classes.*

3.2 SVM for Traceback

In this section we review some basic ideas of support vector machines. The detailed about SVM for classification and non-linear estimation can be found in [3,10,11].

Given the training data set $\{(x_i, d_i)\}_{i=1}^{l}$, with input data $x_i \in R^N$ and corresponding binary class labels $d_i \in \{-1, 1\}$, the SVM classification formulation starts from the following assumption. The class can be represented by subset $d_i = 1$ and $d_i = -1$ are linearly separable $^{\exists}\omega \in R^N, b \in R$ such that

$$\begin{cases} \omega^T x_i + b > 0 \text{ for } d_i = +1 \\ \omega^T x_i + b < 0 \text{ for } d_i = -1 \end{cases} \tag{1}$$

The goal of SVM is to find an optimal hyperplane for which the margin of separation ρ is maximized. The margin of separation ρ is defined by the separation between the separation hyperplane and the closest data point. If the optimal hyperplane is defined by $\omega^T + b_0 = 0$, then the function $g(x) = \omega^T x + b_0$ gives a measure of the distance from x to the optimal hyperplane.

Support Vectors are defined by data points $x^{(s)}$ that lie the closest to the decision surface. For a support vector $x^{(s)}$ and the canonical optimal hyperplane g, we have

$$r = \frac{g(x^s)}{||\omega_0||} = \begin{cases} +1/||\omega_0|| \text{ for } d^{(s)} = +1 \\ -1/||\omega_0|| \text{ for } d^{(s)} = -1 \end{cases} \tag{2}$$

The margin of separation $\rho \propto \frac{1}{||\omega_0||}$. Thus, $||\omega_0||$ should be minimal to achieve the maximal separation margin. Mathematical formation for finding the canonical optimal separation hyperplane given the training data set $\{(x_i, d_i)\}_{i=1}^{l}$, solve the following quadratic problem

$$\begin{cases} \text{minimize } \ \tau(\omega, \xi) = \frac{1}{2}||\omega||^2 + c\sum_{i=1}^{l} \xi_i \\ \text{subject to } d_i(\omega^T x_i + b) \geq 1 - \xi_i \text{ for } \xi_i \geq 0, i = 1, ..., l \end{cases} \tag{3}$$

Note that the global minimum of above problem must exist, because $\phi(\omega) = \frac{1}{2}||\omega||^2$ is convex in ω and the constrains are linear in ω and b. This constrained optimization problem is dealt with by introducing Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian function given by

$$L(\omega, b, \xi; \alpha, \nu) = \tau(\omega, \xi) - \sum_{i=1}^{l} \alpha_i [d_i(\omega_i^T x_i + b) - 1 + \xi_k] - \sum_{i=1}^{l} \nu_i \xi_i \tag{4}$$

$$\frac{\partial L}{\partial \omega} = 0 \iff \omega - \sum_{i=1}^{l} \alpha_i d_i x_i, \ \frac{\partial L}{\partial b} = 0 \iff \omega - \sum_{i=1}^{l} \alpha_i d_i = 0, \ \frac{\partial L}{\partial \xi_k} = 0 \tag{5}$$

The solution vector thus has an expansion in terms of a subset of the training patterns, namely those patterns whose α_i is non-zero, called *Support Vectors*. By the complementarity condition we have

$$\alpha_i [d_i(\omega^T x_i + b) - 1] = 0 \quad for \quad i = 1, ..., N \tag{6}$$

The hyperplane decision function can thus be written as

$$f(x) = sgn(\sum_{i=1}^{l} y_i \alpha_i \cdot (x \cdot x_i) + b) \tag{7}$$

where b is computed using (6). To construct SVM, the optimal hyperplane algorithm have to be argumented by a method for computing dot products in feature spaces non-linearly related to input space. The basic idea is to map the data into some other dot product space (called the feature space) F via a nonlinear map ϕ and to perform the above linear algorithm in F, where $x_i \in R^N, d_i \in \{+1, -1\}$ processes the data with $\phi : R^N \rightarrow F, x \mapsto \phi(x)$where $l \ll dimension(F)$, $(\phi(x_i), \phi(x_j)) = K(x_i, x_j)$. And $K(x_i, x_j)$ can be easily computed on the input space.

3.3 SVM Based DDoS Traffic Filtering/Control Mechanism

Some security mechanisms for securing wire and wireless networks have been proposed and can be classified into two types-intrusion(attack) prevention and intrusion(attack) detection[6]. Intrusion prevention implies developing new secured protocols for wire/wireless networks or modifying the logic of existing protocols to enhance their security. Most of the current work using encryption or authentication mechanisms belongs to this group[3].

In this paper, we focus on the security issues of distributed open networks and propose a Support Vector Machine (SVM) based packet marking and traceback system, which is suitable for real-time DDoS malicious traffic prevention in router as Fig. 1.

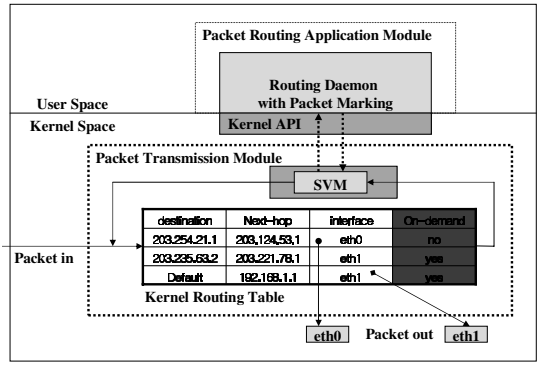


Fig. 1. SVM module in Router for Traffic Filtering/Control

From the viewpoint of a router composing the network, *a hacking attack on the Internet is a kind of congestion.* Thus coping with hacking attacks may be approached from *congestion control* between end systems and relevant technologies. A DDoS attack transmits a large volume of traffic from one or more source hosts to a target host, there should be researches on how to identify and block DDoS traffic in order to cope with hacking attacks on the Internet.

In Fig. 1, we propose a advanced router with packet marking mechanism, which is to control DDoS traffic on it for *aggregate-based congestion filtering / control* with SVM.

SVM based Traffic Control: *If traffic shows congestion exceeding a specific bandwidth based on the characteristic of DDoS attack network traffic, the SVM based control module judges based on congestion signature that a hacking attack has happened and working with a filtering module, provides a function to block the transmission of traffic corresponding to the DDoS attack.*

4 SVM Based Packet Marking Against DDoS Attacks

4.1 Traceback Structure Using SVM

The method proposed in this study does not sample and mark at a fixed probability of p but mark packets when abnormal traffic is found by a SVM module. Of course, unlike the method used in existing marking techniques, when abnormal traffic is found by SVM module, the router can recognize the characteristic of hacking traffic included in the message, performs marking with two router addresses and sent the message to the target system.

The Fig. 2 shows the structure of *SVM-based filter/control* when a router is congested. As in the figure, the process of packet marking is integrated with a SVM based filter/control module. The SVM module confirms a malicious DDoS attack and sends a marked packet to its adjacent next hop router on the network path. In the proposed structure, a router checks the traffic bandwidth of a received packet and if the bandwidth exceeds a certain level the router judges whether it is a congestion signature corresponding to an attack pattern.

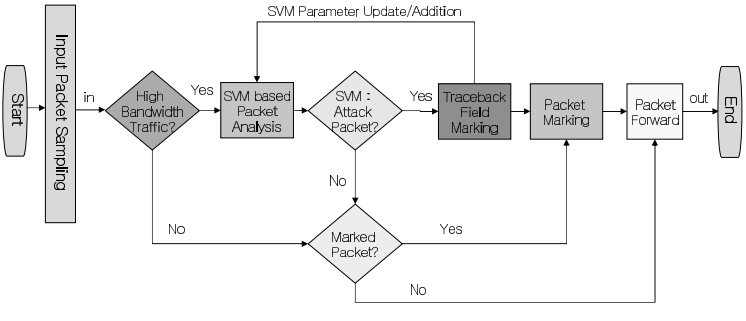


Fig. 2. SVM-based traffic identification/control mechanism

4.2 Traceback IP Marking with SVM

(1) Packet Header Marking Field M_x. Let's say A_x is the IP address of R_x, P_x is IP packet arrived at R_x, and M_x is 24 bits on the header of P_x in which marking information can be stored. In packet P_x, M_x is composed of 8-bit *TOS(type of service)* field, and 16-bit *ID field*. TOS field has been defined is not used currently. Thus the use of TOS field does not affect the entire network. In TOS field, the first 3 bits are priority bits, and next three bits are minimum delay, maximum performance and reliability fields but not used currently.

Recently, however, TOS field is redefined as Differentiated Service field(DS field) according to *RFC2474*, in which only the first 6 bits are used. Thus this study defines the unused 2 bits out of TOS field as *TM(traceback marking flag)* and *CF(congestion flag)*. Particularly for CF, RFC2474 defines it as 1 if the network is congested.

(2) Marking Mechanism Using Packet TTL Field. the IP address A_x of router R_x is marked on 24-bit M_x through the following process. When abnormal traffic happens in the course of marking for the writable 24 bits of a packet, router R_x marks A_x, which is its own IP address, and A_y, which is the IP address of the previous router R_y. To mark the two router addresses within the 24 bits, the router uses address values based on the hash values of the routers, which also provide an authentication function.

TTL(time to live) in all packets is an 8-bit field, which is set at 255 in ordinary packets. The value of TTL field is decreased by 1 at each router until the packet reaches the target.

Currently TTL value is used to secure bandwidth in transmitting packets on the network and to control packets that have failed to reach the target. In previous researches, TTL value was not used but a separate hop counter field was used to calculate the distance that the packet has traveled. This study, however, uses part of TTL value in packets arrived at router R_x for packet marking.

Specifically because the maximum network hop count is 32 in general, the distance of packet transmission can be calculated only with the lower 6 bits out of the 8 bits of TTL field in packet P_x arrived at router R_x. That is, the router extracts information of the lower 6 bits from the TTL field of packet P_x, names it T_x and stores it in TOS 6-bit field P_x^{TF} of the packet.

$T_x = TTLofP_x \wedge 127$ value indicates the distance of the packet from the attack system. If the packet with the value is delivered to target system V, it is possible to calculate the distance from router R_x to target system V using the value V and T_v obtained in V in the same way.

(3) Traceback Path Marking at Routers. When informed of the occurrence of abnormal traffic by the SVM-based traceback module, router R_x performs marking for packet P_x corresponding to congestion signature classified by SVM decision module.

First of all, because the router received a packet, it resets TM field in TOS field as 1. Then it calculates T_x for 8-bit TTL field of packet P_x and stores it in the 6 bits of TOS field. Then the router calculates 8-bit hash value for A_x the address of router R_x and T_x calculated earlier using hash function $H(\cdot)$, and marks the value on P_x^{MF1}, the first 8 bits of ID field. The marked packet is delivered to R_y, the next router on the routing path to the target address.

Now when router R_y checks P_x^{TM} the value of TM field in the packet and finds it is 1, the router applies the hash function to the value obtained by subtracting 1 from P_x^{TM}, which is corresponding to the 6 bits of TOS field in the packet, and router IP address A_x and marks the resulting value on P_x^{MF2}.(Such as $P_x^{MF1} = H(T_x|A_x), P_x^{MF2} = H(P_x^{TF} - 1|A_y)$).

After marking, the router set CF at 1 and sends the packet to the next router. The next router, finding TM and CF are set at 1, does not perform marking because the packet has been marked by the previous router.

5 Traceback Path Reconstruction

5.1 Malicious DDoS Attack Packet Traceback

For a packet transmitted through the network, victim system V restructures the malicious DDoS attack path. As in the figure below, let's assume that malicious DDoS attacks have been made against S_1, S_2, S_3. For the attack packet, router R_x, R_y and R_z marked 24 bits in the packet header with its own IP information and the information of 6-bit TTL field of the packet. When the malicious DDoS attack occurred, the victim systems perform traceback as follows for packets arrived.

First of all, let's say P_v is a set of packets arrived at victim system V. P_v is a set of packets corresponding to DDoS attacking, and M_v is a set of packets within P_v, which were marked by routers.

Step 1: Selects TM field is marked packets. To distinguish M_v from packet set P_v, the system selects packets in which TM field P_x^{TM} and CF field P_x^{CF} have been set at 1 as $M_v = \{P_x | P_x^{TM} == 1 \wedge P_x^{CF} == 1, x \in v\}$. That is, for packet M_i belonging to packet set M_v in a victim system, its 8-bit TTL value can be defined as $TTLofM_i$. The value is compared with T_{M_i} marked on TOS field, and the network hop count $D(M_i)$, which is the distance since packet M_i was marked, is calculated as $D(M_i) = M_i^{TF} - (TTLofM_i \wedge 127)$.

Step 2: Restructure a attack path if $D(M_i) == 2$. If $D(M_i) == 1$, it indicates that the packet was marked at the router just in front of the victim system. The method proposed in this study, however, adopts a traceback technique, it can restructure a attack path using a packet with $D(M_i) == 2$.

5.2 Malicious DDoS Attack Path Reconstruction

Packet M_i satisfying $D(M_i) == 2$ means that the packet was marked by router R_y and R_x two hops apart from the end router in front of the victim system.

Step 3: Identify marked packet. $D(M_i)$ for packet M_i is 2 because the packet was marked by router R_x, which is 2 hops apart from the router directly connected to the victim system. Thus R_x, 2 hops apart from packet M_i can be identified in the following equation $M_i^{MF1} == H(M_i^{TF} | R_x), (R_x \in D(M_i) == 2)$ and $M_i^{MF1} == H((TTLofM_i \wedge 127) + 2 | R_x), (R_x \in D(M_i) == 2)$. Of course, packet M_i can prove in the following way that a packet was marked by router R_y 1 hop apart from the victim system. $M_i^{MF2} == H(M_i^{TF} - 1 | R_y), (R_y \in D(M_i) == 1)$ and $M_i^{MF2} == H((TTLofM_i \wedge 127) + 1 | R_y), (R_y \in D(M_i) == 1)$.

Step 4: Recursively restructure the actual attack path. Now the victim system can restructure the actual attack path through which packets in

malicious DDoS attack packet set P_v were transmitted by repeating the same process for M_j satisfying $D(M_j) == n, (n \geq 3)$. When the proposed method is applied to a network structured as below, DDoS attack path AP to a victim system can be obtained as follows.

$$AP_1 = R_y \rightarrow R_x \rightarrow R_z \rightarrow S1, AP_2 = R_y \rightarrow R_3 \rightarrow R_7 \rightarrow S2,$$
$$AP_3 = R_y \rightarrow R_3 \rightarrow R_7 \rightarrow S3$$

Through the process, routers could perform not only a monitoring/identification function on network traffic using an SVM module but also a network control function using modified traceback technology. What is more, the proposed method could restructure the source of attackers by providing the function of tracing back spoofed packets adopting improved packet marking technology in order to trace back DDoS hacking paths.

6 Performance Analysis for the Proposed Method

6.1 Experiment Results

In order to evaluate the performance of the proposed method(SVM-PM : SVM based Packet Marking), the author analyzed the performance using ns-2 Simulator in Linux. According to the results of the experiment, in existing packet marking methods each router samples and marks at a probability of p to cope with DDoS attacks. Thus the number of marked packets has increased in proportion to DDoS traffic. In the method proposed in this study, a SVM technique is adopted in classifying and control DDoS traffic and as a result the number of marked packets has decreased by 12.8%. We can control the DDoS traffic by

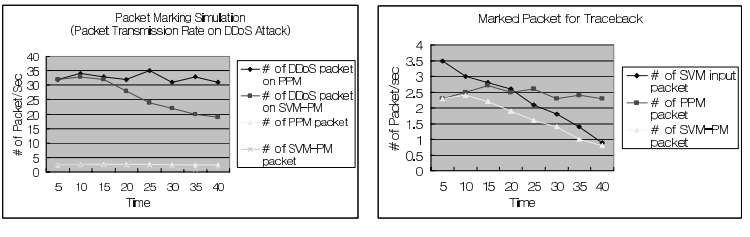


Fig. 3. Traffic Simulation Results by the Proposed Method

issuing traceback message to upper router and marking router's own address in IP packet. So, proposed mechanism(SVM-PM) can identify/control DDoS traffic by using existing SVM module and trace back its spoofed real origin address with fewer marking packet compared with previous PPM mechanism.

6.2 Analysis and Discussions

[Table 1] shows the comparison of the performance of the proposed method with that of existing IP traceback-related technologies. The method proposed in this study(SVM-PM) runs in a way similar to existing PPM, so its management load is low. Furthermore, because it applies identification/control functions to

packets at routers it reduces load on the entire network when hacking such as DDoS attacks occurs. What is more, the method proposed in this study uses an SVM-based congestion control / filter function and marks path information using the value of TTL field, which reduces the number of packets necessary for restructuring a traceback path to the victim system.

Table 1. Comparison of performance with existing IP traceback methods

	Net. load	Sys. load	Memory	Traceback	Security	Filter	Against DDoS
Filter	×	×	×	×	×	△	×
SYN fld.	×	↓	×	×	×	×	×
PPM	↓	↑	↑	△	◇	×	▽
iTrace	↓	↑	↑	△	◇	×	▽
SVM-PM	↓	↑	↑	△	◇	△	△

×:N/A ↓:low ↑:high △:good ◇:moderate ▽:bad

7 Conclusions

The proposed method can filter malicious traffic with SVM congestion signature and improves the bandwidth of the entire network. Therefore we can restructure the path to the source of DDoS attacks with a small number of marking packets. As a disadvantage, the method requires additional memory at routers for the DDoS-related identification function performed by the SVM-based filtering module.

Acknowledgement. This research was partially supported by the MIC, Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA (IITA-2005-(C1090-0502-0020)).

References

1. Computer Emergency Response Team, "TCP SYN flooding and IP Spoofing attacks", CERT Advisory CA-1996-21, Sept, 1996.
2. L. Garber, "Denial-of-Service attacks trip the Internet", Computer, pages 12, Apr. 2000.
3. C.J.C Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol. 2, pp.121-167, 1998.
4. Tatsuya Baba, Shigeyuki Matsuda, "Tracing Network Attacks to Their Sources," IEEE Internet Computing, pp. 20-26, March, 2002.
5. K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack", In Proc. IEEE INFOCOM '01, pages 338-347, 2001.
6. Jongmei Deng, Qing-An Zeng, Dharma P. Agrawal, "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", IEEE, 2003.
7. Steve Bellovin, Tom Taylor, "ICMP Traceback Messages", RFC 2026, Internet Engineering Task Force, February 2003.
8. Cristianimi N., Shawe-Taylor J., "An Introduction to Support Vector Machines", Cambridge University Press, 2000.

RCS: A Distributed Mechanism Against Link Flooding DDoS Attacks

Yong Cui, Lingjian Song, and Ke Xu

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R. China
{cy, slj, xuke}@csnet1.cs.tsinghua.edu.cn

Abstract. DoS/DDoS attacks especially the Link Flooding have exerted severe threat on Internet. In this paper we propose a novel mechanism called Rate Control System (RCS) against Link Flooding based on the correlation analysis of upper link flows. According to the feature of aggregate in DDoS attack, RCS takes DDoS attack problem as a way of flow control to simplify the situation and deploys the flow controller at the routers near the victims. As the key point of our mechanism, an algorithm is designed to differentiate the malicious packets and the normal ones and we classify the packets according to TCP flags in order to tell different flows apart. In addition we detect the malicious aggregate using correlation analysis to make clear the type and the location of the attack. Simulation results demonstrate the performance for detecting the Link Flooding DDoS attacks.

1 Introduction

As the Internet becomes popular, the worms and hacker intrusions frequently annoy people's daily life, which recalls people of the severe problem of network security [1]. Among various attacks, DoS/DDoS attacks attacking Internet by exploiting the flaws of the protocols [2,3] are often referred as to the primary threat to the Internet. The DoS/DDoS attacks can fall into two categories. One kind of the attacks makes target of single host, while another kind aims at the network infrastructure [4]. The former breaks down the victim by exhausting its resource, such as TCP-SYN flood [5], ping of death. The latter congests the network to prevent the normal user form the access to the Internet, for example ICMP flood, Smurf, UDP flood etc. The latter is often called Link Flooding for short. Because Link Flooding is designed against Internet infrastructure like core routers and DNS servers, it can be a disaster. Moreover there is a variation of DDoS Link Flooding called Distributed Reflection Denial of Service (DRDoS). Its detailed information can be found in [6]. How to defend DoS/DDoS attack effectively is a heated topic these years. Many scholars has done a lot of researches and made some progresses in defending attacks. The defenses available are either designed only for several attacks or cost a lot. As new attacks emerge rapidly, a universal defense model is needed to crack the attackers. In this paper we propose a mechanism called RCS based on the correlation analysis of upper link flows

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 764–774, 2006.

to counter the Link Flooding. At the very beginning we make clear the goals of the defense mechanism after deep analysis of Link Flooding. One is to protect the links and avoid congestion collapse and the other is to keep the quality of service during Link Flooding. Then we establish a model using the optimization theory to meet the goals. In the meantime we view defense of Link Flooding in a way of flow control to simplify the situation and deploy the flow controller at the routers near the victims to mitigate the load of links.

2 Related Work

In the recent research, people get some insights into DoS/DDoS attack: (1) More than 90% of the DoS attacks use TCP [2,5]. (2) The attack involves huge volume of packets and has a noticeable increase in terms of the number of packets [7]. (3)The malicious congestion is usually caused by flow aggregate [8]. There are many defensive techniques has been proposed. They can fall into three kinds of method.

Method 1: Filtering the malicious packets according to the pattern of data stream or single packet. This method is also can be divided into two kinds. The first kind is to filter forged packets. Some approaches make use of router, analyze the IP head of each packet and filter the ones with the wrong IP address that is out of normal range, for example ingress filtering [9]and SAVE [10]. In the Hop-Count mechanism [11], a mapping table IP2HC between IP address and the Hop-Count is built to filter the forged packets. In addition Adrian Perrig proposed Pi [4] to counter IP address fabricating. The second kind is to detection the anomaly data flow by analyzing the statistical pattern, for example the recent research PacketScore mechanism [12]. By comparing the suspect traffic pattern and the normal traffic profile, the difference can be view as the result of the attack.

Method 2: As the difference between the bad packets and the good ones is subtle, it is not so easy to take them apart. In addition there is no clear difference to tell whether it is an attack or just a flash crowd [7]. Some scholars then convert the defense problem to resource allocation problem. The Pushback [16] mechanism controls the rate of flows hat cause congestion in the near routers or upper ones by analyzing the granularity of the aggregate. Similarly the Router Throttle [17] controls the flow at the k-level routers using the max-min fair algorithm to allocate the resource. These methods also cost a lot in the deployment of the controller. Moreover when the aggregate is composed of numerous flows, each of which might be low-bandwidth, the Router Throttle will be degraded severely.

Method 3: For the concealment of source of DDoS attack, people do some researches on IP traceback in order to locate the source of attack. Source traceback does not directly address the DDoS problem. It serves as a reactive detection and deterrent tool against attack sources, for example Bellovin's ITRACE [13], D. Dean's algebraic approach [14] and Savage's IP marking mechanism [15]. They are all designed to reveal the location of the attack source. However the DDoS

in large scale especially the DRDoS, which use the reflectors to attack, make the source traceback less effective. It makes no sense to locate the innocent reflectors.

All defenses mentioned above can fall into two kinds in the point of view of deployment: the router-based and the host-based. The router-based method such as the Pushback [16] makes improvements to routers, while the host-based such as Hop-Count mechanism approach enhances the ability to restore of Internet servers against attacks. The router based method take advantage of the resource of router like high-bandwidth, but the deployment is difficult and cost a lost. The latter with less deployment difficulty need strong host server and bandwidth.

The methods or approaches mentioned above all have some shortcomings. As new attacks emerge rapidly, there is no perfect solution so far. How to learn for the methods available and combine their advantage together is a challenge. A universal defense model is needed to crack the attackers.

3 Problem Definition

In this section we describe the target network, and define the related variables and formula. Figure 1 depicts the target network topology. We model it as a connected graph $G = (V, E)$, where$V$ is the set of nodes, E is the set of edges and c_e is the capacity of $e(e \in E)$. The node s stands for the victim, which accesses Internet through the node g. We suppose there are n nodes $\{v_1, v_2, v_3, ..., v_n\}$ called upper nodes can reach s through g. In this model, we only consider the packets flows whose destination is s. For each v_i at time t we define the $r_i'(t)$ as the inbound packet rate and $r_i(t)$ as the outbound one. the Rate from g to s is $r_0(t)$, and the load of the link gs is $r_{load}(t)$, which equals the sum of the entire upper link rate. We have

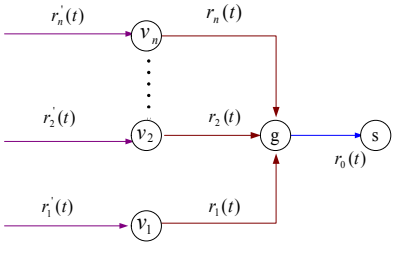


Fig. 1. The topology of target net work

$$r_{load}(t) = \sum_{i=1,2,...,n} r_i(t) \tag{1}$$

The discussion in the rest of the paper is base on the topology to depict the attack.

If the hacker wants to attack s, the link gs is mostly vulnerable. Normally the link load $r_{load}(t)$ is less than c_{gs}, so all the packets can reach s without any trouble. However when hacker employs DDoS or DRDoS attack to flood s, there are numerous flows that converge at g, which cause an aggregate. When a link is congested and persistently overloaded, all flows traversing that link experience significantly degraded service over an extended period time.

Supposing all the packets fall into m categories (the way of classification will mentioned later), for each v_i at time t we define $r_{ij}'(t)$ as the inbound packet rate, which is belong to the j^{th} category, and $r_{ij}(t)$ as the outbound one. We

use the matrices $r(t) = [r_{ij}(t)]_{n\times m}$ and $r'(t) = [r'_{ij}(t)]_{n\times m}$ to denote the rates of flow. The row of the matrix stands for the serial number of node, while the column means the serial number of packet category. From now on we view a flow as a specific category, if there is not any special announcement.

4 Modeling

4.1 Utility Function

The goals of defense we mentioned before are to protect the links and avoid congestion collapse and to keep the quality of service (QoS) during Link Flooding. To describe the QoS, we use the utility function $U(p, q)$ to evaluate it. The parameter p is the probability that a user continues to be served without interrupt and the q is the probability that a new user coming can get the service immediately while attacking. We define the function as follows:

$$U(p, q) = ap + bq \tag{2}$$

Eqn. (2) apparently consist of two parts: the utility of existing users and the new comer. a and b stand for the weight of each part. The reason we classify the users is that different users will affected by different attacks.

4.2 Defense Model

For a better understanding, we use the mathematic tool to describe the defense based on router. In addition we build a simple model employing the optimization theory to depict the mechanism counter the attack. We view the first goal of defense mechanism as the constraint of the problem, and the second goal as target function, converting two targets problem to single one, which simplify the problem. So we have the model:

$$\begin{aligned} &Max\ U(p, q) \\ st.\ &\begin{cases} r(t) &= F[r'(t)] \\ r_{load} &< c_{gs} \end{cases} \end{aligned} \tag{3}$$

Eqn. (3) shows the problem we face in the defense of attack. As the core of the defense mechanism, the formula $r(t) = F[r'(t)]$ means a transform from matrix $r'(t)$ to $r(t)$, which is done by the router just as mentioned in the section 3. The inequation $r_{load} < c_{gs}$ is the constraint that secures the link from overload. In the reality our defense mechanism will make it by limiting the inbound rate so as to prevent the link overloading. The target of the model is to maximize the utility of users with the guarantee of these constraints.

5 Defense Mechanism RCS

In this section we will detail our defense mechanism RCS. RCS will be invoked only when a congestion occur. Even when it is just a normal congestion that may

be caused by flash crowds, the mechanism will play an active role in mitigation of congestion.

5.1 The Classification of Packets

Most of the attacks exploit the shortcoming of the protocols, and different attacks have different characteristics. Learning the feature of DDoS in section 2 and the method mentioned in [12], we classify the packet according to the attributes in transport layer, such as the flags of TCP (URG, ACK, PSH, RST, SYN, FIN), or the ports of UDP which stand for different services. The classification is shown in Table 1.

Table 1. Classification of packets

Category	Description	Possible Attack
DNS query/echo	UDP Port 53	DRDOR
ICMP query/echo	ICMP(0,0)or ICMP(8,0)	DRDOR
TCP SYN	SYN set 1	SYN flooding
TCP reply of SYN	SYN and ACK are 1,data length is 0	DRDOR
TCP data	PSH is 1,data length > 0	DDoS flooding
TCP ACK of data	ACK is 1	DRDOR
TCP FIN	FIN is 1data length is 0	DDoS flooding
TCP reset	RST is 1	DDoS flooding
Other data	The other description	DDoS flooding

5.2 Detection of Aggregate

Once a serious congestion is found, the mechanism will be invoked to found the cause and determine whether it is an aggregate or not. The phenomenon called flash crowds can also cause congestion which is elaborated in some literatures [7]. We define this phenomenon as follows:

Definition 1 (Correlated flows). *Two flows are correlated, if and only if the rates of them have positive correlation relationship.*

We all also define the correlation coefficient to evaluate the how them are correlated. Before the definition, we see the rate of a flow as a stochastic variable r.

Definition 2. *r_1 and r_2 stand for the rate of two flows, the correlation coefficient of the two flows is:*

$$re = \frac{Cov(r_1, r_2)}{\sqrt{D(r_1)} \cdot \sqrt{D(r_2)}} \tag{4}$$

In the real measurement the correlated flows may not be confined by the liner relationship due to the unstable network, the definition is so strict. We use an empirical threshold to determine whether two flows are correlated or not.

5.3 Flow Control

In RCS a new flow control protocol is raised to counter the link flooding. When an aggregate of specific kind is detected, packets will be dropped proportionally to mitigate the load of link. In this way we not only make sure the efficiency of the links, but also differentiate the different packets and pass more normal packets.

We define a rate control coefficient matrix as η , which has $\eta = [\eta_{ij}]_{n\times m}$. η_{ij} is the coefficinet of link i and type j. So we have the equation:

$$r_{ij}(t) = r'_{ij}(t) \cdot \eta_{ij} \tag{5}$$

Eqn.(5)in fact realizes the function F mentioned Eq. (3). Introduce Eqn. (5) to Eqn.(3) we get :

$$\begin{aligned} &Max\ U(p,q) \\ &st.\ \textstyle\sum_{i=1..n,j=1...m} r'_{ij}(t) \cdot \eta_{ij} < c_{gs} \end{aligned} \tag{6}$$

From Eqn. (6) the solution to defend attacks is to choose appropriate η. If we found the j^{th} kind of flow that pass the node i malicious, we can reduce η_{ij} to counter the attack. So that more normal packets can get to pass, and the utility of users will rise.

5.4 Flow Control

According to Eq.(3) the transform function F is the core of defense mechanism. In this part we propose a algorithm call Link Control (LC) algorithm to convert r to r' .In every period of T, we sample the inbound rate $r(t)$ at an interval of τ and then we get the matrix $R(\tau, T)$in time series of which is also n rows and m columns.. In each column there are n rate sequences of flows in the same kind, with which we can compute a correlation matrix according to the Eq. (4). We use the vector $Re(\tau, T)$ to represent the m matrices in m columns which are explained as Eqn. (7).

```
Algorithm Rn(R(τ,T))
(1) count Re(τ,T) according to Eqn.(7);
(2) Rn initialized to [0]_{m×n};
(3) for each Re^h_{hl}(τ,T) ∈ Re(τ,T);
(4)     if Re^h_{hl}(τ,T) > φ
(5)     Rn_{hj} = 1; Rn_{lj} = 1;
(6) return Rn
```

Fig. 2. Rn function

$$\begin{cases} Re(\tau,T) &= [Re^j(\tau,T)]_m \\ Re^j(\tau,T) &= [Re^j_{hl}(\tau,T)]_{n\times n} \\ Re^j_{hl}(\tau,T) &= \frac{Cov(R_{hj}(\tau,T),R_{lj}(\tau,T))}{\sqrt{D(R_{hj}(\tau,T))}\cdot\sqrt{D(R_{lj}(\tau,T))}} \end{cases} \tag{7}$$

We define a matrix $Rn = [Rn_{ij}]_{n\times m}$ to record the correlated flows. The computation of Rn is shown in Fig. 2. The program firstly gets the $Re(t, T)$using $R(t, T)$ according to Eqn. (7) and initializes the Rn (line 1-2). Then it will search $Re(t, T)$ and if the correlation coefficient of two flows is larger than the

Algorithm LC ()
(1) Initialize to $[1]_{n\times m}$;
(2) $r_{past} = -\infty$;
(3) WHILE (1)
(4) get $R(\tau, T)$ in last T period from the monitor
(5) get current $r_{load}(t)$ from the monitor;
(6) control the flow according to current η;
(7) timer=0;
(8) DO
(9) Sleep(t_0);//wake up every t_0
(10) IF ($r_{load} > U_{gs}$)
(11) $Rn = Rn(R(\tau, T))$ //count Rn
(12) IF ($Rn! = [0]_{n\times m}$) for all Rn_{ij}: if $Rn_{ij} = 1$ then $\eta_{ij} = w \cdot \eta_{ij}$;//adjust η according to Rn
(13) ELSE for all $j, \exists i_0$: if$r_{i_0 j}(t) > \rho \times \sum_{i=1\ldots nr_{ij}(t)}$, then $\eta_{i_0 j} = w \times \eta_{i_0 j}$; /*attack through a single line*/
(14) ELSE for each η_{ij}:$\eta_{ij} = w \times \eta_{ij}$ /*no anomaly detected*/
(15) ELSE IF ($r_{load} < L_{gs}$)
(16) IF ($r_0(t) - r_{past} < \varphi$)
(17) turn off the RT;
(18) exit(1);
(19) ELSE $\forall Rn_{ij}$if $Rn_{ij} = 1$ Than $\eta_{ij} = v \times \eta_{ij}$;
(20) timer=timer+t_0;
(21) $r_{past} = r_{load}$;
(22) WHILE (timer¿T)

Fig. 3. Pseudocode of LC Algorithm

φ, the corresponding Rn_{ij}will be 1(line 3-5). Fig.3 shows the pseudocode of LC Algorithm.

The algorithm pays special attention to two anomalies. One is that the huge a mount of malicious packets come from a single link. Another is the flash crowds for the special event. It is worth while to mention that our mechanism is an online real-time defense system. Every T interval the correlation analysis will be done, and η will be adjusted at the interval of τ.

6 The Simulation and Results

We design a plan of simulation to evaluate our defense mechanism and analyze the effectiveness and the stability. We use the matlab 7.0 as our platform of simulation. In the LC algorithm we use $T = 20s, t = 2s, t_0 = 1s$and $t_0 = 90\%$ as our default configuration. In the simulation we have 5 upper nodes, 1000 hosts, and a victim host s. Each of these hosts, 20% of which are attacking hosts, will send packets through one of the 5 nodes. The attacking hosts sending rate is 20 times of normal one.

6.1 The Analysis of φ

We run our LC algorithm in the simulation of attack with different φ, and make a record of Rn respectively to determine the φ. The table below is showing the average number of correlated links (ANCL) in 100 times experiments.

Table. 2 tell us smaller φ causes more wrong correlated links, larger φ ignores more malicious ones. The principle of our algorithm is to maximize the degree of differentiation. We say $P(re_1 > \varphi) > 90\%$, $P(re_2 > \varphi) > 90\%$ and $P(re_3 > \varphi) > 90\%$ are enough, which is fulfilled when φ equals 0.65 as shown in Fig. 4.

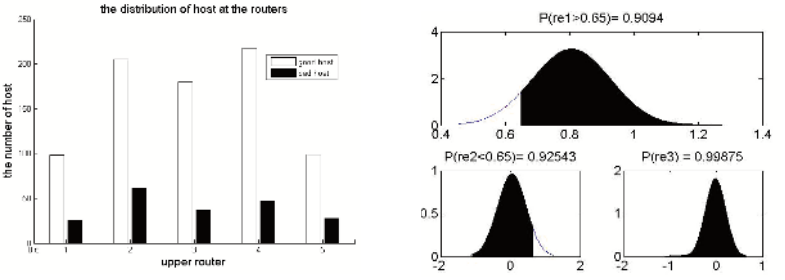
Table 2. Classification of packets

a. normal condition

φ	0.4	0.5	0.6	0.7	0.8	0.9
ANCL	2.7	2.31	1.23	0.65	0.23	0.04

b. attacking condition

φ	0.4	0.5	0.6	0.7	0.8	0.9
ANCL	5	5	4.98	4.85	4.42	2.76



a. the distribution of host *b*. three probability when $\varphi = 0.65$

Fig. 4. The analysis of φ

6.2 The Analysis of Convergence

In this part we will evaluate the convergence of our algorithm with different w and v by analyzing the fluctuation of link load. Fig. 5 show the fluctuation of link load under attack when out algorithm operate with different w and v. We use the range [10 Mbps, 5 Mbps].

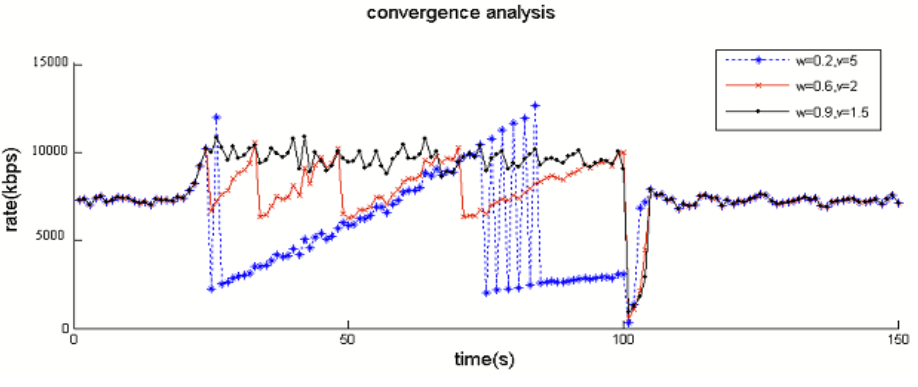
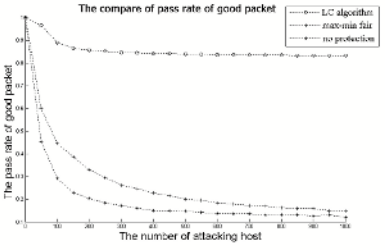
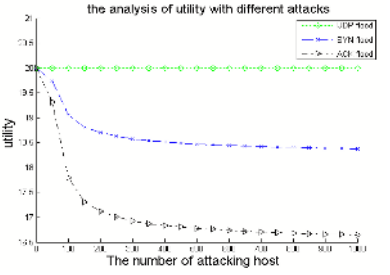


Fig. 5. The comparison of convergence with different w and v

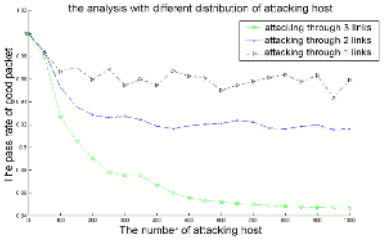
The parameter w reflects how tolerant our algorithm is when the load is beyond the bounds capacity. If w is small, say 0.2 in the figure, our algorithm will respond severely to limit the aggregate aggressively which makes the load unstable and less convergence. The parameter v conveys that our algorithm will



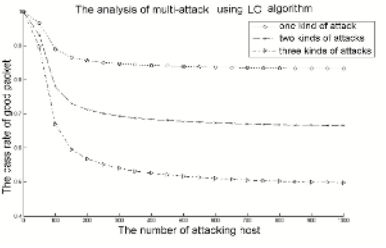
a. different algorithms and no protection



b. the utility under different attacks



c. different distribution of attacking host



d. the degradation under various attacks

Fig. 6. The analysis of performance of our algorithm

recover the flows, when the flows are over-controlled. From the figure after 100s the larger v is the more convergent it will be. But between 70s and 80s the larger v is the load will fluctuate more severely. It is worthwhile to mention that w and v should be adjusted according to the degree of attack in order to achieve the optimized result.

6.3 The Analysis of Performance

The performance of defense mechanism we understand is the pass rate of good packets (PRGP). The pass rate is the proportion of packets that is not dropped to the total. Fig.6.*a* depicts the performance comparison of our LC algorithm, max-min fair algorithm [17] and the situation with on protection. The experiment shows the advantage of LC algorithm against one kind of attack. The experiment in Fig.6.*b* is designed to analyze the utility under different attacks with $a = b = 10$. We assume the victim s is a web server, and the attack lunch SYN flooding, UDP flooding and ACK flood respectively. As the result show UDP flooding do not affect the TCP services, and SYN flooding will affect the request packets. The ACK flood affects both of them which cause a least utility. Fig.6.*c* shows the performance under attack from different number of links. If the attacking packet comes form less links, the attacking behavior are more obvious, and our algorithm is easier to recognize and control it. As there are more links that do not

need limitation, the pass rate of good packet is surely higher. Our algorithm has the assumption that hacker attack victim using only one kind of attack. Finally we discuss the degradation of our algorithm when attacker lunch multi-attacks shown in Fig.6.*d*. As we see the more attack types the worse the performance is. The experiment proved that when attacker coins all kinds of packets to attack, our LC algorithm will be degraded to Max-min fair. It is way to improve.

6.4 Conclusion

In this paper we develop and evaluate a mechanism against Link Flooding based on the correlation analysis of upper link flows. Deployed on the near routers our defense mechanism can mitigate the congestion caused by excessive volume of traffic and effectively differentiate the malicious flows and the normal ones. We evaluate our mechanism by simulation, and the result shows our mechanism is a promising method to counter the link flooding. More attention was paid to the attacks in which the hacker uses only one kind of attacks which is mostly prevalent. We will do researches on the multi-attacks in our future work.

References

1. A.K Ghosh, J. Wanken, F. Charron : Detecting anomalous and unknown intrusions against programs. Proceedings of the 14th Annual Computer Security Applications Conference
2. D. Moore, G. Voelker and S. Savage : Inferring Internet Denial of Service Activity. Proceedings of USENIX Security Symposium, 2001,August 2001
3. L. Garber: Denial-of-service attack rip the internet. IEEE Computer, April 2000
4. A. Yaar : Pi: A path identification mechanism to defend against ddos attacks. In Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA, May 2003
5. H. Wang, D. Zhang, and K. Shin : Detecting SYN flooding attacks. In Proceedings of IEEE INFOCOM, pages 1530 -1539, June 2002
6. Vern Paxson : An Analysis of Using Reflectors for Distributed Denial-of-service. Computer Communication Review31(3) 2001
7. Jung, J., Krishnamurthy, B., AND Rabinovich, M.: Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites. In Proceedings of the 11th WWW Conference (Honolulu, HI, May 2002)
8. Ratul Mahajan, Steven M. Bellovin, Sally Floyd, and John Ioannidis : Controlling high bandwidth aggregates in the network. Submitted to ACM SIGCOMM 2001
9. P. Ferguson and D. Senie : Network Ingress Filtering: Defeating Denialof-service Attacks which employ IP Source Address Spoofing. http://www.ietf.org/rfc/rfc2827.txt, 2000
10. J. Li, J. Mirkovic, M.Wang, P. Reiher, and L. Zhang : SAVE:Source address validity enforcement protocol. In Proceedings of IEEE INFOCOMM 2001, Apr. 2001
11. C. Jin, H. Wang and K. G. Shin : Hop-count filtering: An effective defense against spoofed DDoS traffic. In Proceedings of the 10th ACM Conference on Computer and Communications Security, October 2003
12. Yoohwan Kim, and Wing Cheong Lau : PacketScore: Statistics-based Overload Control against Distributed Denial-of-Service Attacks. IEEE INFOCOM 2004

13. Bellovin : "ICMP Traceback Messages" AT&T Labs. Research http://www.cs.columbia.edu/~smb/papers/draft-bellovin-itrace-00.txt.
14. D. Dean, M. Franklin, and A. Stubblefield : An algebraic approach to IP traceback. ACM Transactions on Information and System Security, May 2002
15. S. Savage, D. Wetherall, A. Karlin and T. Anderson : "Practical Network Support for IP Traceback," Proc.ACM/SIGCOMM, pp. 295-306, August 2000
16. J. Ioannidis : Implementing pushback:Router-based defense against DDoS attacks. In Proceedings of the 2002 ISOC Symposium on Network and Distributed Security
17. David K.Y.Yau, John C.S.Lui,and F.Liang : Defending Against Distributed Denialof-service Attacks with Max-min Fair Server-centric Router Throttles. In IEEE International Workshop on Quality of Service (IWQoS), 2002

Detecting Unknown Worms Using Randomness Check*

Hyundo Park and Heejo Lee**

Korea University, Seoul 136-713, South Korea
{hyundo95, heejo}@korea.ac.kr

Abstract. From the appearance of CodeRed and SQL Slammer worm, we have learned that the early detection of worm epidemics is important to reduce the damage caused by their outbreak. One prominent characteristic of Internet worms is to choose next targets randomly by using a random generator. In this paper, we propose a new worm detection mechanism by checking the random distribution of destination addresses. Our mechanism generates the traffic matrix and checks the value of rank of it to detect the spreading of Internet worms. From the fact that a random binary matrix holds a high value of rank, ADUR (Anomaly Detection Using Randomness check) is proposed for detecting unknown worms based on the rank of the traffic matrix. From the experiments on various environments, we show that the ADUR mechanism effectively detects the spread of new worms in an early stage, even when there is only one host infected in a monitoring network.

1 Introduction

An Internet worm is the one of malicious codes that propagates by copying itself onto other computers. Such a self-replicating malicious code scans vulnerable hosts on a network, replicates itself to the vulnerable hosts without user intervention. Starting from one decade after the Morris worm in 1988, the number of incidents according to Internet worms is growing drastically. CodeRed and Nimda worm infected hundreds of thousands of vulnerable computers at year 2001. At that time, from public institutes to personal users suffered from the damage caused by CodeRed and Nimda. The amount of damage was accounted for millions of dollar [1-4].

On the history of Internet worms, the SQL Slammer worm is known as the fastest spreading worm. It spends mere 10 minutes to infect 90 percent of vulnerable hosts in the Internet. And the number of infected hosts increased two times per 8.5 seconds. This speed is much faster than CodeRed, which increases two times per 37 minutes [5]. The first step toward countering with the epidemic of a worm is "early detection" as soon as possible. However, signature-based detection algorithms are not proper to detect new worms or polymorphic worms since they can always change their codes. On the contrary, anomaly-based approaches can be used for detecting such worms, at the expense of higher degree of complexity and false alarms.

* This work was supported in part by the ITRC program of the Korea Ministry of Information & Communications under the grant IITA-2005-(C1090-0502-0020) and the BK21 program of the Korea Ministry of Education.

** To whom all correspondence should be addressed.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 775–784, 2006.

Anomaly-based approaches have been studied in several ways. Zou et. al. [6] proposed a Kalman filter-based detection algorithm which detects the trend of illegitimate scans to a large unused IP space. Wu et. al. [7] proposed a victim counter-based detection algorithm that tracks the increased rate of new infected hosts. The algorithm warns when abnormal events occur consecutively over a certain number of times. Berk [8] proposed an anomaly detection algorithm using ICMP "Destination Unreachable" messages which can be collected at border routers to infer worm activities. Previous approaches determine an on-off binary condition based on a "threshold" value, but rarely provide further information for defending the worm epidemic such as infected subnet locations in a monitoring network.

In this paper, we propose a new method for detecting the spread of Internet worms, which is called ADUR (Anomaly Detection Using Randomness check). The ADUR mechanism can detect a new worm by measuring the randomness of destination addresses in network traffic, where the randomness is formed when a worm propagates randomly over the Internet. By checking the randomness of address distribution, ADUR distinguishes between the status of normal conditions and the state of worm epidemics. Two main features of ADUR are the use of "matrix" representations and exclusive-or (XOR) operations. Matrix representations give many benefits to implement particular operations regardless of network size and traffic volume. And the XOR operator diminishes the effect of normal traffic and magnifies the effect of worm traffic, which results in reduced false alarms. By measuring the dynamics of the rank of traffic matrix, ADUR can detect the worm epidemic in an early stage.

Main contributions of this study is three-fold. First, we propose a novel approach to detect unknown worms based on the randomness of worm traffic. Second, we suggest an anomaly detection algorithm based on matrix and its simple operation of XOR, which greatly increase the flexibility and the accuracy of detection. Finally, the algorithm gives additional information such as infected subnet locations when a worm is detected.

The rest of this paper is organized as follows. In Section 2, we explore scanning methods of Internet worms which have been used to choose a target host. Section 3 describes how to check the randomness of traffic. In Section 4, we propose the ADUR mechanism and show the reason why it uses the XOR operator. The evaluation of the proposed ADUR mechanism is shown in Section 5. We summarize our results and conclude the paper in Section 6.

2 Scanning Methods of Active Worms

In this section, we describe the scanning methods of Internet worms and show the relationship to the randomness of traffic generated by an individual scanning method. Scanning methods can be classified onto four categories [9]: hitlist scanning, topological scanning, local scanning, and permutation scanning.

In order to show the spreading speed of Internet worms, we can use an analytical model such as AAWP (Analytical Active Worm Propagation) [15]. AAWP is a worm propagation model based on discrete time.

In the AAWP model, the number of infected hosts at time tick i is shown in Eq. (1), where N is the total number of vulnerable hosts in the Internet, T is the size of address

Table 1. Attack signatures of nine attacks

Scanning method	Description	Example
Hitlist scanning	The list of vulnerable hosts is used and outgoing connections are increasing suddenly.	Warhol [10]
Topological scanning	The information of target is gathered on the infected host, and outgoing connections are increasing suddenly.	Morris [11]
Local scanning	Most targets are selected within the local network, and failed messages for connection requests are increasing.	CodeRed[12], Nimda [13]
Permutation scanning	This is a sort of local scanning but avoids the overlapping of scanning ranges.	Slammer [14]

space used by the worm to scan, s is the scan rate, and n_i is the number of infected hosts at time tick i.

$$n_{i+1} = n_i + [N - n_i]\left(1 - \left(1 - \frac{1}{T}\right)^{sn_i}\right) \tag{1}$$

On Eq. (1), let us assume that the initial time tick is 0, i.e. i=0. And the value of n0 is equal to the initial hitlist size. Fig. 1 shows the distribution of infected hosts as a function of time tick. Even with a different hitlist size, the number of infected hosts increases drastically when the value of n passed 10,000, as shown in Fig. 1.

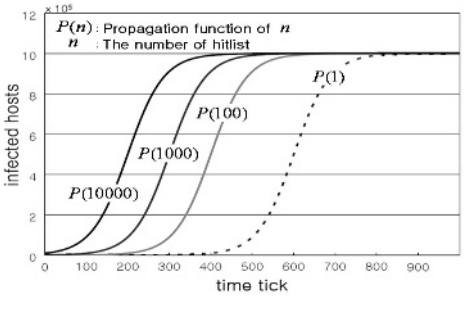


Fig. 1. The number of infected hosts as a function of time tick when n0 = 1, 100, 1000, and 10000, respectively

At the beginning of worm propagation, the speed of propagation is slow. However, when the number of infected hosts exceeds a certain point, e.g. 10,000 in Fig. 1, the infection is growing rapidly. This is caused by the fact that, as the propagation proceeds, the number of scanning packets also increases along with the increased number of infected hosts. Thus, infection is accelerated by finding remaining susceptible hosts more rapidly.

A hitlist scanning worm has a list of IP addresses and the sequence of the addresses is not likely to form a uniform distribution. Thus, the traffic generated by hitlist scanning

has the property of randomness. Topological scanning worms gather the information of target hosts from the information on the infected host. The sequence of target IP addresses, gathered on the infected host, is not uniformly distributed. Therefore, the traffic generated by the topological scanning worm has also the randomness. The local and permutation scanning worm uses a random generator to generate target IP addresses with particular constraints. Eventually, the sequence of those IP addresses generated by the local or permutation scanning, have the randomness.

Thus, conventional worm propagation strategies produce the property of "randomness" in target hosts. This implies that we can monitor the spreading of worms by measuring the randomness of destination addresses in network traffic. In this study, we attempt to measure the degree of randomness in traffic in order to catch the spreading of high speed worms.

3 Matrix Rank as a Randomness Metric

Many approaches have been proposed for testing the randomness and one cost-effective approach is checking the linear-dependency among fixed-length substrings of its original sequence. In order to check the linear-dependence among rows or columns of a matrix, we can use the rank of a matrix [17]. The randomness test using by the rank of the matrix has been broadly used as defined in a specification of one of tests coming from the DIEHARD [18] battery of tests.

One easy way to compute the rank of a matrix is counting the number of non-zero rows after applying the Gaussian elimination method to the matrix. In other words, the rank of the matrix is equal to the number of leading 1's [16].

In case of a random m x n binary matrix, the value of rank has the following probability where rank r = 1, 2,. . . , min(m, n) [17].

$$2^{r<n+m-r>-nm} \prod_{i=0}^{r-1} \frac{\left(1-2^{i-n}\right)\left(1-2^{i-m}\right)}{\left(1-2^{i-r}\right)} \tag{2}$$

From the Eq. (2), we can get the distribution of probability for a given random matrix. In case of 64x64 random binary matrices, the distribution of probabilities for the value of rank is shown as Fig. 2.

If one 64x64 random matrix is given, the probability that the value of rank exceeds 60 is over 99.995% from Eq. (2). This implies that, if the 64x64 binary matrix is a random matrix, the rank of the matrix will be larger than 60 with high probability. Thus, we can use the rank of a matrix to determine the randomness of element distribution.

4 Anomaly Detection Using Randomness Check

We propose an anomaly-based worm detection algorithm, called ADUR (Anomaly Detection Using Randomness check). This section describes the ADUR mechanism which includes a matrix representation and its XOR operations. We show how to express network traffic on a matrix, and how to use the XOR operation in order to diminish the

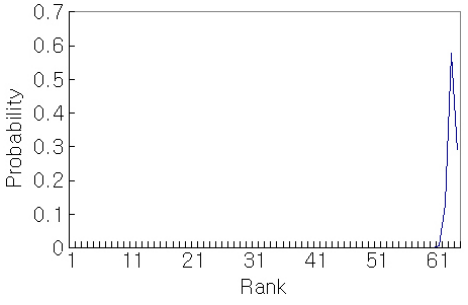


Fig. 2. Probability distribution of the rank of a 64x64 random binary matrix

effect of normal traffic and magnify the effect of worm traffic. Then, the rank of the matrix is used for measuring the randomness of traffic.

4.1 ADUR System Design

The ADUR mechanism is to detect the spreading of Internet worms through checking the randomness of traffic. Traffic data can be classified into two categories based on their direction: incoming and outgoing. ADUR checks two directions respectively in order to get more accurate attack information such as either entering or departing the network. Checking the randomness of traffic can be accomplished by calculating the value of rank of the matrix representing network traffic for a given period of time.

Let M_I denote the matrix marked with incoming traffic. And we let M_O represent the matrix marked with outgoing traffic. $R(M_I)$ and $R(M_O)$ represent the value of rank of matrix M_I and M_O, respectively. Then, the value of $R(M_I)$ and $R(M_O)$ can be used to determine whether worm is active or not. There are four states depending on the ranks $R(M_I)$ and $R(M_O)$ of traffic: calm, flowing, ebbing and flooding.

1. Calm: When both of $R(M_I)$ and $R(M_O)$ remains in a small range, there is no suspicious activity of worms.
2. Flowing: When $R(M_I)$ suddenly increases but $R(M_O)$ remains steady in a small range, the internal network is under attack by the worms in other networks.
3. Ebbing: When $R(M_I)$ remains steady in a small range but $R(M_O)$ suddenly increase, the worms in the internal network are attacking other networks.
4. Flooding: When both $R(M_I)$ and $R(M_O)$ suddenly increase, the internal network is flowing and ebbing.

4.2 ADUR System Design

The ADUR mechanism detects the worm spreading using the value of rank of the matrix marked with network traffic. Matrix representation for both incoming and outgoing traffic is described as follows.

In IPv4, we can capture source IP address and destination IP address and divide each IP address into four octets. The length of each IP_1, IP_2, IP_3 and IP_4 is one octet.

$$IP_1.IP_2.IP_3.IP_4 \tag{3}$$

Matrix expression consists of two operations. The first one is placement. The other one is storing information. When a packet is captured in a monitoring network, the packet is mapped into a specific location of a matrix, which is determined by the destination address of the packet. Without loss of generality, we assume that the matrix size is 64x64 when the size of monitoring network is /24. Then the placement function is described as Eq. (4).

$$\begin{aligned} i &= (IP_4/16) \times 4 \\ j &= (IP_4\,(mod\ 16)) \times 4 \end{aligned} \tag{4}$$

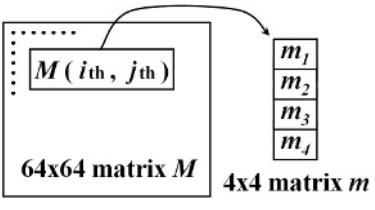


Fig. 3. Construction of matrix M by mapping a packet to a sub-matrix m

Next, we need to store some information of the packet onto the matrix. The matrix size 64x64 can hold 16 bits information for each packet. Since worms are likely to change the last two octets more frequently than the first two octets, the last two octets of an IP address will be stored at the 4x4 sub-matrix. In case of outgoing traffic, the destination address of a packet is used for storing such information onto the matrix. In case of incoming traffic, the source address of a packet is used instead of the destination address. Fig. 3 illustrates the information storing on a 4x4 sub-matrix, where the matrix size is 64x64. Furthermore, the 4x4 sub-matrix consists of four 1x4 sub-matrices, i.e. m1, m2, m3, m4 in Fig. 3. The contents of 1x4 sub-matrices are described in Eq. (5).

$$\begin{aligned} m_1 &= first\ 4\ bit\ of\ IP_3 \\ m_2 &= last\ 4\ bit\ of\ IP_3 \\ m_3 &= first\ 4\ bit\ of\ IP_4 \\ m_4 &= last\ 4\ bit\ of\ IP_4 \end{aligned} \tag{5}$$

We can extend the 64x64 matrix representation in a /24 network to larger networks. When monitoring a /16 network, we can increase the size of matrix such as the 256 number of 64x64 matrices, i.e. 256x64x64. In this way, the principle of matrix expression can be scalable to any size of network. Reducing the matrix size for larger networks is the future work of our study.

4.3 XOR Operator on Matrix Sequence

We need to consider only suspicious traffic but eliminate the effect of legitimate traffic. It is found that the simple operation XOR is greatly effective for this purpose. Let M_t

denote the matrix at time t. Eq. (6) shows the XOR operation on a sequence of matrices, which dramatically reduces the influence of normal traffic on the value of rank.

$$R(M'_t) = R(M_t \oplus M_{t-1}) \tag{6}$$

Matrix M' is made by the XOR operation of two consecutive matrices, which eventually remove the most portion of legitimate traffic in the matrix because legitimate traffic lives longer than one time unit. When time tick is an apt short, the legitimate traffic is sufficiently removed without the loss of the impact of suspicious traffic. Experimental results will be shown for different time ticks in next section. The XOR operation on the consecutive matrices is the key factor to detect the worm spreading.

5 Evaluation of ADUR

In order to evaluate the effectiveness of ADUR, we show the exponential results with real traffic. The traffic data is used the packets captured at a university network in July 29, 2004. For the purpose of making a situation of worm epidemic, we have injected various types of random scanning packets on the traffic.

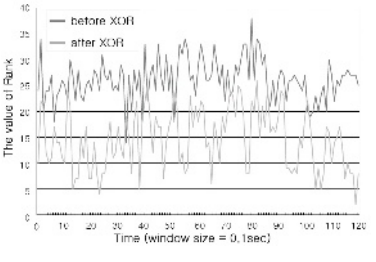


Fig. 4. The rank of matrix before and after XOR operation

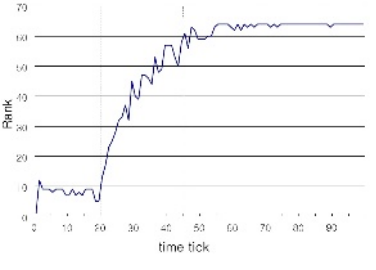


Fig. 5. The rank of matrix when randomly generated connections are added

5.1 Effect of XOR

The ADUR mechanism eliminates the effect of legitimate connections in the traffic by the use of XOR operation on the consecutive matrices. Fig. 4 shows the values of rank "before" and "after" XOR operation. Since the matrix after the XOR operation has only suspicious traffic, such as new connections on the network, the value of rank greatly reduces comparing with the matrix before the operation. In other words, the ranks will increase sharply when new worms spread over the network.

5.2 Rank and Random Connection

Fig. 5 shows the value of rank when adding more random connections. After 20th tick, we injected one random connection per time tick. When there are more than 25 random connections, the value of rank becomes over 60. This shows the transition of state from "calm" to "ebbing." From the fact that conventional worms can generate thousands of

new connections per second, we can confirm that the ADUR mechanism has an ability to detect new worms in an early stage of the worm propagation.

5.3 Effect of Window Size on the Value of Rank

Traffic matrix M is constructed by the traffic gathered for a given time period, called "window size," then the value of rank is measured after passing every window size. This unit time is counted as one time tick in this paper. Here we conduct an experiment for measuring the effect of window size.

The Fig. 6 shows the rank of matrix as a function of time tick with three different window sizes. Each graph shows the result with the same traffic data in calm state. As the window size increases, the amount of traffic for construction a matrix M also increases. It implies that the rank of M will increase as the windows increases. However, this increment is quite limited in a certain boundary, e.g. 20 as shown in Fig. 6, but adding random connections will greatly increase the range of the rank. It shows that the ADUR mechanism is robust to the window size and also to the traffic volume. Also, it shows the possibility that ADUR can be used in high-speed networks.

In order to evaluate the effectiveness of ADUR, we measured the dynamics of ranks as the worm proceeds. Fig. 7 shows the number of infected hosts which is modeled with AAWP. Fig. 8 shows the variation of ranks associated with the propagation shown in Fig. 7. The rank of the traffic matrix by AAWP model responds quicker than the speed of infection, which is depicted by Fig. 7 and Fig. 8. In Fig. 8, two different networks were considered: /24 and /16 as the size of a monitoring network. Monitoring larger networks can give better looking glasses so that results in earlier detection. Even in a small network such as a /24 network, we can catch the symptom rapidly such as 650 unit time in Fig. 8, where only 20% of hosts in the network got infected as shown in Fig. 7. It implies that the ADUR mechanism is effective even by monitoring a small network. The effectiveness of ADUR is measured in a real network traffic. The traffic is gathered in a university network and worm traffic is injected to the normal traffic,

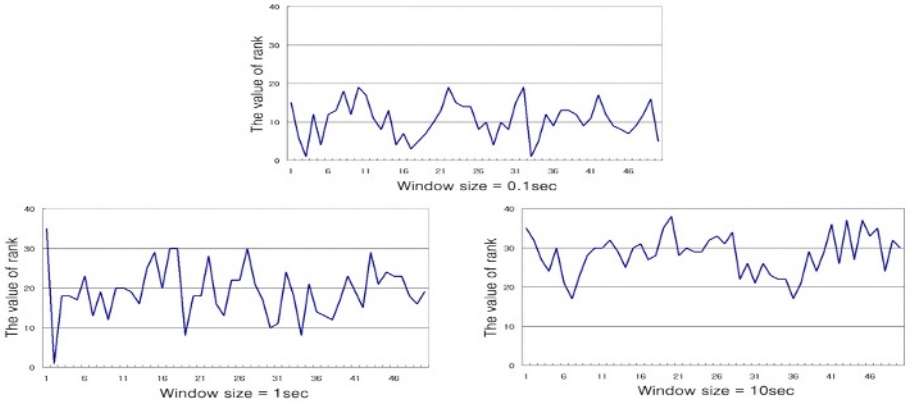


Fig. 6. The relation between the value of rank and the windows size

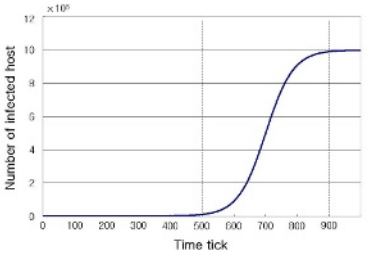


Fig. 7. The number of infected hosts modeled by AAWP as a function of time tick

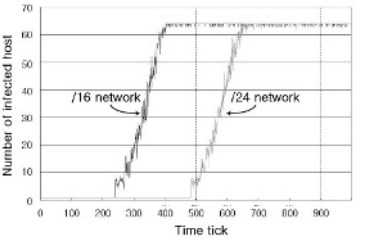


Fig. 8. The corresponding value of rank when worms spread with the AAWP model

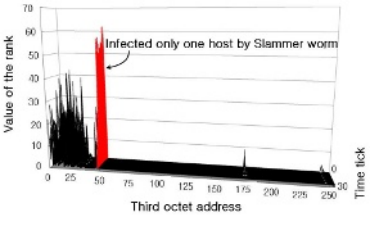


Fig. 9. Rank distribution for a /16 network, where only one host is infected by Slammer

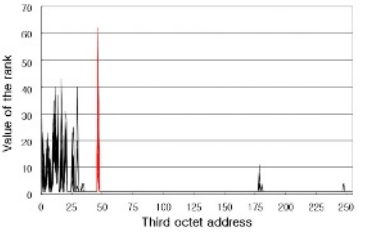


Fig. 10. Corresponding 2-D graph to Fig. 9, which also shows the infected subnet location

where only one host is infected by the Slammer worm. The rank distribution is shown in Fig. 9 and Fig. 10. This is the situation that one host located in an unused network is infected. Fig. 9 and Fig. 10 show the case that the infection of a subnet 48, which means the third octet is 48 in a /16 network. Why we input the infected host on unused local network? Because when the infected host exists in an unused network, the change of the rank by the Slammer worm correctly distinct from the normal condition. The value of rank with normal traffic is under about forty. But the value of rank with the traffic including one host infection is over sixty. So, even though only one host is infected by a worm, the ADUR mechanism can detect such infection and give additional information such as infected subnet locations.

6 Conclusion

We have proposed an unknown worm detection algorithm called ADUR. The proposed mechanism examines whether destination addresses have the property of random distribution. Matrix expression of network traffic and simple XOR operation on two consecutive matrices give a clue of worm spreading by measuring the rank of the matrix. This paper showed that the ADUR mechanism can detect unknown worms in an early stage of worm spreading and robust to the size and speed of a monitoring network and the volume of traffic. We have a plan to run this mechanism in high speed networks and try to experiment with more instances. Furthermore, we will extend the mechanism to work when multiple worms spread simultaneously.

References

1. R. Russell and A. Machie: CodeRed II worm, Tech. Rep., Incident Analysis, Security Focus, Aug. 2001.
2. A. Machie, J. Roculan, R. Russell, and M. V. Velsen: Nimda worm analysis, Tech. Rep., Incident Analysis, SecurityFocus, Sep. 2001.
3. CERT/CC: CERT Advisory CA-2001-26 Nimda Worm, `http://www.cert.org/advisory/CA-2001-26.html`, Sep. 2001.
4. D. Song, R. Malan, and R. Stone: A snapshot of global Internet worm activity, Tech. Rep., Arbor Network, Nov, 2001.
5. H. Park and H. Lee: Evaluation of malicious codes, Tech. Rep., IIRTIRC, 2004.
6. C. C. Zou, L. Gao, W. Gong, and D. Towsley: Monitoring and early warning for Internet worms, Proc. of ACM CCS, Oct. 2003.
7. J. Wu, S. Vangala, L. Gao, and K. Kwiat: An efficient architecture and algorithm for detecting worms with various scan techniques, Proc. of NDSS, Feb. 2004.
8. V.H. Berk, R.S.Gray, and G. Bakos: Flowscan: Using sensor networks and data fusion for early detection of active worms, SPIE AeroSense, Vol. 5071, pp. 92-104, 2003.
9. S. Sraniford, V. Paxson, and N, Weaver: How to own the Internet in your spare time, the 11th USENIX Security Symposium (Security '02), Aug. 2002.
10. N. Weaver: Warhol worms: The potential for very fast Internet plaques, `http://www.cs.berkely.edu/~nweaver/warhol.html`.
11. M. Eichin and J. Rochlis: With microscope and tweezers: An analysis of the Internet virus of November 1988, IEEE Symposium on Security and Privacy, 1989.
12. C. C. Zou, W. Gong, and D. Towsley: CodeRed worm propagation modeling and analysis, Proc. of ACM CCS, Nov. 2002.
13. A. Machie, J. Roculan, R. Russell, and M. V. Velzen: Nimda worm analysis, Tech. Rep., Incident Analysis, SecurityFocus, Sept. 2001.
14. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver: Inside the slammer worm, IEEE Magazine of Security and Privacy, pp. 33-39, Jul./Aug. 2003.
15. Z. Chen, L. Gao, K. Kwiat: Modeling the spread of active worms, IEEE INFOCOM, 2003.
16. H. Anton: Elementary linear algebra, 7th ed. John Wiley & Sons, Inc. 1994.
17. G. Marsaglia and L. H. Tsay: Matrices and the structure of random number sequences, Linear algebra and its applications 67, pp. 147-156, 1985.
18. G. Marsaglia: DIEHARD: a battery of tests of randomness, `http://stat.fsu.edu/~geo/diehard.html`.

A Hypothesis Testing Based Scalable TCP Scan Detection

Qianli Zhang and Xing Li

Tsinghua University, Beijing 100084, China
zhang@cernet.edu.cn

Abstract. The wide spread of worms, DDOS attacks and scan activities have greatly affected the network infrastructure security. For scan detection, traditionally most detection methods are flow based, thus undesirable for gigabits or multi-gigabits networks. To deal with this scalability problem, in this paper, a novel scan detection method is proposed, in which no flow record is required to maintain. Based on the observation that scans will generally generate a large volume of return RST packets, a hypothesis testing based approach is proposed. Experiments in practical network and on the DARPA 1998 datasets indicate that this algorithm is effective.

1 Introduction

The wide spread of worms, DDOS attacks and scan activities have greatly affected the network infrastructure security[1]. To mitigate these security risks, many solutions have been suggested. Among them are the intrusion detection (and mitigation) systems. Two classical approaches to intrusion detection have been anomaly detection and signature detection. Signature detection[2] is useful to detect an important class of attacks (e.g., known vulnerability exploit) but is not helpful in detecting other attacks (e.g., scans, DDOS attacks) which are not characterized by a signature, but by unusual behavior across a set of packets. While anomaly detection also targets such attacks, anomaly detection is often very general, and works by automatically identifying a baseline for normal network behavior (using say wavelets [5] or change point detection [6]) and then flagging deviations from such behaviors as possible attacks. The key factor of anomaly detection is the establishment of normal behaviors.

Scan detection is one of the most important applications of anomaly detection techniques. Scan is defined as a tentative process when the attackers probe the destination system for possible vulnerabilities or services. The scan detection is important because, in most cases, scans indicate the initial interest of the attackers. An accurate scan detection process may be valuable to take protection activity in advance. Also, worms are spreading with the heavy application of scan. The detection of large range scanning activities may shed light on the discovery of worms.

There have been a number of researchers and vendors interested in the scan detection problem[12][7]. Scan detection has been excessively researched and

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 785–792, 2006.

most important IDS has implemented the scan detection like snort[2] and bro[3]. In most cases, scan detection is implemented as part of a network based IDS and will process the traffic traces to accumulate flows to detect scans. For example, Snort maintains a large vector per-source to count all the ports and destinations each source talks to. Not only does this plugin take a large amount of space, but it also slows down the snort code considerably when it is enabled. Bro[3] also maintains per-flow state in order to detect evasion and other attacks. Since the wide deployment of high speed network (gigabits and multi-gigabits network), running such kind of detection engine may require very expensive computation resources, thus make it undesirable in high speed network anomaly detection.

In this paper, we want to address the problem of scan detection in a scalable method. The organization of this paper is as follows. In Section 2 we describe the previous scan detection algorithm. Section 3 will introduces the disproportional RST packets based hypothesis testing and its application in scalable scan detection. In Section 4, we describe experimental evaluation of this algorithm and we conclude in section 5.

2 Related Works

Free open source IDS tools such as Snort[2], Bro[3] can be used to detect port scans, but they (as far we can ascertain) employ per-flow state. Most scan detection techniques[9][2] in the literature are based on detecting N events in T seconds. We call this approach as burst-based algorithm since generally it assumes N is notable larger than in normal scenario. In this algorithm, both the records of all the hosts and the flows are required to maintain. Another approach[3] relies on failed connections as a better indicator of a scan. Leckie et.al[11] use probabilistic approaches to estimate the degree to which a given local IP address is unusual. SPICE[16] is an offline analysis algorithm to detect stealthy scans and cannot be performed scalably in the network. A recent paper by Jung et.al[10] apply threshold based random walks for fast portscan detection. The need to track for each remote host the different local hosts to which it has connected to makes the scheme unscalable. MULTOPS[8] is a data-structure maintained by each network device that detects bandwidth attacks by the significant, disproportional imbalance between packet rates going to and coming from the victim or attacker.

The general notion of scalable attack detection has been addressed recently by Yaar et.al in [13]. However, their work requires routers to implement marking along with some header changes to support marking of packets. Their work builds on other traceback related schemes. In [14] the authors introduce a scalable scan detection algorithm, partial completion filters(PCFs) based scan detection. This approach relies on the fact that the gap between SYN packets number and FIN packets number should not be too large for a given host. Despite of the fact that many busy systems may have a large gap between SYN packets and FIN packets in some specific period, this algorithm is restricted to detect SYN scans and could not be extended to detect stealthy scans.

3 Hypothesis Testing Based Scalable Scan Detection

To better present the hypothesis testing based scan detection, we will first summarize the known TCP scan methods. In the early stage, TCP port scans only use the standard connect function to probe the destination ports and will take a lot of time. To deal with this drawback attackers improve the scan techniques with the introduction of SYN scan, in which only one SYN packet is sent to know the status of destination port. The above two method of scan will generate a large proportion of SYN packets thus are conspicuous to the early scan detectors[17][18]. So in [4], a new scan technique, stealthy scan is invented. Stealthy scan tries to avoid the heavy use of SYN packets, instead, it use some less conspicuous packets like FIN, NULL etc. The basic idea behind stealthy scan is: in TCP, inbound packets in question will be ignored for an open port while a RST packet will be generated for a close port. Attackers could send an odd packet and wait for the response, a RST packet received will indicate destination port close while no packet received in a given time will suggest an open port. Classified by the packets attackers send, stealthy scan can be NULL scan, which will send no flag set packet, CHRISTMAS scan, with FIN+URG+PUSH flags set packets to scan, and FIN scan, with only FIN flag set, these methods are described in table 1. Except the above mentioned methods, slow scan is also a popular "stealthy" scan method since the slow scan speed avoid most of burst-based detectors' attention.

Table 1. Scan methods

Probe packets	Open port return	Close port return
SYN	SYN+ACK	RST
FIN	×	RST
NULL	×	RST
CHRISTMAS	×	RST

From the above table, an observation may be noticed that in scans, attackers may receive more than normal returned RST packets. Considering the fact that most systems only have a small proportion of ports open, this observation is not surprising. Thus if a host has received disproportional large number of RST packets, this host may be engaged in a scan.

RST packets may also be generated in normal situations. For example, some servers may use RST packets to close a session quickly. Thus it will introduce discriminations to busy servers if only count the received RST packets. A natural approach is to calculate the percentage of received RST in all received packets for a given host, if it exceeds a threshold, then a scan is happening. Since the threshold is difficult to set properly, instead an adaptive process is suggested.

Assume a random variable X_i to be 1 if the i'th received packet has flag RST set. Let p be the probability of a TCP packet being RST in normal situation, i.e.

$$X = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases} \tag{1}$$

Assume in a specific period T, there are n^H packets, $X_1^H, X_2^H, ..., X_{n^H}^H$, received by given host H. Excluding packets received by H, a total number of n TCP packets, $X_1, X_2, ..., X_n$, have been received, the MLE of their correspondent p is:

$$\begin{array}{l} \widehat{p} = \frac{\sum_{i=1}^{n} X_i}{n} \\ \widehat{p^H} = \frac{\sum_{i=1}^{n^H} X_i^H}{n^H} \end{array} \tag{2}$$

To decide whether or not H is scanning, based on the assumption that most host are not scanning, the following hypothesis test could be constructed.

$$\begin{array}{l} H_0(\text{not scan}) : p^H \leq p \\ H_1(\text{in scan}) : \ \ p^H > p \end{array} \tag{3}$$

Let

$$\widehat{p_T} = \frac{\sum_{i=1}^{n} X_i + \sum_{i=1}^{n^H} X_i^H}{n + n^H} \tag{4}$$

It could be proved[15] that if $p^H = p$:

$$Z = \frac{\widehat{p^H} - \widehat{p}}{\sqrt{(\frac{1}{n} + \frac{1}{n^H})\widehat{p_T}(1-\widehat{p_T})}} \rightarrow n(0,1) \tag{5}$$

Thus the test will reject H_0 if $Z > Z_\alpha$ for a given level α.

In practice, to avoid the false positive when a host has only received several packets and most of them are RST packets, A threshold of minimum number of RST packets received is also set. In this paper, this threshold is set to be 5. The Z_α is set to be 3, which represents a confidence level of 0.9987.

Comparing to flow based scan detection, an obvious merit of this algorithm is that no flow record is required. For each packets received, only the record of the destination host is affected. Consider the fact that the number of the host records are generally much less than the number of the flow records, this algorithm requires much less memory. Also, since no flow record is required, the processing for each received packets is much simpler: only one lookup and at most two addition operations are needed.

4 Experiments and Results

4.1 DARPA 1998 Datasets

To evaluate this algorithm's performance, the DARPA[19] 1998 datasets are used. DARPA intrusion detection evaluation datasets are the first formal, repeatable, and statistically-significant evaluations datasets for intrusion detection

systems. In this paper, only a small part of the datasets is used: the 1998 training datasets, week6, Thursday's tcpdump data. The reason to select this day's data is because there are more portsweeps in this day's data. The comparison is between the algorithm presented in this paper and the algorithm adopted by Snort. The results is indicated in table 2. (◯ for detecting successfully and × for failing to detect)

Table 2. Experiment with DARPA datasets

Week	Day	Attack Name	Time	Source Machine	Snort result	This algorithm result
6	Thurs	dict	10:34:46	206.186.80.111	×	◯
6	Thurs	neptune	11:32:23	230.1.10.20	◯	×
6	Thurs	portsweep	12:03:45	202.247.224.89	◯	◯
6	Thurs	portsweep	12:29:51	207.103.80.104	◯	◯
6	Thurs	neptune	13:31:08	10.20.30.40	◯	×
6	Thurs	satan	13:57:45	195.115.218.108	◯	◯
6	Thurs	ipsweep	14:10:09	197.218.177.69	◯	◯
6	Thurs	portsweep	14:41:47	206.48.44.18	×	◯

The specific description of attacks is provided by DARPA datasets, here we only excerpt the ones of interest.

Table 3. Attacks description of DARPA datasets

Name	Description
ipsweep	Surveillance sweep performing either a port sweep or ping on multiple host addresses.
portsweep	Surveillance sweep through many ports to determine which services are supported on a single host.
satan	Network probing tool which looks for well-known weaknesses. Operates at three different levels. Level 0 is light.
neptune	Syn flood denial of service on one or more ports.
dict	Guess passwords for a valid user using simple variants of the account name over a telnet connection.

In all 3 portweep, 1 TCP based ipsweep, 1 satan attacks, Snort detects 4 of them while this algorithm detects all. The reason why snort cannot detect the third portsweep is because Snort use burst-based algorithm, thus cannot detect slow ongoing probes. On the other hand, this algorithm cannot be evaded by slow scan, since it accumulate the RST packets count over a long time period.

The false positive may be worth of further research. For the false positive caused by Snort, 2 of them are caused by neptune attacks, since syn-flood often initialize a large number of flows within a short period. The false positive of this algorithm is a dict attacks, which will often failed and thus generate a lot of RST

packets. These false positive may better reflect the algorithm level difference between this algorithm and snort. Except those, another 11 false positive are introduced in Snort while only 3 for this algorithm.

We also measure the estimated $\widehat{P_T}$ for the first 7 hours, as shown in figure 1, the ratio stabilizes after reaching 0.01. It demonstrates that even in a datasets dedicated for intrusion detection evaluation, the RST packets are rare, thus provide more support for the design of this algorithm.

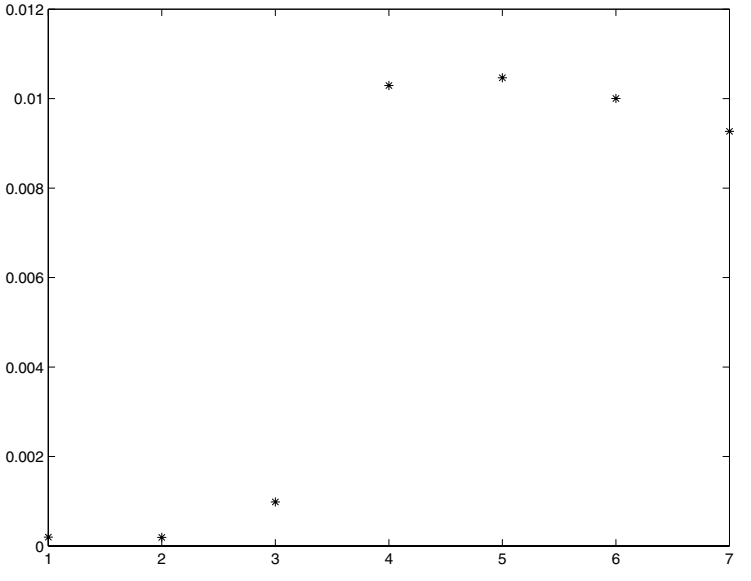


Fig. 1. $\widehat{P_T}$ for the first 7 hours

4.2 Experiments in Practical Network

Another experiment, to evaluate the algorithm's performance in practical networks is also made. Two separated network are selected, one is the location of scanners, and the other is the destination of scans. NMAP[4] is used to generate the following scans. All packets are captured by the detector. The result is shown in table 4.

Experiment results indicate that this algorithm could discover the stealthy scans reliably.

5 Conclusion

In this paper, a new method to detect scans is purposed and its performance is studied. An initial experiment indicates that this algorithm is effective in the detection of both traditional SYN scans and more advanced stealthy scans. Since

Table 4. Experiments in practical network

Type	Command	Destination	Detection result
SYN	nmap -sS	host	◯
FIN	nmap -sF	host	◯
NULL	nmap -sN	host	◯
CHRISTMAS	nmap -sX	host	◯
SYN	nmap -sS	net	◯
FIN	nmap -sF	net	◯
NULL	nmap -sN	net	◯
CHRISTMAS	nmap -sX	net	◯

it does not require to keep every flow's record, it could scale to the online detection of very high speed network. Despite its obvious value, several limitations still exist and is worth of further study. Firstly, RST scans may produce false positive. Since no flow record is kept, if a malicious attacker send a large volume of RST packets to an innocent system, this system may be labelled as attacker. Secondly, sometimes attacks with spoof address may also generate RST packets destined to other hosts. More research is being continued to resolve this two problems.

Acknowledgment

This research was supported by the research Program of China (863) under contract number 2005AA112130.

References

1. Moore, D., Voelker, G., and Savage, S.: Inferring internet denial of service activity. In USENIX Security Symposium (2001).
2. Martin Roesch: Snort. http://www.snort.org.
3. Paxson, V. Bro: A system for detecting network intruders in real-time. In Computer Networks, 31(23-24), pp. 2435-2463 (Dec. 1999).
4. Fyodor: nmap manual page, http://www.insecure.org/nmap/
5. Barford, P., Kline, J., Plonka, D., and Ron, A.: A signal analysis of network traffic anomalies. In Proceedings of ACM SIGCOMM Internet Measurement Workshop (Nov. 2002).
6. Krishnamurthy, B., Sen, S., Zhang, Y., and Chen, Y.: Sketch-based change detection: methods, evaluation, and applications. In Proceedings of the conference on Internet measurement conference (2003), ACM Press, pp. 234-247.
7. Staniford, S. J.: Containment of scanning worms in enterprise networks. In Journal of Computer Security (Nov. 2003).
8. Gill, T. M., and Poletto, M. MULTOPS: a data-structure for bandwidth attack detection. In USENIX Security Symposium(2001).
9. Heberlein, L. T., Dias, G. V., Levitt, K. N., Mukherjee, B.,J.Wood, and D.Wolber.: A network security monitor. In Proc.IEEE Symposium on Research in Security and Privacy (1990), pp. 296-304.

10. Jung, J., Paxson, V., Berger, A., and Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing. In Proceedings of IEEE Symposium on Security and Privacy (2004).
11. Leckie, C., and Kotagiri, R.: A probabilistic approach to detecting network scans. In Proceedings of the Eight IEEE Network Operations and Management Symposium (Apr.2002).
12. Staniford, S., Hoagland, J. A., and McAlerney, J. M.: Practical automated detection of stealthy portscans. In Proceedings of the 7th ACM Conference on Computer and Communications Security (2000).
13. Yaar, A., Perrig, A., and Song, D. Pi: A path identification mechansim to defend against ddos attacks. In Proceedings of the IEEE Symposium on Security and Privacy (2003).
14. Ramana Rao Kompella, Sumeet Singh, George Varghese: On Scalable Attack Detection in the Network. In ACM SIGCOMM 2004
15. George Casella, Roger L. Berger: Statistical Inference, pp 467-511. Duxbury 2002.
16. Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A.,and Zamboni, D.: Analysis of a denial of service attack on tcp. In Proceedings of IEEE Symposium on Security and Privacy(May 1997).
17. Hyperion hyperion@hacklab.com: Watcher, Phrack53-11
18. Solar designer solar@false.com: Designing and Attacking Port Scan Detection Tools, phrack53-13
19. DARPA: Intrusion Detection Evaluation datasets. URL: http://www.ll.mit.edu/IST/ideval/index.html;.

An IP Address Anonymization Scheme with Multiple Access Levels

Qianli Zhang and Xing Li

Tsinghua University, Beijing 100084, China
zhang@cernet.edu.cn

Abstract. Real world traffic traces are important for Internet research, but public available traffic traces are rare for privacy concerns. IP address anonymization may serve to avoid privacy issues. There are many IP address anonymization schemes according to different requirements and trustworthy levels of the expected users. However, anonymized traces often have to address several groups of researchers at the same time, each with a distinct trustworthy level. Previously known IP address anonymization schemes have to be applied separately to form multiple copies each corresponding to a scheme. In this paper, we propose a scheme which will anonymize the original trace into one single trace, and with different knowledge (secret key) users may recover different traces from it.

1 Introduction

Real-world Internet traffic traces are important for network research such as workload characterization, traffic engineering, web performance, and more generally network performance analysis and simulation. However, most ISPs are reluctant to share their traffic traces and only a few traffic traces (e.g., by NLANR/MOAT Network Analysis Infrastructure (NAI) project [1] and ACM ITA project [2]) are freely distributed. One major reason why traffic trace owners hesitate to make the traces publicly available is the concern that the confidential (commercial) and private information regarding the senders and receivers of packets may be inferred from the trace. In cases where a trace has been made publicly available, the trace is typically subjected to an anonymization process[3][4][5][6] before being released. IP address anonymization is one of the most important part in this process.

There are many anonymization methods available. A straightforward approach to anonymize IP addresses is to map each distinct IP address appearing in the trace to a randomly selected 32-bit address. The only requirement is that this mapping be one-to-one. Anonymity of the IP addresses in the original trace is achieved by not revealing the random one-to-one mapping used. Since such anonymization does not retain the prefix relationships among the IP addresses, it can not be used in research where such relationship is important (e.g., routing performance analysis, or clustering of end systems). Prefix preserving anonymization scheme, on the other hand, will try to preserve such relationship

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 793–802, 2006.

between the anonymized addresses. It demands that if two original IP addresses share a k-bit prefix, their anonymized mappings will also share a k-bit prefix. Between these two extremes, there are also some other anonymization schemes defined. For example, most research requirements will be satisfied to map the first 24 bits randomly while remain the last octet unchanged.

The choice among these anonymization schemes is often limited by the trustworthy level of the expected trace users. However, in many situations, the same trace may be of interest to multiple groups of research users, each with a distinct trustworthy level. For example, an ISP may monitor a link and provide the researchers in the organization with the original traces, while provide researchers who have signed NDA with prefix-preserving anonymized traces, and provide all other researchers with one-to-one mapped traces. It is very cumbersome to provide these traces separately since traces are often very large. Besides, it is often desired that research results from the anonymized traces can be reflected back to the original traces by ISP, i.e. ISP can recover the anonymized IP addresses back to the original one.

Also it is often important for the mappings to be consistent among multiple traces, i.e., the original IP address is mapped into the same anonymized address in different traces. It is because: firstly, if the traffic anonymization process is intermittent, traces from different intervals may take different mappings for inconsistent schemes, thus make these traces unusable for consistent research; secondly, there is a real need for simultaneous (yet consistent) anonymization of traffic traces in different sites, e.g., for taking a snapshot of the Internet. It would be very cumbersome if hundreds of traces have to be gathered first and then anonymized in sequence.

In this paper, we will propose an IP address anonymization scheme with multiple access levels. Users with different access levels may recover from a common trace to different traces with different keys. The rest of this paper is organized as follows. In section 2 we briefly introduce related works, including the operation of TCPdpriv and Crypto-pan. In section 3 we describe our scheme in details. In section 4, we evaluate the security threats to IP address anonymization. The paper is concluded in section 5.

2 Related Works

2.1 TCPdpriv

TCPdpriv is a program developed by Greg Minshall[7] and further modified by K. Cho[8] to eliminate confidential information from packets collected on a network interface (or, from trace files created using the -w argument to tcpdump). It implements several schemes to anonymize IP addresses with different security levels. Level 0 maps different addresses to integers (counting from 1). Level 1 maps the upper and lower 16 bits, separately, to integers (counting from 1); the upper and lower maps are independent. Level 2 maps each byte of the address separately; each byte map is independent. Level 50 is prefix-preserving anonymization scheme.

TCPdpriv can be viewed as a table-based approach: it stores a set of $< raw, anonymized >$ binding pairs of IP addresses to maintain the consistency of the anonymization within one trace. The binding is generated randomly when a new address is anonymized. We refer readers to the source code of TCPdpriv for the actual data structure and algorithm. Despite the elegance and simplicity of the TCPdpriv implementation, it is not consistent: the mappings are determined by the raw IP addresses and the relative order in which they appear in a trace. Therefore, a raw address appearing in different traces may be mapped to different anonymized addresses by TCPdpriv, hence the inconsistency. Also, TCPdpriv can only use one scheme at the same time.

2.2 Crypto-pan

Crypto-pan[9] is a deterministic prefix preserving mapping function from raw addresses to anonymized addresses and is further applied in netflow address anonymization by Wang[10]. With the same key, it can anonymize traffic traces consistently. This algorithm is based on the Canonical Form Theorem[11]:

Theorem 1 (Canonical Form Theorem). *Let f_i be a function from $\{0,1\}^i$ to $\{0,1\}$, for $i = 1, 2, \ldots, n-1$ and f_0 is a constant function, Let F be a function from $\{0,1\}^n$ to $\{0,1\}^n$ defined as follows. Given $a = a_1 a_2 \ldots a_n$ let $F(a) = a'_1 a'_2 \ldots a'_n$, Where $a'_i = a_i \bigoplus f_{i-1}(a_1, a_2, \ldots, a_{i-1})$, and $\bigoplus$ stand for the exclusive-or operation, for $i = 1, 2, \ldots n$. We claim that (a)F is a prefix-preserving anonymization function and (b) A prefix-preserving anonymization function necessarily takes this form.*

Let f_i be: $f_i(a_1 a_2 \ldots a_i) := L(R(P(a_1 a_2 \ldots a_i); K)), i = 0, 1, \ldots, n-1$, where L returns the least significant bit, R is a pseudorandom function or a pseudorandom permutation (i.e., a block cipher) such as Rijndael[12], and P is a padding function that expands $a_1 a_2 \ldots a_i$ into a longer string that matches the block size of R. K is the cryptographic key used in the pseudorandom function R. Its length should follow the guideline (e.g., between 128 and 256 bits in 32-bit steps in Rijndael) specified for the pseudorandom function that is actually adopted. Since the cryptography based anonymization function is uniquely determined by K, same address appearing in two different traces will be mapped to the same anonymized address if the same key is used. Thus it is a consistent prefix-preserving anonymization scheme. Crypto-pan can only anonymize IP addresses with prefix-preserving scheme.

3 Multiple Access Level IP Address Anonymization

3.1 Architecture

The architecture of this scheme is indicated in Fig.1. The original address A is first anonymized with scheme S_1^a. The result is then anonymized with scheme S_2^a to A'. For users without more information, the trace is anonymized with scheme S_2^a. For users with K_2, they can recover the traces back to traces anonymized

with scheme S_1^a. For users with key K_1, the traces can be further recovered back to the original form. K_2 can be derived from K_1 with a cryptographic secure one way function like HMAC[13]. This architecture can be further modified to include more schemes. However, in most cases, two schemes are enough. Although not required by the architecture, it is often suggested that the scheme S_{i+1} provides less information than scheme S_i. For example, the first scheme is prefix preserving anonymization scheme, while the second scheme is the one to one anonymization scheme.

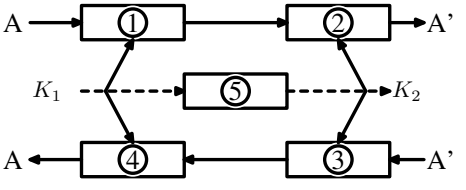


Fig. 1. Architecture

To implement this architecture, the anonymization scheme must be recoverable, i.e., the anonymization function map is a bijection and the recovering is computationally feasible. Most anonymization schemes currently do not meet this requirement. For example, tcpdpriv is hard to recover back since the anonymization process is defined with a randomly generated table. Crypto-pan does not discuss the recovering process either. The following subsections will provide more information on the design of anonymization schemes. Especially we are interested to build recoverable consistent anonymization scheme.

3.2 Prefix Preserving Anonymization Scheme

First we will establish the following result.

Lemma 1. *A prefix preserving address anonymization function is a bijection.*

Proof. Firstly, we will prove that prefix preserving address anonymization function is one-to-one function. i.e., if for addresses $a \neq b$, their correspondent prefix-preserving anonymized addresses are a' and b', then $a' \neq b'$. In other words, if $a' = b'$, $a = b$.

Assume $a = a_1 a_2 \dots a_{32}$ and $b = b_1 b_2 \dots b_{32}$, have the same anonymized address $c = c_1 c_2 \dots c_{32}$. A prefix-preserving anonymization function necessarily takes the form presented in Canonical Form Theorem, thus $c_1 = a_1 \bigoplus f_0$, $c_1 = b_1 \bigoplus f_0$, $a_1 = c_1 \bigoplus f_0 = b_1$.

If $a_1 a_2 \dots a_{i-1} = b_1 b_2 \dots b_{i-1}$, since

$$\begin{aligned} c_i &= a_i \bigoplus f_{i-1}(a_1, a_2, \dots, a_{i-1}) \\ c_i &= b_i \bigoplus f_{i-1}(b_1, b_2, \dots, b_{i-1}) \\ a_i &= c_i \bigoplus f_{i-1}(a_1, a_2, \dots, a_{i-1}) \\ &= c_i \bigoplus f_{i-1}(b_1, b_2, \dots, b_{i-1}) \\ &= b_i \end{aligned} \tag{1}$$

Thus $a = b$. Since prefix preserving anonymization function is 1-1, its domain and codomain are both finite sets of the same size, prefix preserving anonymization function is a bijection. □

The above lemma also defines the recovering process. That is:

Lemma 2. *For an IP address $a = a_1a_2 \dots a_n$ is mapped to IP address $a'_1a'_2 \dots a'_n$ with the process defined by Canonical Form Theorem, with the knowledge of f_i, $i = 0, 1, \dots n-1$, the raw address could be recovered by the following process:*

$$\begin{aligned} a_1 &= f_0 \bigoplus a'_1 \\ a_i &= f_{i-1}(a_1, a_2, \dots, a_{i-1}) \bigoplus a'_i \end{aligned} \tag{2}$$

Proof. According to Cannonical Form Theorem, given $a = a_1a_2 \dots a_n$, $a'_i = a_i \bigoplus f_{i-1}(a_1, a_2, \dots, a_{i-1})$. Thus

$$\begin{aligned} & f_{i-1}(a_1, a_2, \dots, a_{i-1}) \bigoplus a'_i \\ &= f_{i-1}(a_1, a_2, \dots, a_{i-1}) \bigoplus a_i \bigoplus f_{i-1}(a_1, a_2, \dots, a_{i-1}) \\ &= a_i \end{aligned} \tag{3}$$

□

The recovery process of prefix-preserving scheme has the same computation complexity with the anonymization process.

3.3 One to One IP Address Anonymization and Recovery

The one to one IP address anonymization scheme can be considered to be a 32-bits block cipher, which "encrypted" an 32 bits IP address into another 32 bits block, the "decryption" of the result is the recovering process. Any block cipher can be such an anonymization/recovering process. In this paper, the RC5-16/20/32[14] is used. RC5 cipher was invented by Professor Ronald L. Rivest of the Massachusetts Institute of Technology in 1994. It is a very fast and simple algorithm that is parameterized by the block size, the number of rounds, and key length. These parameters can be adjusted to meet different goals for security, performance, and exportability. There have not been public known serious flaws found since its invention. The RC5 block cipher has a word-oriented architecture for variable word sizes w = 16, 32, or 64 bits. It has an extremely compact description, and is suitable for hardware or software. The number of rounds r and the key byte-length b are also variable. It is successively more completely identified as $RC5-w$, $RC5-w/r$, and $RC5-w/r/b$. In this paper, $w = 16$, $r = 20$ and $b = 32$.

Similarly, to anonymize the first 24 bits, we can also use a block cipher. There are two approaches for 24 bits anonymization, the first approach is to use 24 bits block ciphers, the second approach is to use 32 bits block cipher until the result meets some predefined requirements. For example, padding the 24 bits of address with 8 bits 0 to form a 32 bits block, encrypting it until the last 8 bits of result are also 0. The second approach is slower since in average 128 rounds are required to make the last 8 bits zero. In this paper a modified RC5 cipher is used to encrypt and decrypt in 24 bits block.

3.4 Consideration for Special Addresses

It is possible for some address to be anonymized to IPv4 special address like 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255, 192.168.0.0-192.168.255.255 and 224.0.0.0-255.255.255.255. Since special address often have special semantics, to avoid this, we demand:

1. An address that is mapped to a special address will be anonymized until the result is not a special address. It is may be of interest whether such a process will result in a loop in the special address range. In fact, it is impossible for a bijection. Consider an address $a^{(0)}$ is mapped to a special address $a^{(1)}$, and subsequently $a^{(1)}$ is mapped to special address $a^{(2)}$, ..., $a^{(i-1)}$ is mapped to special address $a^{(i)}$, $a^{(1)} \neq a^{(2)} \ldots \neq a^{(i)}$, if address $a^{(i)}$ is mapped to address $a^{(j)}, 1 \leq j \leq i$, then both $a^{(j-1)}$ and $a^{(i)}$ will be mapped to $a^{(j)}$, thus the conflict.
2. Special addresses will not be anonymized.

3.5 Implementation and Experiment Results

The software has been successfully implemented and the source code can be freely downloaded from https://sourceforge.net/projects/ipanon, including a fast prefix preserving anonymization algorithm. To evaluate the proposed anonymization scheme with multiple access levels, we use the traffic traces captured from WIDE in Feb 27, 2003. The traffic is stored in pcap format and contains a total of 364,483,718 packets. After gzip compression, the traffic traces occupy about 10G disk space. The proposed multiple access levels scheme is applied on it. Result is shown in the following table. With a modified prefix preserving algorithm, the speed can be further accelerated.

Table 1. Experiment results for anonymization with multiple access levels

	Crypto-pan and first 24 bits	Fast prefix preserving and first 24 bits
Time	23443.5s	9,093s

4 IP Address Anonymization Scheme Security Analysis

4.1 Semantic Based Attack Analysis

All IP address anonymization schemes are faced with two kinds of security threats, cryptographic attacks and semantic based attacks. To protect the user's privacy, anonymization scheme should be thoroughly designed to protect from the cryptographic attacks. Cryptographic attack means, aided by the knowledge of the compromised raw anonymized address pairs, the intruder may try to infer the cryptographic key used in the anonymization algorithm using all possible cryptanalysis techniques. In this scheme, this threat is considered in design.

Semantic based attacks pose a more practical threat to this scheme. Semantic attack means[15], an intruder is assumed to have compromised(gain full knowledge to) the bindings between certain number of raw and anonymized address pairs through means other than compromising the key. The semantic attack is common to all IP address anonymization schemes. According to the knowledge attackers has, several categories of attacks are possible.

1. Provided attackers could inject packets into the traces, for example, the traces is consistently anonymized in a long period, and attackers are from the monitored network or attackers know the IP address range of the monitored network. In this scenario, the most serious threat is the salting attacks. By using pathological flag, and header field combinations, attackers can inject packets that could be easily recovered and thus an arbitrarily large number of known IP addresses can be obtained. Also, active attacker could generate a sequential scan of the addresses to the specific IP address range to resolve the the mapping for that network. Since most networks today experience frequent scans, this kind of attack is often very difficult to prevent.
2. Attackers cannot inject packets into the trace, however, attackers know the ip address space where the traces are captured. It is often the scenario when traces are anonymized only once or in random time slot. There are still chances to recover the bindings. Firstly, attackers could make use of the frequency analysis. It has been shown that in practical networks, a small number of hosts may account for a large portion of traffic. If the monitored network is fairly sparse, it may be relatively easy to map active hosts based on the services they use or provide. For example, if there is busy public servers like DNS servers or WWW servers, it is fairly easy to find them in anonymized traces. Secondly, attackers could probe all the hosts in the original networks. The information from active probe and the analysis from captured traces can be further compared to gain the binding relationship. For example, since TCP or UDP port number is seldom anonymized (thus will make the traces undesirable in research), the binding information can be easily inferred from particular ports open. More advanced technique, like comparison through the passive os fingerprint[16] and nmap OS fingerprint[17], could also be applied to the anonymized traces analysis. Some rare systems could be spotted in the anonymized tracese easily.
3. If attackers could not inject packets into the traces and the only information known is the ISP the traces from, attackers have to guess the specific IP ranges the traces are from. Typically it is not very easy.
4. Attackers have no information about the traces. For this category, it is often very hard to even recover some pairs of $< raw, anonymized >$ addresses.

Suggestions when providing consistent anonymized traces:

1. Capture traffic from intranet with private IP addresses if it satisfies the specific research requirements.
2. Ensure that users of the monitored network are trustworthy and do not mount attacks on the anonymized traces.

3. Provide minimum information of the original network. Do not release the IP address range or ISP of the original traces if possible.
4. Ensure there is no special systems or obvious servers in the original network. For example, there is no well-known servers or some rare operating systems which could be fingerprinted easily.

We have noted that some well known anonymized traces, like that provided by MAWJ[8], have also taken some of above considerations.

4.2 Threats Specific to Prefix Preserving Anonymization Schemes

Since prefix-preserving schemes introduce more limitations, more information may be inferred from the correlation of the anonymized IP addresses. We demonstrate that with some IP address' mapping information known, it is still difficult to compromise another random IP address.

Consider a set Θ of $N, N \geq 1$ compromised $< raw, anonymized >$ prefix-preserving anonymized IP address pairs, we will study two typical scenarios. One is the address pairs are chosen deliberately and the other are chosen randomly.

For randomly chosen IP address pairs, each IP address is compromised with the same probability. For a random IP address a, let the probability of exactly first l bits prefix can be inferred is $p_\Theta(l)$, this implies that the longest prefix match with a in Θ is $l-1$. For a single IP address, the probability is $p\{l\} = 1/2^l$, the probability $P\{l \leq n\} = \sum_{l=1}^{n} 1/2^l = 1 - 1/2^n$. Thus $P_\Theta\{l \leq n|\Theta\} = (1 - 1/2^n)^N$, $0 < n < 32$.

For the deliberately chosen IP address pairs, the set is constructed as follows to ensure minimal IP addresses are required to completely compromise the first m bits' mapping.

1. Compromise a randomly chosen IP address $< raw, anonymised >$ pairs.
2. Given a set of compromised $< raw, anonymised >$ pairs, the next selected raw IP address would share at most $[log_2 n]$ bits with any pair in the set, where n is the number of compromised pairs and $[x]$ means the integer no more than x
3. Continue 2 until a predefined number N pairs are generated.

Let $m = [log_2 N], M = 2^m$, since there must be at least one IP address with first m bits matched, $P_\Theta\{l \leq n|\Theta\} = 0, n \leq m$. For $n > m$, consider the process in two steps, for the first step the left $N - M$ pairs are distributed evenly in M prefixes, the second step is like the previous scenario (randomly chosen). Thus the probability is:

$$\begin{aligned} P_\Theta\{l \leq n|\Theta\} &= \sum_{i=0}^{N-M} \binom{N-M}{i} (1/M)^i (1 - 1/M)^{N-M-i} (1 - 1/2^{n-m})^{i+1} \\ &= (1 - 1/2^n)^{N-M} (1 - 1/2^{n-m}) \\ &m < n < 32 \end{aligned} \tag{4}$$

It is smaller than the randomly chosen scenario, which indicates that it is more probable to infer longer prefix.

To compare these two kinds of scenarios, we implement a simulation with 131072(1024*128) randomly chosen compromised IP address pairs. The number of pairs required to fully compromised the m-bit prefix IP address is shown in table 2. It indicates that the number of pairs required for randomly chosen attacks is considerably more than that required by deliberately chosen attacks. Also, as the prefix length increase, the gap is also increasing. Not all possible permutations of prefix longer than 14 bits can be recovered for randomly chosen attacks with 131072 compromised pairs. Consider this fact, traces after prefix-preserving anonymization can be generally considered to be secure against the semantic attack if suggestions in previous subsection are obeyed.

Table 2. Required pairs number to compromise the specified length prefix

prefix bit length	1	2	3	4	5	6	7	8	9	10	11	12	13	14
deliberately	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192
randomly	1	2	4	18	82	133	255	531	1525	3870	7571	18965	40281	84219

5 Conclusion

In this paper, we propose a new IP address anonymization algorithm which can anonymize the original trace into single trace, and with different knowledge (secret key) users may recover different traces from it. We also study the security threats to general anonymization schemes and specifically to prefix-preserving anonymization schemes. Based on these analysis, we suggest some considerations when providing anonymized traces.

Acknowledgment

This research was supported by the research Program of China (863) under contract number 2005AA112130.

References

1. Tony McGregor, Hanserner Braun, and Jeff Brown, The NLANR network analysis infrastructure, IEEE Communications Magazine, vol. 38, no. 5, pp. 122– 128, May 2000.
2. The Internet traffic archive, http://ita.ee.lbl.gov/, Apr. 2000.
3. Markus Peuhkuri, A Method to Compress and Anonymize Packet Traces, SIGCOMM IMW 2001
4. Ruoming Pang, Vern Paxson, A high-level programming environment for packet trace anonymization and transformation. SIGCOMM 2003
5. A. Slagell and W. Yurcik, Sharing Computer Network Logs for Security and Privacy: A Motivation for New Methodologies of Anonymization, SECOVAL: The Workshop on the Value of Security through Collaboration, held in conjunction with SecureComm, Athens, Greece, September 2005.

6. Yifan Li, Adam Slagell, Katherine Luo, and William Yurcik, CANINE: A Combined Converter and Anonymizer Tool for Processing NetFlows for Security , International Conference on Telecommunication Systems - Modeling and Analysis (ICTSM) , Dallas, Texas, November 17-20, 2005.
7. Greg Minshall, TCPdpriv Command Manual, 1996.
8. K. Cho, K. Mitsuya, and A. Kato, Traffic data repository at the wide project, in Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track, San Diego, CA, June 2000.
9. J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, On the design and performance of prefix-preserving IP traffic trace anonymization, SIGCOMM IMW 2001
10. A. Slagell, J. Wang and W. Yurcik, Network Log Anonymization: Application of Crypto-PAn to Cisco NetFlows, Secure Knowledge Management Workshop, Buffalo, NY, 2004.
11. J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, Prefix-preserving IP address anonymization: measurement based security evaluation and a new cryptography-based scheme, ICNP 2002
12. J. Daemen and V. Rijmen, AES proposal: Rijndael, Tech. Rep., Computer Security Resource Center, National Institute of Standards and Technology, http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, Feb 2001.
13. H. Krawczyk, M. Bellare, R. Canetti, RFC 2104: HMAC: Keyed-Hashing for Message Authentication, February 1997
14. A. J. Menezes, P.C. v. Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press New York, 1997, p. 269.
15. T. Ylonen, Thoughts on how to mount an attack on tpcpdriv's "-50" option, in TCPpdpriv source distribution, 1996.
16. Michal Zalewski, http://lcamtuf.coredump.cx/p0f.shtml
17. Fyodor: nmap manual page, http://www.insecure.org/nmap/

A Compression Method Designed for SMTP over TLS

Daigo Manabe, Shigetomo Kimura, and Yoshihiko Ebihara

Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8573, Japan

Abstract. TLS (Transport Layer Security) is a well-known protocol used to provide authentication and private communication for application protocols like HTTP (Hypertext Transfer Protocol) and SMTP (Simple Mail Transfer Protocol). The specification of TLS lays down that in the TLS record protocol, data transferred from the upper layer can be compressed before it is encrypted. However, since there are certain restrictions on the transfer of an E-mail, such as the need for text encoding of binary data and the recompression of already compressed data, using the compression mechanism on SMTP over TLS may result in a reduction in the compression performance. In order to solve these problems, this paper introduces a compression method specifically designed for use with SMTP over TLS. In network experiments, compression ratios and file transfer times for three kinds of E-mail were observed, with emulated bandwidth restrictions to represent three typical communication media, and the results show that the proposal method works efficiently especially for narrow bandwidth networks.

1 Introduction

Ever since the first implementation of the internet, electronic mail (E-mail) has been one of basic network services. Certain protocols such as SMTP (Simple Mail Transfer Protocol) [2] are typically used to transfer an E-mail message, but such protocols do not incorporate encryption of the data to protect the contents of the E-mail against eavesdropping.

TLS (Transport Layer Security) [3] is a well-known protocol which is used to provide authentication and private communication for application protocols like HTTP (Hypertext Transfer Protocol). It also can be introduced into SMTP to form SMTP over TLS [1] , which is the one of the solutions to overcome the above problem.

The specification of TLS lays down that in the TLS record protocol, data transferred from the upper layer can be compressed before being encrypted. Although no compression algorithm is explicitly specified in the specification, in 2004 DEFLATE [4] was introduced as the first compression algorithm for TLS. When the compression mechanism of TLS is used for SMTP, however, two restrictions on the transfer of an E-mail may degrade the compression performance. First, since general SMTP only allows US-ASCII characters to be transferred, binary data must be encoded into text in SMTP, which increases the transfer

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 803–812, 2006.

size. However, such text encoding is not relevant or necessary for TLS. Moreover, it may result in degradation in the performance of the compression mechanism. Second, some binary data, such as photographs, are already compressed. Since recompressing compressed data does not contributes significantly to an improved compression ratio in general, the operation is a waste of time and resources.

In order to solve these problems, this paper introduces a compression method specifically designed for use in SMTP over TLS. In this method, an E-mail is parsed, based on MIME (Multipurpose Internet Mail Extensions) [6], and the types of the contents are identified. Based on these results, encoded binary data are decoded to reduce the file size, and compressed data are not recompressed a second time, so saving time.

To evaluate the proposed method, network experiments have been carried out on a prototype system. Compression ratios and file transfer times for three kinds of E-mails were observed over three typical communication media, emulated by bandwidth restrictions, and the results show that the proposed method works efficiently, especially for narrow bandwidth networks.

The rest of the paper is structured as follows. Section 2 provides a brief introduction to SMTP and TLS, and explains the problems of compression in SMTP over TLS. Section 3 proposes the new compression method, specifically designed for SMTP. Section 4 describes network experiments, including an evaluation of the results, and Section 5 summarizes the conclusions.

2 SMTP and TLS

This section provides a brief explanation of SMTP (Simple Mail Transfer Protocol) and MIME (Multipurpose Internet Mail Extensions). It then introduces TLS (Transport Layer Security), and shows why the compression mechanism in TLS over SMTP results in degradation in compression performance.

2.1 SMTP

When a user sends an E-mail, the MUA (Mail User Agent) generally constructs the message header and body in accordance with RFC2822 [7], and passes it to the MTA (Mail Transfer Agent) for transfer by a mail transfer protocol such as SMTP [2].

According to RFC2822, only US-ASCII characters can be used in the E-mail header and body. To send binary data in the E-mail, MIME provides mechanisms for specifying the content type of the data and encoding any data into US-ASCII characters. For the latter, Base64 and Quoted Printable are defined as the text encoding methods, as follows.

Base64. Three 8-bit values are divided into four 6-bit values represented by alphanumeric characters and certain other symbols in the US-ASCII character set.

Quoted Printable. The hexadecimal value of an 8-bit octet is used, prefaced by the character =. For example, =80 represents a value 128. Therefore, every

1-byte character, except for alphanumeric or page break codes is encoded into 3 characters.

Furthermore, the multipart facility provided from MIME allows two or more kinds of data to be attached to a single E-mail.

2.2 TLS

TLS is a secure communication protocol, positioned between a trusted transport layer like TCP and application protocols like SMTP. As described in RFC2246 [3], TLS has two main protocols, i.e., a handshake protocol and a record protocol. The handshake protocol is basically invoked at the start of communication to negotiate the compression and encryption algorithms and exchange secret information such as a cipher key between the end peers.

The record protocol actually compresses and encrypts the data based on the algorithms determined by the handshake protocol. For this process, the following three structures are used.

- TLSPlaintext structure
  A message from the application layer is fragmented and stored in TLSPlaintext structure. RFC2246 specifies that the length of the fragmented data should not exceed 16,384 bytes.
- TLSCompressed structure
  The fragmented data in the TLSPlaintext structure is compressed, and stored in TLSCompressed structure. According to RFC2246, the length of the compressed data should not exceed 16,384+1,024 bytes.
- TLSCiphertext structures
  The compressed data in the TLSCompressed structure is encrypted, and stored in TLSCiphertext structures for transmission at the lower layer. RFC2246 specifies that the length of the encrypted data may not exceed 16,384+2,048 bytes.

Fig. 1 illustrates the TLSCompressed structure. The other two structures are omitted, since they have almost same construction.

In the figure, the structure has 5-byte header. The Type field shows the data type in the structure such as application data and messages for the handshake protocols. The Version field contains the version number of TLS, which is currently 3.1. The Length field, which is 16-bits long and so occupies two bytes, gives the length of the data following this field.

2.3 Problems Resulting from the SMTP and TLS Compression Mechanisms

When the compression mechanism of TLS is used for SMTP, the compression performance suffers for the following reasons.

First, for typical implementations of TLS, since the recode protocol fragments the data from the upper layer, the boundaries of the parts of a multipart message

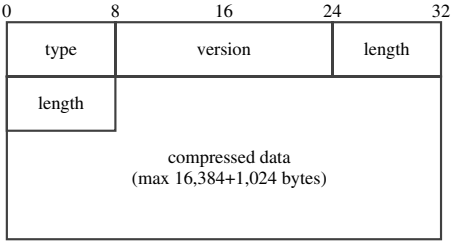


Fig. 1. Format of TLSCompressed structure

and thus the content type of each fragment cannot be recognized. As a result, even when the parts of the message are previously compressed photographs, video, archived data, and so on, these are still compressed again when they are stored in the TLSCompressed structure. However, such recompression serves no purpose, since it cannot further reduce the size. Moreover, the process of recompression may actually expand the data size by appending information about the compression. Therefore, recompression is a waste of time and resources.

Secondly, as mentioned in Section 2.1, binary data must be encoded into US-ASCII characters. For example, Base64 increases the size of the binary data by 33%. However, such text encoding may have a bad effect on the compression ratio. In addition, the text encoding is not relevant or necessary for TLS.

3 Compression Method Specifically Designed for SMTP over TLS

In order to overcome the problems described in Section 2.3, this section proposes a compression method specifically designed for SMTP over TLS. This method does not require any modifications to SMTP. It should also not to MUA and MTA, since TLS can select the proposed compression method when it discriminates the upper layer is SMTP from the destination port number.

3.1 Fragmentation Based on MIME

In RFC2246, although the maximum fragment size is defined, it is possible to select the position where the data is divided. Using this facility, the proposed method analyzes the structure of an E-mail received from the upper layer based on MIME, and fragments the message according to the three types of data, i.e., text data, non-compressed binary data, and compressed binary data.

In the example shown in Fig. 2, the E-mail to be transmitted has three attached files, namely a text message, compressed data, and non-compressed binary data. The compressed data and non-compressed binary data are just extracted and stored in TLSPlaintext structures. If these items of data are too large to store into the structures, they are individually fragmented up to the maximum size.

Before and after each attached file, a boundary string and MIME header relating to the file are inserted. At the top of the E-mail, the mail header also exists. Since all of these can be categorized as text data, such boundary strings and headers can be concatenated with the adjacent text messages. For example, the first item of text data in Fig. 2 includes not just the text message in the E-mail, but also the mail header and the boundary string, and MIME header of the text message, all of which are concatenated. Moreover, the boundary string and MIME header of the next compressed data item are also in text form, and so are concatenated with the first item of text data.

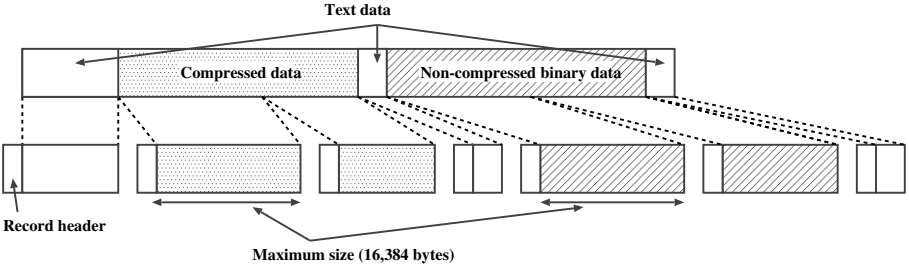


Fig. 2. Fragmentation for each type of data

3.2 Compression Based on Data Types

In the proposed method, each fragment, constructed as described in the previous subsection, is compressed in a manner appropriate for the type of data, before being stored in the TLSCompressed structure. To allow decompression of these compressed fragments at the receiver, an additional 1-byte method field is added at the start of the compressed data, as shown in Fig. 3.

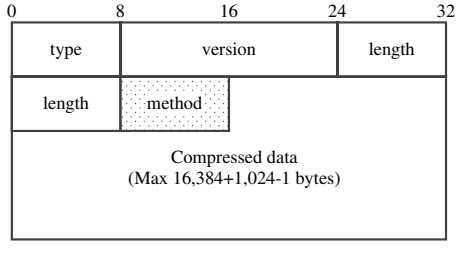


Fig. 3. Addition of Method Field

In the method header, the least-significant four bits indicate the compression algorithm and the most significant four bits shows the text encoding method. In our prototype system, two compression algorithms are defined, as shown in

Table 1, and three text encoding methods, as shown in Table 2. As shown, the prototype system adopts DEFLATE, which is the standardized compression algorithm of TLS, as a compression algorithm, and Base64 and Quoted Printable, which are defined in the specification of MIME, as encoding methods.

Table 1. Compression algorithms

compression algorithms	value
Non-compression	0x00
DEFLATE	0x01

Table 2. Encoding methods

text encoding	value
No encoding	0x00
Base64	0x01
Quoted Printable	0x02

In the proposed method, each fragment in the TLSPlaintext structure should be processed by the compression algorithms as listed below. If the fragment has already been encoded into text, then it should be decoded back into the original binary data before compression is applied. Note that when the SMTP transfer adopts the 8BITMIME option, binary files can be sent without text encoding, and then the decoding process in the compression algorithm can be omitted.

- Text data and non-compressed binary data should be compressed using the DEFLATE algorithm.
- Compressed binary data should not be subject to further compression.

For example, consider a message passed from the upper layer which contains three items of simple text data, one item of text-encoded compressed, and one item of text-encoded non-compressed data as illustrated in Fig. 4. First, these individual items of data are fragmented. All fragments of text data are then simply compressed using DEFLATE. The other text-encoded fragments are first decoded to the original binary data. The non-compressed fragments are then compressed using DEFLATE, but the compressed fragments are not subject to further compression.

In some cases, the compressed fragments are not applied the best compression to reduce compression time or keep higher quality of pictures or videos. Since MIME headers do not give such information, recompressing is required to investigate whether the compressed fragments can be reduced their size or not. Although this paper does not address these cases, recompression options can be applied to the proposed method for highly narrow band networks.

4 Network Experiments

To evaluate the method proposed in the previous section, this section describes network experiments using a prototype system, and reports the compression ratios and the times taken to transfer E-mails.

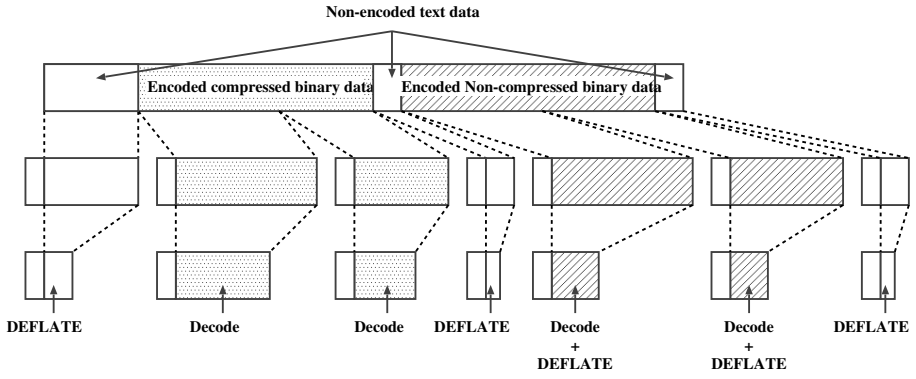


Fig. 4. An example illustrating the compression procedures adopted in the proposed method

4.1 Experimental Conditions

Fig. 5 shows the network environment used for the experiments. For the prototype system, the proposed compression method was implemented in OpenSSL 0.9.7e. For the SMTP server, qmail 1.03 with a TLS patch was used. For the SMTP client, Sylpheed 1.0.0 was used, since it already supports TLS. For the DEFLATE compression algorithm, the zlib 1.2.1 library was adopted. In order to emulate the transfer speeds of three typical communication media, i.e., 28.8 kbps for analog modems, 64.0 kbps for ISDN, and 10 Mbps for a LAN, the mail server used the firewall ipfw2 with a traffic control system dummynet in FreeBSD 5.3-RELEASE-p5.

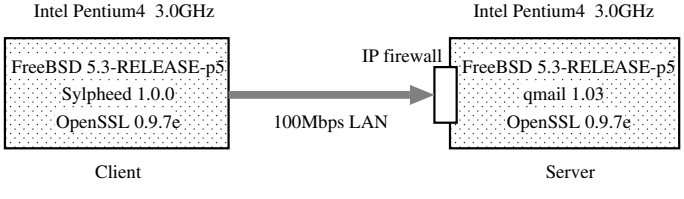


Fig. 5. Network environment for the experiment

For the E-mails to be transferred, three kinds of E-email samples were prepared, as shown in Table 3. In the table, file size means the total size of each E-mail, including the header. The MS-Word file only contained standard text, with no graphics, tables, etc.

4.2 Experimental Results

First, the compression ratio of the proposed compression method was evaluated. Table 4 shows the compression ratio of the three samples of E-mails listed in

Table 3. E-mail samples used in experiments

Type of data	Text encoding	Attached file	File size
Text data	None	Text file	399KB
Non-compressed binary data	Base64	MS-Word file	544KB
Compressed binary data	Base64	JPEG file	578KB

Table 4. Compression ratios achieved for different E-mail samples

Kinds of data	No compression	DEFLATE	Proposed
Text data	399KB (1.0)	128KB (0.31)	128KB (0.31)
Non-compressed binary data	544KB (1.0)	104KB (0.19)	59KB (0.10)
Compressed binary data	578KB (1.0)	454KB (0.78)	449KB (0.77)

Table 3. For the compression algorithm in the TLS record layer, the three methods compared are: no compression, compression using the DEFLATE algorithm, and the proposal method. In the table, values in parentheses shows compression ratios compared to the original file size.

From Table 4 it may be seen that not only the text data and non-compressed binary data, but also the compressed binary data are reduced in file size by using the proposed method. The difference between the DEFLATE method and proposal method is mainly the existence of text decoding in the compression procedures. With text data and with compressed binary data, there is little difference between the DEFLATE method and the proposed one. However, in the case of the MS-Word file, the compression ratio obtained with the proposed method is twice that obtained with the DEFLATE method. From these results we can conclude that although the compression ratio of text data and compressed binary data is not influenced by Base64 text encoding, in the case of non-compressed binary data text encoding does degrade the achievable compression ratio.

Next, the average transfer time was observed for the three kinds of E-mail sample. Tables 5–7 show the results obtained when the E-mails were transferred over three types of communication media. The transfer time was measured from the beginning of each TLS session for sending a mail sample to the end of the TLS session after sending the mail sample. For all TLS sessions, the digital signing algorithm used was RSA, the public-key encryption algorithm was AES, and the secure hash algorithm for MAC (Message Authentication Code) was SHA-1. The measurements were executed three times to obtain the average.

From the tables, we can see that average transfer time of the proposed method is better than that of the method in which no compression was applied. These improvements are particularly remarkable when the communication media has a narrow bandwidth i.e., 28.8 kbps and 64.0 kbps. In particular, for the non-compressed binary data, the average transfer times using the proposed method are reduced by a factor of nearly 10, compared to that of the œB!Hœ($Bnocompression$œB!Iœ(B method. However, these differences are small

Table 5. The average transfer time of E-mail (28.8Kbps)

Kinds of data	No compression	DEFLATE	Proposed
Text data	158.17 s	49.50 s	49.48 s
Non-Compressed binary data	212.61 s	40.86 s	24.20 s
Compressed binary data	225.71 s	171.30 s	165.07 s

Table 6. The average transfer time of E-mail (64.0Kbps)

Kinds of data	No compression	DEFLATE	Proposed
Text data	56.89 s	17.90 s	17.83 s
Non-Compressed binary data	76.47 s	14.74 s	8.78 s
Compressed binary data	81.20 s	61.65 s	59.42 s

Table 7. The average transfer time of E-mail (10Mbps)

Kinds of data	No compression	DEFLATE	Proposed
Text data	0.60 s	0.26 s	0.26 s
Non-Compressed binary data	0.64 s	0.25 s	0.22 s
Compressed binary data	0.69 s	0.53 s	0.56 s

when the transmission media is a LAN, because the average transfer time itself is much shorter.

Compared with the general use of DEFLATE compression, the average transfer time of the proposed method for text data and compressed binary data is a little less than or almost same to that of the DEFLATE method. However, the average time of proposed method for compressed binary data in Table 4 is 0.03s larger, since the parsing process in the prototype system took longer time than the compression process in the DEFLATE method. By tuning the parsing program, the average time of the proposed method should be smaller. In the cases of the non-compressed binary data, the proposed method shows a reduction for narrow bandwidths of nearly a half compared with that obtained using DEFLATE, since the compression ratio in the proposed method is also about half of that of DEFLATE, as shown in Table 4. From the results, the effectiveness of the proposed method is shown.

5 Conclusions and Future Work

This paper has proposed a compression method specifically designed for SMTP using the framework of compression in TLS. The results of the transmission experiments indicate that the proposed method can achieve a higher compression ratio and the shorter average transmission time than the DEFLATE method, especially for non-compressed data like attached word-processor files. The improvement is especially marked when the transmission bandwidth is narrow.

Broadband spread in a city, but narrowband user remain globally. However, for the measurements of transmission time, only one TLS session was used for each experiment. We need to measure the processing load of the proposed method in order to estimate how many sessions can be handled simultaneously by the prototype system.

The proposed method can use other compression algorithms, which differ from DEFLATE, for text data and non-compressed data. Making use of this fact, it is expected that the compression ratio may be further improved by using a static dictionary [8] to include words which appear frequently, such as tags in HTML text. Moreover, the proposed method can be improved so that it can also to be applied to other mail transfer protocols such as IMAP and POP.

As shown in Section 6 in RFC3749 [5], combining compression with encryption can sometimes reveal information, since the length of the compressed data might provide some information to eavesdroppers. As long as the framework of compression in TLS, i.e., fragmentation before compression, is retained, it seems to be too difficult to hide such information. In the future, the authors plan to propose the new compression method to fit the length of the compressed data to the maximum data size by introducing a new structure after TLSCompressed.

Acknowledgment

This work was partly supported by MEXT KAKENHI (16700049).

References

1. P. Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security," RFC3207, February 2002.
2. J. Klensin, "Simple Mail Transfer Protocol," RFC2821, April 2001.
3. T. Dierks and C. Allen, "The TLS Protocol Version 1.0," RFC2246, January 1999.
4. P. Deutsch, "DEFLATE Compressed Data Format Specification Version 1.3," RFC1951, May 1996.
5. S. Hollenbeck, "Transport Layer Security Protocol Compression Methods," RFC3749, May 2004.
6. N. Freed and N. S.Borenstein "Multipurpose Internet Mail Extensions," RFC2045 to RFC2049, November 1996.
7. P. Resnick, "Internet Message Format," RFC2822, April 2001.
8. N. Okamoto, S. Kimura, and Y. Ebihara, "An Introduction of Compression Algorithms into SSL/TLS and Proposal of Compression Algorithms Specialized for Application Protocols," Proceedings of AINA2003, pp. 817–820, March 2003.

Applications and Services

Design of a Video Door Phone Service Providing Personal Mobility Based on Home Gateway System

Yeon-Joo Oh, Eui-Hyun Paik, and Kwang-Roh Park

Ubiquitous Home Service Research Team, Digital Home Research Division,
Electronics and Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, Korea
{yjoh, ehpaik, krpark}@etri.re.kr

Abstract. Though typical video door phone systems can be used to talk to the visitors at the door, they have some shortcomings: the resident should stay in house and he/she could not be informed of any visitor information on his/her return to home. To overcome these limitations, we propose an architecture supporting user mobility, with which the resident at anywhere can view and converse with a visitor at the front door using PC's, PDA's and home servers with internet connections. We implemented the proposed architecture using SIP(session initiation protocol) and a home gateway system connected to a conventional intercom or a video door phone system.

1 Introduction

Based on the home networks connected to the broadband networks, it is already possible to access and control home appliances from remote sites on the internet. For this purpose, they typically use integrated control solutions that enables its user to manage home appliances from inside home locations such as home servers, home pads, wall pads and PC's and also from outside home locations such as his/her mobile phones, office PC's etc.[1]

In this paper, we suggest an enhanced video door phone service architecture combining typical video door phone services with the home network facility. Existing video door phone solutions can be classified into two categories. One is the traditional intercommunication system: it only provides a simple video phone connection between the front door location and the interphone device in the living room. The other is more recently introduced video door phone systems with home network supports. In addition to the simple video phone connection, it also provides communications to the pre-registered location such as the resident s mobile phone or security office when the resident is absent. Though this type of service is more flexible, the resident should set the telephone number of his/her wireless communication device or the security office, before his/her out-going.

The limitations of these conventional video door phone systems can be summarized as follows: First, the resident can converse with the visitor using only

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 815–822, 2006.

the video phone monitor device in the living room, even though he/she already have home network-connected video facilities such as digital TV's, PC's and so on. Second, the registered phone number cannot be changed dynamically from an outside home location.

To remove these limitations, we propose the Adaptive Video Door Phone Service (AVDPS) architecture. Using this service, residents can converse with visitors from any locations inside or outside home at anytime, through the home gateway system and SIP (session initiation protocol)[2,3,4,5].

When a resident is at home, a door bell ringing event invokes notification messages on the communication-possible devices such as the video phone monitor device in the living room, the home server system in the dining room, the PC in the study room, etc. According to the selection of the preference device by the resident, it establishes a multimedia communication between the selected device and the front door phone camera device. When the resident goes out the house, our architecture connects to the pre-registered terminal device. The registration of the terminal device can be dynamically updated by the resident from time to time. Thus, the resident may use a mobile phone during his/her driving a car, and change the connection to the office PC, when he/she arrives in the office room. Therefore, our AVDPS architecture can achieve the personal mobility and dynamic modification of user-preference on the device at any place, at any time.

2 Adaptive Video Door Phone Service

2.1 Personal Mobility Support Based on SIP

Personal mobility (i.e., user mobility) allows us to address a single user located at different terminals by the same logical address[2,3,5]. 1-to-n(one addrss, many

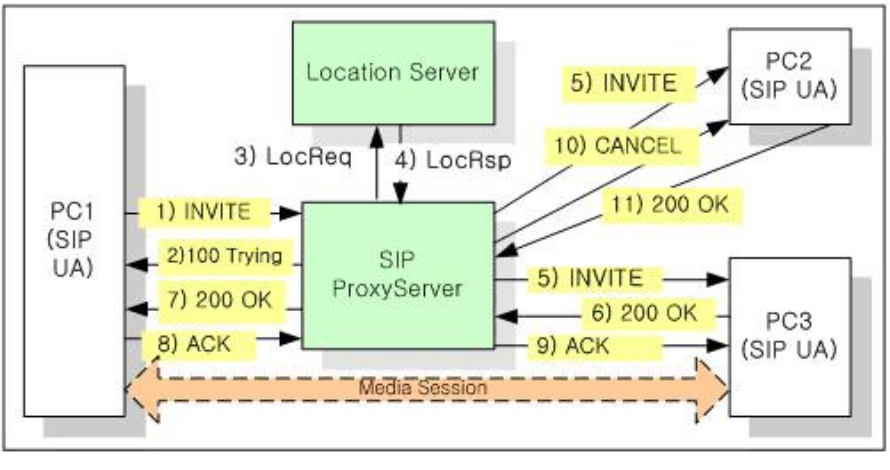


Fig. 1. Message flow of SIP forking proxy mechanism

potential terminals) mappings are useful, as illustrated in Fig. 1. We use the mechanism based on SIP which can provide the user mobility for the resident in the home. For example, a resident may want to be reachable via PC2, and PC3. He/she may use these devices either at the same time or alternate between them. Using a SIP forking proxy, he/she can be reached at any of the devices at the same time, making her device choice transparent to others.

Therefore, we define a homegateway(HG) as the AVDP service entity which performs the SIP server functionalities such as proxy, registrar, and location server.

Also, the HG as a SIP UserAgent(SIP UA) has video/audio codec and performs events for the door phone device and is connected to the internet. Thus, people that reside in the home within the HG could communicate with their preference or the closest terminal to the HG at any time and anywhere. The preference terminals mean devices that the resident has registered to the HG in order to converse with the visitor.

They have a User Terminal Agent based on SIP UA that communicates with the video door phone device connected to the HG using the SIP session control mechanism via the HG and performs functionalities that unlock the front door of his/her house.

2.2 Proposed Architecture

The architecture of an Adaptive Video Door Phone Service consists of components as follows: an SIP Server which control session establishment, Door Phone Agent, and User Terminal Agent. Fig. 2 shows this components and the following will explain each component.

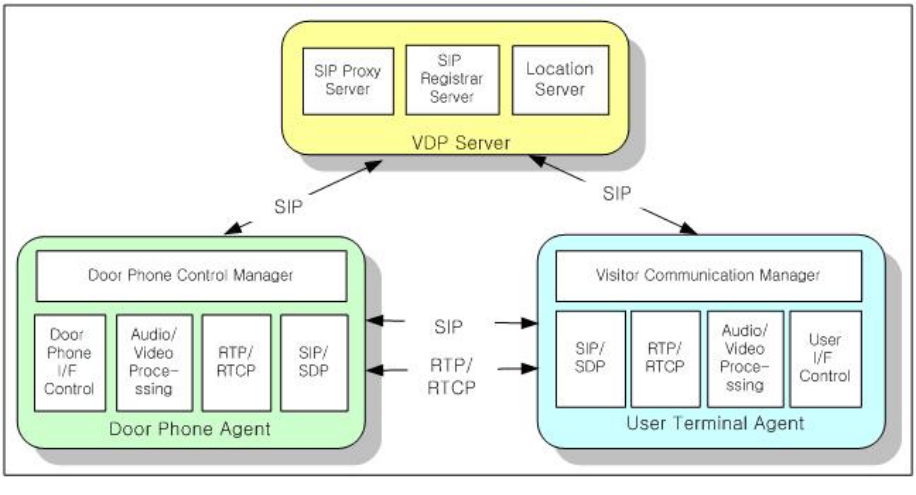


Fig. 2. Components for the Adaptive Video Door Phone Service

Video Door Phone Server (VDP Server): It contains Proxy Server, Registrar Server, and Location Server modules, which provides the services for user registration, user management, user location, call-forward, and user preferences.

Door Phone Agent (DPA): It controls the legacy door phone device via DoorPhone I/F Control and transforms the video/audio signal inputted from the device. It also contains an SIP UA functionality for SIP session control, and an RTP stack for multimedia transmission.

User Terminal Agent (UTA): it has a graphical user interface (GUI) and notifies the resident a door bell event which has been received from Door Phone Agent. It also contains a SIP UA functionality for SIP session control, and a RTP protocol for transmitting encoded video and audio data.

2.3 Usage Scenario

Registration of the selected terminals: A resident launches the User Terminal Agent application based on an SIP UA on his/her preference terminals from inside or out-side his/her home. When the User Terminal Agent application starts up, it automatically registers to the homegateway that acts as the SIP registrar. At this time, the resident may want to be reachable via more than one device, and he/she can register all the devices manually with the "REGISTER" menu of the application.

Event notification and conversation: When a visitor rings the bell at the door, the HG is received the bell signal from the door phone device. The HG generates an SIP INVITE message and sends o the terminals which have been registered by the resident. The resident receives a door bell event from several terminals which are already registered by the resident and clicks the "OK" button in the nearest terminal selected by him. And then, the audio data is exchanged between the resident's terminal and the HG while the video data is transmitted via HG to the resident's terminal.

Opening the Door by the user terminal: The resident confirms the visitor and opens the door by clicking the "DOOR OPEN" button in the User Terminal Agent application. The UTA sends the event message into the Door Phone Agent in the HG which controls the door lock and unlock.

Terminating conversation: By clicking the "CLOSE" button in the User Terminal Agent application by the resident, the UTA generates and sends an SIP BYE message into the Door Phone Agent in the HG. When the Door Phone Agent in the HG is received the SIP BYE message, it sends a SIP OK message to the user terminal. Finally, the multimedia session between them is released.

3 Implementation and Results

The Adaptive Video Door Phone Service architecture proposed in this paper was developed on both of the Linux(kernel 2.4.20) and WinCE 4.0 operating system and could be adopted to home gateway, home server, a PC and PDA, etc. Hardware specifications of the devices are stated in Table 1.

Table 1. Testbed Hardware Specification

Devices	Homegateway	PC	PDA
Functional Component	VDP Server & VDA	UTA	UTA
CPU	533MHz	1.3GHz	624MHz
RAM	128M	1024M	512M
Kernel version	Linux 2.4.20	WinXP	WinCE4.2
Language & SDK	C	VisualC 6.0	Embeded VC 5.0
video codec	MPEG4 Encoder (H/W)	MPEG4 Decoder (S/W)	MPEG4 Decoder (S/W)
Audio codec	G.723.1 /G.711 (S/W)	G.711(S/W)	G.711(S/W)

Fig. 3 shows the system architecture for providing the proposed service. It mainly consists of devices as follows: a traditional video door phone system, a home gateway (HG), user preference terminals such as PC, PDA, and home server that is connected the Internet and view and converse with the visitor at the door.

We defined a homegateway system as a service entity for providing the AVDP service. Therefore, the homegateway system performs the VDP server functionalities such as SIP Proxy, SIP Registrar, and location server.

Also, the system performs a Door Phone Agent functionality which has an SIP UA and video/audio codecs and controls events for the door phone device and is connected to the internet. The preference terminals mean devices that the

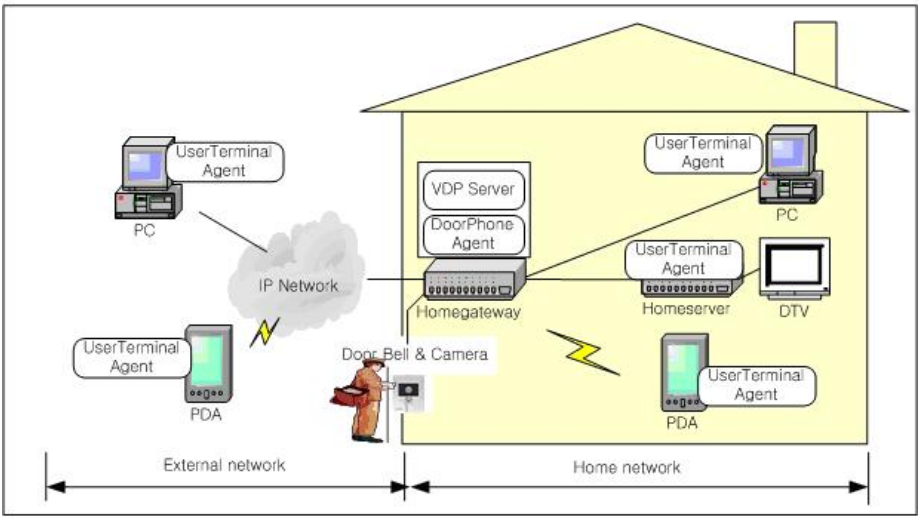


Fig. 3. The system architecture for supporting Adaptive Video Door Phone Service

resident has registered to the homegateway in order to converse with the visitor. They have the User Terminal Agent application based on the SIP UA[5] that communicates with the video door phone connected to the HG and performs functionalities that unlock the front door of his/her house. We connected the physical interface of the existing door phone camera device located at the door to the home gateway which has functionalities that control the door phone system and transmit video and audio signal received from video door phone camera device to the user terminal and vise versa.

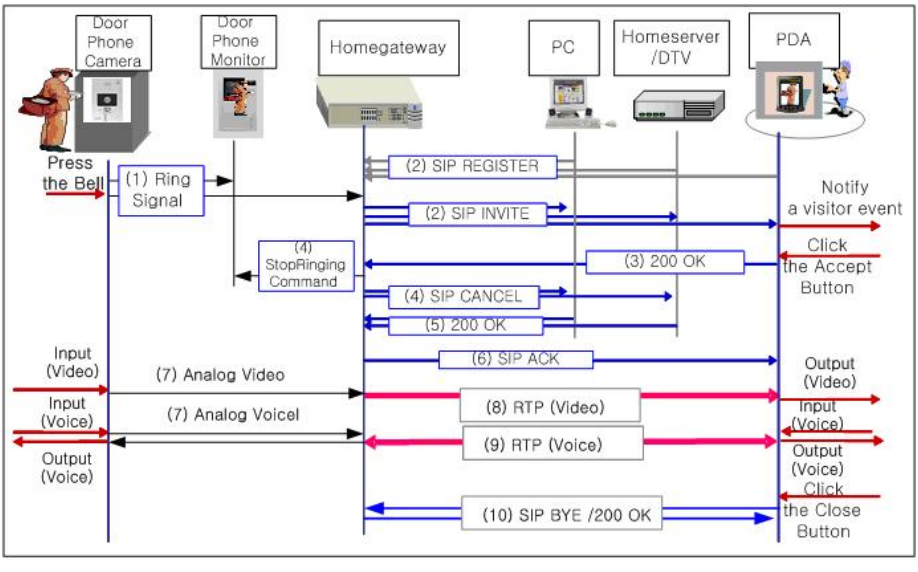


Fig. 4. Message sequence flows for the AVDP Service

In our architecture, the SIP[5] was used as the signaling protocol for multimedia sessions and the RTP[6] was used as the protocol for multimedia streaming between HG and the video door phone camera device or between HG and the user terminal. Fig. 4 illustrates the flow of the messages exchanged between the door phone system and a terminal selected by the visitor, when a visitor rings the bell. The proposed system follows the call processing procedure and message formats specified in IETF RFC 3261[5].

Fig. 5 shows a procedure for door ring event notification and conversation with the visitor at the door using the PDA device as one of user preference terminals.

The performance of the homegateway system, operating as the SIP stateful proxy server forks and forwards the INVITE request to more than one location, depends on the number of the contact addresses as well as the implementation issue. And if the server could not complete forwarding the request to the multiple

Fig. 5. The User Terminal Agent application with a PDA, for door ring event notification and communication

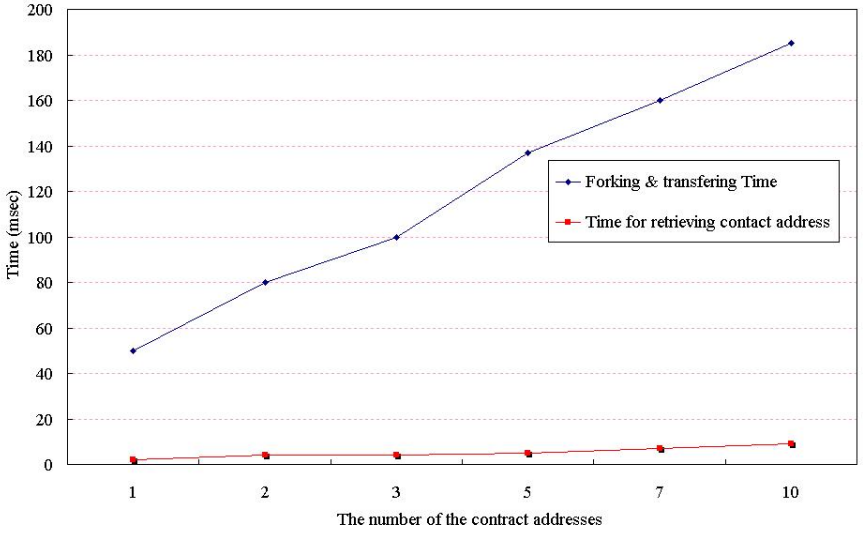


Fig. 6. The processing time for parallel forking and transferring in HG regarding the number of contacted locations

destinations within the limited time, it may affect the transaction state management of the VDA including SIP User Agent Client functionality as a logical entity creates a new request. Therefore, we have evaluated the performance of the homegateway system. The Fig. 6 shows the results for the HG system performance. The processing time that includes in forking and transferring a request packet per each destination is increased slightly, although the number of the contact addresses as the delay factor regarding packet forwarding is increased linearly.

4 Conclusion

We have proposed and implemented an adaptive video door phone service system as the architecture in which a resident can confirm and converse with the visitor at anytime at anywhere, based on SIP using a homegateway system. Proposed architecture has compatibility with existing door phone systems. Also, a user in the home can converse with the visitor not only using the User Terminal Agent application but also using the traditional SIP phone software such as linphone[7] or kphone[7]. At this time, we are planning to extend our architecture to include using RF sensors or Bluetooth protocol in the home[8,9,10].

References

1. http://www.samsung.com/homenetwork/homevitasolutions/
2. Schulzrinne, H., Wedlund, E.: Application-layer mobility using SIP. ACM SIGMOBILE Mobile Computing and Communications Review **4**(3) (2000)
3. El-Khatib, K., Zhang, Z.E., Hadibi, N., v. Bochmann, G.: Personal and service mobility in ubiquitous computing environments. Wireless communications and mobile computing **4** (2004) 595–607
4. Rahman, M., Akinlar, C., Kamel, I.: On secured end-to-end appliance control using SIP. In: Proceedings of the 5-th IEEE International Workshop on Net-worked Appliances. (2002) 24–28
5. Rosenberg, J., Schulninne, H., Camadlo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: session inifidon protocol. RFC 3261, Intemet Engineering Task Force (IETF) (2002)
6. H. Schulninne et. al.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF (2003)
7. http://www.cs.columbia.edu/sip/implementations.html
8. H. Schulninne et. al.: Ubiquitous computing in home networks. IEEE Communications Magazine (2003)
9. Tsang, S., Moyer, S., Marples, D., Schulzrinne, H., RoyChowdhury, A.: SIP Extensions for Communicating with Networked Appliances. Draft, IETF (2000)
10. Rahman, M.: Remote access and networked appliance control using biometrics features. IEEE Transactions on Consumer Electronics **49**(2) (2003)

Exploiting Domain Ontologies and Intelligent Agents: An Automated Network Management Support Paradigm

Sameera Abar[1], Yukio Iwaya[2], Toru Abe[3], and Tetsuo Kinoshita[3]

[1] Department of Applied Information Sciences, GSIS, Tohoku University, Japan
sameera@ka.riec.tohoku.ac.jp
[2] Research Institute of Electrical Communication, Tohoku University, Japan
iwaya@fir.riec.tohoku.ac.jp
[3] Information Synergy Center, Tohoku University, Japan
beto@isc|kino@riec.tohoku.ac.jp

Abstract. This paper presents a domain-ontology driven multi-agent based scheme for representing the knowledge of the communication Network Management System (NMS). The scope of this work is focussed on the performance analysis and fault detection functional areas which are of prime importance as far as the management of the communication network systems is considered. The proposed network knowledge model has been constructed in accordance with the CommonKADS methodology, to facilitate its reusability and shareability. In the proposed knowledge-intensive framework, the static domain-related concepts are articulated as the domain knowledge ontology. The empirical knowledge for managing the network is represented as the fault-state causal reasoning models, and it is explicitly encoded as the core knowledge of multi-agent middleware layer as heuristic production-type rules. This task-oriented experiential knowledge manipulates the domain content and structure during the diagnostic sessions. The inference chains during the agents' cooperative problem solving are supported by the java-based networking routines in conjunction with the run-time log information. The proposed approach can be regarded as one of the pioneered steps towards representing the network knowledge via reusable domain ontology and intelligent agents for the automated network management support systems.

Keywords: Multi-agent System, Domain Ontology, Knowledge Acquisition, Problem Solving, Network Management.

1 Research Overview

Given the scale and complexity of communication networked systems, it's becoming increasingly important that they are able to deal with most of the tasks of network management themselves, intelligently and autonomously. Traditional network management approaches which rely on the direct manual intervention procedures are not appropriate for managing today's network systems, because

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 823–832, 2006.

the all-over human management is not possible owing to time or location constraints. Besides this, managing huge distributed networks in order to ensure that the system operates within desirable parameters, in an extremely cumbersome task and poses many challenges for network administrators. Hence, for the significant administrative overhead reduction and increased robustness, the errors and failures must be worked around to adaptively optimize the normal state of the network itself.

The tasks of an NMS deal with collecting and analyzing information about a network, including both hardware resources (e.g., configuration of a router) and software (e.g., the number of incoming IP-packets to a router). In the manager-agent type of network management paradigm, the attribute values of managed objects are the key to understanding, analyzing and controlling the behavior of a management system. The state changes of the managed objects are reported in the form of SNMP traps or events. This is a complex process as it involves not only the information about the managed network, i.e. topology/configuration, but also knowledge about the format and meaning of each individual event, causal and temporal relationships among events. Thus the pre-requisite for the automation of management functions is the detailed interpretation of network-related knowledge resources.

Traditionally, knowledge engineering was viewed as a process of "extracting" knowledge from a human expert and transferring it to the machine in computational form. Today, knowledge engineering is approached as a modeling activity [1]. The applications are characterized by the tasks and domains involved. Knowledge modeling can therefore be divided into two conceptual sub-activities: modeling the task and domain knowledge [2]. In the proposed work, the characterization of the network knowledge model has been performed in-line with the CommonKADS [3]— a methodology for expertise modeling embraces the application-intensive knowledge in three types as: domain, inference, and task knowledge structures.

Recently, Multi-Agent System (MAS) has emerged as a flexible way to manage the resources of distributed systems. In this paper, MAS-based approach has been adopted to deploy the functionality of the intelligent behavior of proposed network knowledge model. Multi-agent systems are composed of multiple interacting agents where each agent is a coarse-grained computational system in its own right, as well as independently modifiable [4]. Agents, while being well-focused on their automated tasks, provide inherently distributed solutions. The concept is to specialize agent interactions for autonomously and flexibly managing the operational knowledge of network devices thereby reducing the workloads of a network administrator remarkably.

1.1 Motivation

Motivation for this research originated from the need to devise a MAS-mediated and ontology-driven knowledge-based strategy, in support of the automatic provision of just-in-time and just-enough, context-dependent knowledge for actively managing the data communication network systems [5]. So far, a little work has

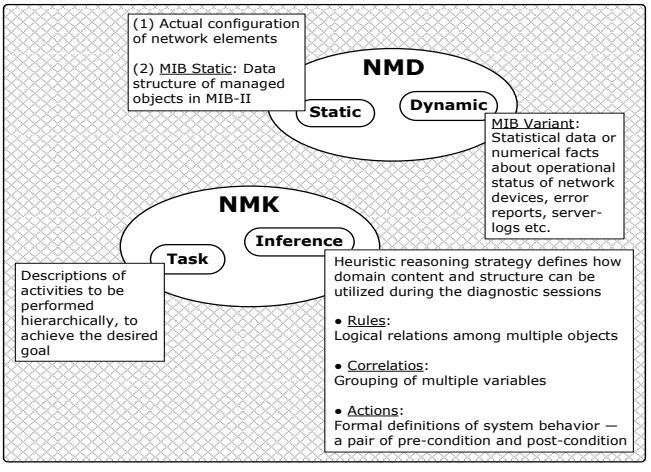


Fig. 1. Network Management Data (NMD) versus Network Management Knowledge (NMK)

been done for managing the operational knowledge of the communication network systems. Hence, the proposed idea can be regarded as an initial step towards the knowledge acquisition, representation, and sharing of widely distributed network knowledge.

Another reason for devising our network knowledge representation scheme stems from the fact that not many knowledge modeling techniques have been developed for the diagnostic technical domains. Building new knowledge-based systems today usually entails constructing new knowledge bases from scratch. It could instead be done by assembling reusable components [6]. Therefore, this work can serve as a test-bed to be reused for various practical diagnostic domains. The proposed approach embodies the network-related knowledge in the form of uniformly represented semantic models, thereby providing a promising mechanism to achieve reusability and maintainability.

1.2 Paper Organization

The paper is structured as follows. Next section outlines the research works most relevant to our approach of the network knowledge representation. Section-3 presents our formalization of the communication network-related knowledge, the actualization of proposed knowledge model with the multi-agent approach, and the application scenarios for performing real-time prototypical tests. Finally, we conclude in Section-4, and later the directions of our future work follows.

2 Related Works

Several efforts have been reported in the literature as far as the automation of network management functions is concerned. However, many comparable

Table 1. Characterization of network management information

Network Management Information	Network Management Data		Network Management Knowledge
	Static	Dynamic	
Description	**Configurational Specifications** An organized set of domain specific concepts/facts and relationships among them	**Statistical Data or Numerical Facts** Operational status of network devices, error reports, server-logs … Network administrator analyzes this raw data to locate the root cause of network failures	**Empirical Knowledge** For performing the management tasks **I: Task Knowledge for Fault-diagnosis** `[Implicit Design]` Set of modeling components: 1: Symptom Detection 2: Hypothesis Generation 3: Hypothesis Discrimination **II: An agent-based Library of Fault-state Causal Reasoning Models of Network Resources** `[Explicit Design]` Modeling the cause-effect relations among the occurring faults
Incorporation	**Domain Knowledge Ontology** Nomenclature of network resources, ip-addresses, port numbers, Routing information, Internet Domain Names, Application settings & versions **[Protégé-ver: 3.1]**	**Dynamically Generated Run-time Network Data** Collected through RMON (I & II)-MIB or SNMPv2-MIB **[Syslog Functions]**	**Actualized as Multi-agent System** Production-rule type knowledge of the DASH-agent describes the behavioral characteristics of faults **[IDEA-1.2]** **+** Java-code (serves as the base-processes of DASH-agents) maps the functionality of empirical knowledge and provides inference (reasoning) strategy **[Java-2-SDK-ver: 1.4.2]**

studies focus primarily on the expert systems, or refitting the agent paradigm to management solutions. Further, these works seriously overlook many important issues regarding an ample exposition of diverse kind of network knowledge resources, for the efficient multi-agent interactions. A lot of work has been done on the data structure of managed objects, but nothing has been done regarding the management knowledge of communication networks. Since the scope of our research is confined towards exploiting the knowledge resources within the network management domain, therefore, we will mention the ones that we assess as the most relevant to our network knowledge representation scheme.

In this regard, the foremost effort about the importance of network management knowledge is discussed in [7], and that the network-related knowledge can be formalized in a similar way as the SMI-based general definitions for MIBs in an ASN.1 format, for sharing in the distributed automated networking environment. The KACTUS project [8] investigates the feasibility of ontological knowledge reuse in the context of complex technical diagnostic systems. [9] constructs the network static and dynamic information-related active resources, as well as experiential knowledge-related active resources for the network management domain. In [10], Lemos et al led to a generic knowledge acquisition mechanism based on the concept of domain and MIB-variable based causal models, for the communication network management support systems. However this work lacks many important issues about the mapping and actualization of various knowledge models. We aspire to further extend this concept towards building a functional system to monitor and control a network in an efficient manner, by elaborating the underlying knowledge resources.

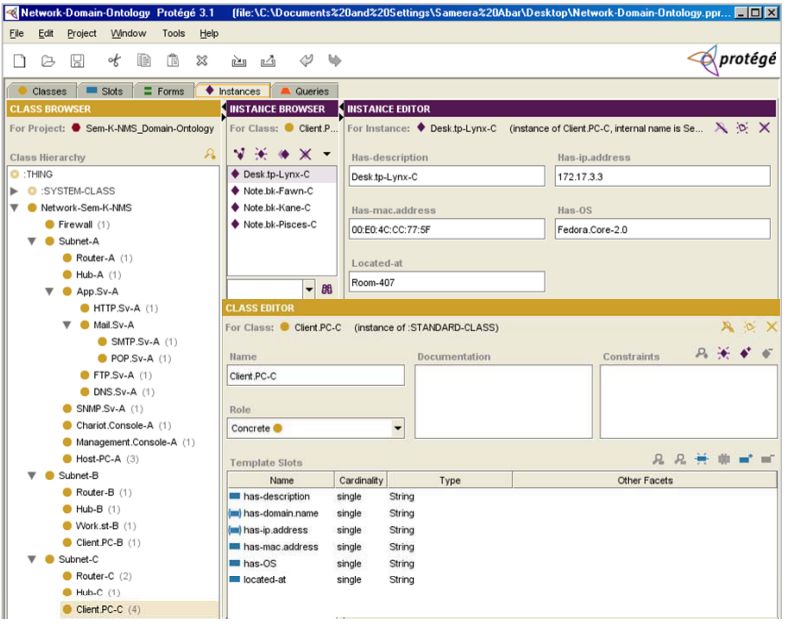


Fig. 2. Domain ontological model in Protégé-3.1.1

3 Design of Prototype System

3.1 Characterization of Network-Related Knowledge

Our modeling approach categorizes the network-related knowledge as the domain knowledge (static domain content, and dynamic status information), and the experiential management knowledge (inference strategy, and task structures), as illustrated in the Fig. 1. The proposed knowledge models have been constructed in-line with the CommonKADS [3], which is a comprehensive methodology for structuring the application intensive knowledge for the expert systems.

As shown in Table 1, the proposed modeling approach for knowledge acquisition results in a set of concise and logically consistent knowledge components and specifications, such as the domain ontologies, the fault-state causal reasoning models and the generic task-structures for handling the diagnostic sessions.

I: "Domain Factual Knowledge" specifies the static domain-specific content and structure in a declarative form. It consists of network systems' actual configurations and infrastructure, IP-routing details etc. In the proposed system, the network domain-specific concepts and their relationships are hierarchically organized as the domain knowledge ontology as shown in Fig. 2. These domain structures enable the empirical management content to navigate through them during the course of diagnostic reasoning process.

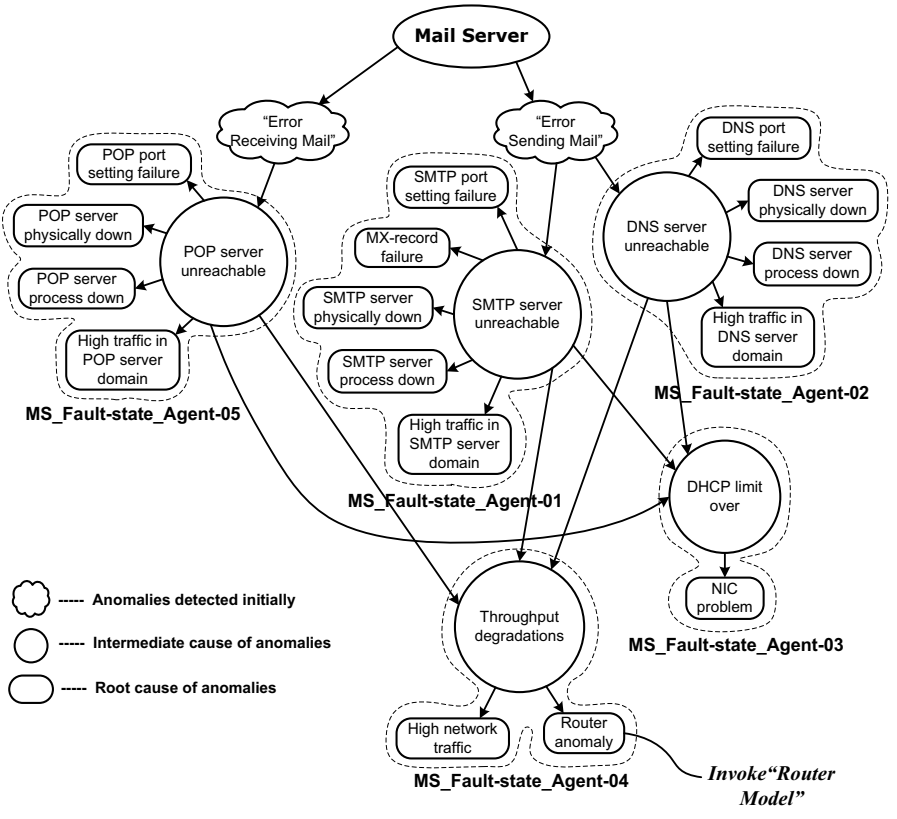


Fig. 3. Repository of fault-state causal reasoning models

The construction of our domain knowledge ontology is done using a knowledge acquisition and ontology editor tool—Protégé-ver: 3.1.1 [11]. While designing the network domain ontology for our experimental system, a particular care has been taken to ensure its reusability, maintainability, and modifiability in a flexible manner.

II: "Inference Knowledge" is modeled as the fault-state causal reasoning models which depict the behavior of various anomalies occurring in an operational network. The causal reasoning models represent the fault-states of network resources (physical devices and software applications), and are based on the observation and experience of the network experts. The reasoning models are constructed from a-priori expert knowledge of the probabilities that a specific set of cascading faults, causes the initial symptom. This basically means that we use a causal model of the anomalies to generate those hypotheses that cover or imply all the faulty observations. For instance, if the initial symptom reported is the error in sending e-mail, then one of the intermediate-cause of this problem could be the SMTP-server unreachable due to the throughput bottleneck, which

occurs as a consequence of high traffic or congestion in the router. Fig. 3 depicts an application scenario of a mail server anomalies, where nodes represent the events and whose directed edges represent causality.

These cause-effect relationships among the faults can be represented as the production rule-type representation of the DASH–agents' knowledge [12]. These small grain agents can be easily reused and modified. More generally, these rules provide inferences by invoking the underlying java networking routines (which serve as the base-processes of the intelligent agents) through interaction with the runtime dynamic status information (event-logs) of the operational network system. The inference knowledge rules support the basic reasoning steps required to achieve the management tasks in terms of operations on the domain content. When a rule is fired, its consequents are interpreted by the interaction of java-based programs with the networks' "syslog" information, and the knowledge of agents modifies dynamically according to the operational characteristics just as MIB's information is retrievable and modifiable in the conventional management solutions.

III: "Task Knowledge" is designed as a generic hierarchy of tasks which prescribe the activities to be performed in a domain of interest. The diagnosis is defined as the task of identifying the cause of a fault that is manifested by some observed behavior. For instance, the main network diagnosis task decomposes as the symptom detection, hypotheses generation, and hypotheses discrimination which in-turn break down in a sequence of sub-tasks. We refer to these task structures implicitly within the agents' cooperative problem solving behavior, during the network fault monitoring and detection phases. These three sub-tasks are the core of our network diagnostics system, and they are the ones for which the knowledge of the DASH-agents has been designed. The key functionality of the diagnosis tasks in the NMS domain is to identify the cause of the occurring fault symptoms.

3.2 Multi-agent Middleware System

The proposed system architecture is supported by the Agent-based Distributed Information Processing System (ADIPS) framework [13], which is a flexible computing environment for the implementation of the multi-agent systems. This framework employs a repository-based system development methodology which allows the autonomous adaptive actions on each designed distributed system. The intelligence layer of the proposed knowledge model consists of multi-agents for the automated handling of the empirical task-relevant knowledge to relieve the network operators from the tedious monitoring and control of an NMS.

The agents embedded in the middleware intelligent support layer, interact with each other cooperatively during the course of network-monitoring and fault-diagnosing sessions. As shown in Fig. 4, the SD-A, HG-A, and HD-A agents actualize the fault symptom detection, hypotheses generation, and hypotheses discrimination tasks, respectively. SD-A segregates the real-time incoming events into various failure cases in conjunction with the pre-defined fault-case taxonomy. The function of HG-A is to generate possible fault hypotheses in cooperation with the agents of fault-state reasoning models of the network objects. HD-A corresponds

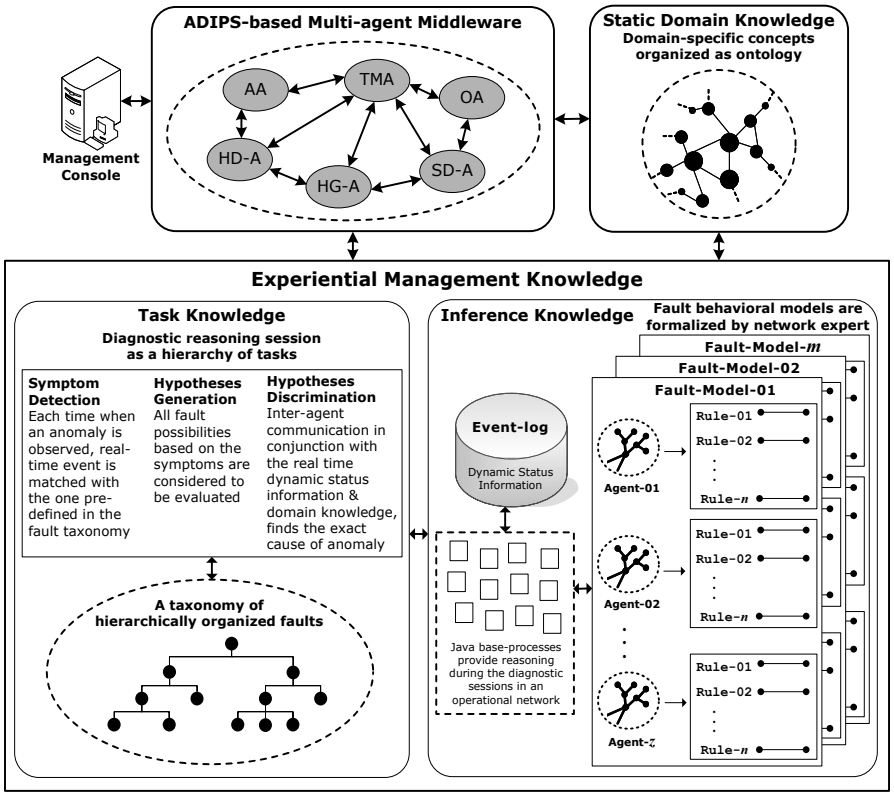


Fig. 4. Functional representation of networks' knowledge

to the sub-agents of each fault-state reasoning model. In fact, the hypotheses discrimination agent generates the explanation of these hypotheses or failures, and determines the root-cause of run-time faults, upon interaction with the java base-processes. OA, the ontology agent acts as a wrapper for the network domain knowledge resources. TMA administers and controls the organization of all the agents, and the functionality of the AA as an administrator agent includes the interactions with the management console. The multi-agent module of the proposed prototype is supported by IDEA-1.2 — an Interactive Design Environment for Agent designing framework-ADIPS/DASH–1.9.7h, and Java-2-SDK–ver: 1.4.2. This provides a repository-based multi-agent computing infrastructure that includes the agent model, development tools, rule-type agent description language, protocols, agent simulation (testing/debugging), and an execution environment.

3.3 Application Scenarios

Various application scenarios (related to the common network anomalies in our test environment) have been designed for deployment with the prototype system. A common observation is that the failures in the TCP/IP networking

environment occurring at the lower layers ("Data-link" or "Network" layer), cause the malfunctioning at the "Application" level significantly. Therefore, the lower level failures in the communication network must be monitored and detected effectively. In this context, the bit errors causing high packet loss-rate have been taken into account, which result in time-outs or broken connectivity in the router. Some other lower layer scenarios, for instance, the workstation congestion as well as high traffic encountered at the application server, have also been modeled. Furthermore, some application scenario representing the web-browsing unavailable, an error in sending/receiving email, DNS/SMTP/POP port-setting or configuration problems etc.) have been developed and are being tested with the prototype system. For the Physical/Data-link Layer Anomalies, NIC (Network Interface Card) failure has been considered. Due to the NIC failure, the broadcast storms or ethernet collisions cause increased number of re-transmissions, thereby reducing the throughput of the network to a minimum.

The prototype system monitors and detects the networks'operational anomalies automatically, generates hypotheses and then in conjunction with the operational networks' dynamic status information (incorporated with Java base-processes) locate the root cause of failure. Hence, the workload of the network administrator has been reduced remarkably. For evaluating the performance of our prototype system, some real-time tests to be compared with the conventional network management tools and mechanisms. The evaluation criteria is determined in terms of the accuracy, as well as the reduction in the time-taken and the effort-done in detecting the network failures.

4 Concluding Remarks

Focus point of our work is the elicitation of a network knowledge model in a generic and reusable manner, to be applicable during the course of automated performance analysis, and anomaly monitoring and detection in the computer (data) communication networks. More specifically, the designed network resource knowledge module comprises of network static content represented as the domain ontologies, and the management expertise as fault-state causal reasoning models of the network objects, which are actualized as the production rule-type knowledge of the software multi-agent based middleware system. The agents' rules along with the embedded generic java-based problem solving algorithms and real-time log information perform the automated management tasks.

This effort can be regarded as one of the initial step towards the construction and mapping of domain and reasoning knowledge models for the automated management support among the distributed communication network systems. For a proof of concept, the deployment of some test-bed application scenarios is under-way.

5 Future Issues

Our future work involves the further experimentation with few more knowledge templates depicting the test scenarios, to confirm the validity of our approach when compared to the conventional network management support tools.

Currently, we are looking into the SNMP-managed TCP/IP-based networking environment. For a more realistic view, later the key concept can quite feasibly be interpolated to OSI/CMIP-management, or other dominant network management standards such as IEEE 802.x, Web-based Management (WBEM, JMX) etc. Furthermore, the MAS-based semantic knowledge model described here can serve as a practical test-bed for the real world distributed applications in the various diagnostic reasoning domains, for instance, the electrical and telecommunication network domains.

References

1. Abu-Hanna, A., Jansweijer, W.: Modeling Domain Knowledge Using Explicit Conceptualization. IEEE Expert, Vol. 9, (1994) 53-64
2. Brewster, C., O'Hara, K.: Knowledge Representation with Ontologies: The Present and Future. IEEE Intelligent Systems, Vol. 19, (1985) 289-350
3. Schreiber, G., Wielinga, B., Hoog, R. de., Akkermans, H., Velda, W. V. de.: CommonKADS: A Comprehensive Methodology for KBS Development. IEEE Expert, Vol. 9, (1994) 28-37
4. Hamdi, M. S.: MASACAD: A Multi-agent based Approach to Information Customization. IEEE Intelligent Systems, Vol. 17, (2006) 60-67
5. Abar, S., Hideaki, H., Abe, T., Kinoshita, T.: Agent-based Knowledge Acquisition in Network Management Domain. Proceedings of the 19th IEEE AINA Conference, Taipei, Taiwan, (2005) 687-692
6. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W. R.: Enabling Technology for Knowledge Sharing. AI Magazine, Vol. 12, (1991) 36-56
7. Li, J., and Leon, B. J.: SNMK: Simple Network Management Knowledge. Proceedings of the 6th IEEE NOM Symposium, Vol. 2, (1998) 381-390
8. Bernaras, A., Laresgoiti, I., Bartolome, N., Corera, J.: Building and Using an Electrical Network Ontology for fault diagnosis. Engineering Intelligent Systems, Vol. 6, (1998) 3-11
9. Konno, S., Iwaya, Y., Abe, T., Kinoshita, T.: Design of Network Management Support System Based on Active Information Resource. Proceedings of the 18th IEEE AINA Conference, Fukuoka, Japan, (2004) 102-106
10. Lemos, M. A. de., Barros, L.N. de., Bernal, V., Wainer, J.: Building Reusable Knowledge Models for the Communication Network Domain. Proceedings of the 4rth AKAW, Sidney, Australia, (1999) 381-390
11. Protégé — A Knowledge Acquisition Tool. http://protege.stanford.edu/
12. DASH — **D**istributed **A**gent **S**ystem based on **H**ybrid Architecture. http://www.agent-town.com/dash/index.html
13. Kinoshita, T., Sugawara, K.: ADIPS Framework for Flexible Distributed Systems, In: Ishida, T. (ed.): Multiagent Platforms. Lecture Notes in Artificial Intelligence, Vol. 1599. Springer-Verlag, (1999) 18-32

An Early Decision Algorithm to Accelerate Web Content Filtering

Po-Ching Lin[1], Ming-Dao Liu[1], Ying-Dar Lin[1], and Yuan-Cheng Lai[2]

[1] Department of Computer Science,
National Chiao Tung University, 300 Hsinchu, Taiwan
{pclin, mdliu, ydlin}@cis.nctu.edu.tw
[2] Department of Information and Management,
National Taiwan University of Science and Technology, 106 Taipei, Taiwan
laiyc@cs.ntust.edu.tw

Abstract. Real-time content analysis can be a bottleneck in Web filtering. This work presents a simple, but effective early decision algorithm to accelerate the filtering process by examining only part of the Web content. The algorithm can make the filtering decision, either to block or to pass the Web content, as soon as it is confident with a high probability that the content should belong to a banned or an allowable category. The experiments show the algorithms can examine only around one-fourth of the Web content on average, while the accuracy remains fairly good: 89% in the banned content and 93% in the allowable content. This algorithm can complement other Web filtering approaches to filter the Web content with high efficiency.

1 Introduction

Massive volume of Internet content is widely accessible nowadays. One can easily view improper content at will without access control. For example, an employee may watch stock information during office hours. Web filtering products can enforce the access control. The up-to-date products have widely adopted content analysis besides the *URL-based* approach [1]. Content analysis works with the URL-based approach to relieve the efforts of maintaining the URL list and to reduce the number of false negatives. The analysis classifies the Web content to a certain category first, and makes the filtering decision, either to block or to pass the content.

Despite the ongoing research on image and video content classification, *text classification* is typically the most efficient approach to Web content analysis. Many text classification algorithms have been around with high accuracy. They are often assumed to run off-line, so their execution time is rarely discussed. However, the efficiency of these algorithms is critical because slow content analysis in Web filtering incurs long user response time. The issues of accelerating the analysis should deserve attention.

This work presents a simple, but effective *early decision* algorithm to accelerate the filtering from the observation that the filtering decision can be made

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 833–841, 2006.

before scanning the *entire* content, as soon as the content can be classified into a certain category. A fast decision is particularly important since most Web content is normally allowable and should pass the filter as soon as possible.

The rest of this paper is organized as follows. Section 2 provides the background of this work. The early decision algorithm is described in Section 3. Section 4 exhibits the accuracy and efficiency of this algorithm from the experimental results and discusses the deployment issues in a practical environment. Finally, Section 5 concludes this work.

2 Background

Yang et al. and Sebastiani [2], [3] gave a comprehensive survey of existing text classification algorithms. These algorithms are shown to achieve around 80% of accuracy or higher, measured by the average of recall and precision. Recall is defined to be the ratio of the number of correct positive predictions divided by the number of positive examples, while precision is the ratio of the number of correct positive predictions divided by the number of positive predictions. Among these algorithms, we choose Naïve Bayesian classification as the base of the early decision algorithm for its simplicity. Other classification algorithms can follow the introduced principle to accelerate the classification.

The Bayesian classification is divided into two stages: training and classification. The training stage learns the probabilistic parameters of the generative model from a set of training documents, $D = \{d_1, d_2, \ldots, d_{|D|}\}$. Each document consists of a sequence of words from a vocabulary set $V = \{w_1, w_2, \ldots, w_{|V|}\}$ and has been labeled with some category from a set of categories $C = \{c_1, c_2, \ldots, c_{|c|}\}$ before the training. Two types of parameters are included in the model: (1) $P(w_t|c_j)$: the estimated probability of word w_t given category c_j and (2) $P(c_j)$: the estimated probability of category c_j. These parameters are derived by [4]

$$P(w_t|c_j) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i|d_i \in c_j)}{|V| + \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} N(w_t, d_i|d_i \in c_j)}, \tag{1}$$

where $N(w_t, d_i)$ is the times word w_t appears in document d_i, and

$$P(c_j) = \frac{1 + \sum_{i=1}^{|D|} P(d_i \in c_j)}{|C| + |D|}. \tag{2}$$

In the classification stage, the posterior probability $P(c_j|d_i)$ that a test document d_i belongs to category c_j is derived. The category c_j that maximizes $P(c_j|d_i)$ is the one that d_i belongs to. $P(c_j|d_i)$ is derived by

$$P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{P(d_i)} = \frac{P(c_j)\prod_{k=1}^{d_i} P(w_{d_i,k}|c_j)}{P(d_i)}, \tag{3}$$

where $w_{d_i,k}$ is the k-th word in document d_i. Notice that the document d_i is viewed as an ordered sequence of $< w_{d_i,1}, w_{d_i,2}, \ldots, w_{d_i,|d_i|} >$, with the assumption that the probability of a word occurrence is independent of its position in

the document, given the document category c_j, so that $P(d_i|c_j)$ can be written as the product of individual probabilities $P(w_{d_i,k}|c_j)$.

3 The Early Decision Algorithm

The philosophy behind the early decision algorithm is to make the filtering decision from the front partial Web content. Fig. 1 presents the average keyword distribution of both banned and allowable Web pages in our investigation. The keyword position is normalized by the page length and presented in percentage. The keywords in almost all Web pages tend to be distributed uniformly throughout the content or appear more in the front part according to this investigation. The Web content in a banned category starts to exhibit much more keywords than that in an allowable category since the front part. In other words, keywords from the front partial content can reveal the category of the Web content and serve as the clues to filtering.

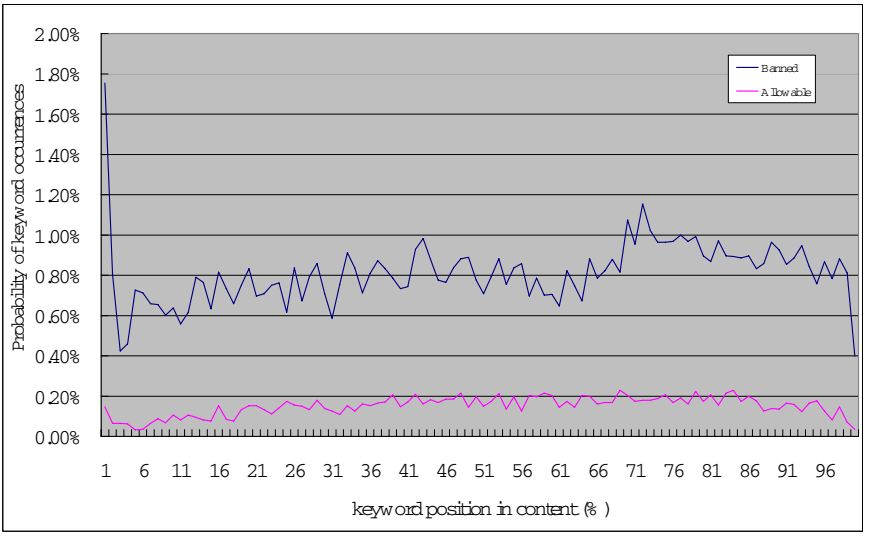


Fig. 1. The distribution of keyword positions in typical Web pages

Like the Bayesian classification, the filtering engine is trained off-line from the Web content in the banned categories. The *Bow* library and its front-end, Rainbow [5] perform the training herein, extracting keywords as the features from the target categories. The keywords with the information gains larger than a threshold are selected. Stop words, such as "the", "of" and so on, should be dropped because they help little in classification. The words inside the HTML tags are also ignored so that a malicious user cannot stuff unrelated content in the tags, particularly in the front part of the Web page, to deceive the filter. If

the malicious user fills the Web text outside the tags with irrelevant content to confuse the filter, the irrelevant content will be displayed in the browser and will spoil the layout of the Web pages — a great limitation on the design of the Web pages.

The score of keyword w_t that should belong to a category c_j is defined to be $logP(w_t|c_j)$, which can be derived in the training stage. Taking the logarithm simplifies the computation of the posterior probability $P(c_j|d_i)$ from multiplication operations to score accumulation with independence assumption between words [4]. The scores are accumulated while the content is scanned from the front to the end.

In the filtering stage, given $n\%$ of the content that has been scanned and the accumulated score of at least m, the probability that the content should belong to a category c is derived from

$$P(c|D_{n,m}) = \frac{P(D_{n,m}|c)P(c)}{P(D_{n,m}|c)P(c) + P(D_{n,m}|c')P(c')}. \tag{4}$$

1. $D_{n,m}$: the event that the filter has read n% of the content and has observed the accumulated score of at least m.
2. $P(c)$: the estimated probability that category c appears in typical Web content.
3. $P(c')$: the estimated probability that category c does not appear in typical Web content. $P(c) = 1$ - $P(c')$.
4. $P(D_{n,m}|c)$: the estimated probability that $D_{n,m}$ happens given that the content belongs to category c. The estimate of $P(D_{n,m}|c)$ is the number of Web pages in c that $D_{n,m}$ happens divided by the number of Web pages in c.
5. $P(D_{n,m}|c')$: defined similarly as $P(D_{n,m}|c)$, except that c is replaced with c'.

In the training phase, two two-dimensional indexed tables of $P(D_{n,m}|c_i)$ and $P(D_{n,m}|c_i')$ are built for each n and m from the training examples, for each $c_i \in C$. The values of $P(c_i)$ and $P(c_i')$ can be estimated beforehand or dynamically tuned in a running environment by recording and analyzing actual Web content. Next paragraph presents the early decision algorithm. Two thresholds, T_{bypass} and T_{block}, are defined to be 0.1 and 0.9 herein. PCD_i is the estimate that the content should belong to a category c_i. If PCD_i is less than T_{bypass} for all c_i in the list of banned categories, this means the content is unlikely to be banned and the remaining content should be bypassed. In contrast, if there exists some c_i in the list of banned categories such that PCD_i is larger then T_{block}, this means the content is likely to belong to c_i and should be blocked by the filter. A minimum of the content should be scanned in the process to avoid deciding too early from only the little front part of the content, which may make the filtering result incorrect.

The pseudo code of the early decision algorithm

```
Earlybypass = False;
Earlyblock = False;
```

```
n = 0;
Do {
  Read next keyword; // Skip stop words and the HTML tags.
  n = the percentage of content that has been scanned;
  m = the accumulated score;
  If (n > Min_Scan) {
    // scanning at least Min_Scan% of document,
    // Min_Scan=10 herein
    For (each category c_i in the set of banned categories) {
      PDC_i = P(D_{n,m}|c_i) of current scanning position;
      PDC'_i = P(D_{n,m}|c'_i) of current scanning position;
      PCD_i = (PDC_i*P(c_i))/(PDC_i*P(c_i)+PDC'_i*P(c'_i));
    } // end of For
    If (for every banned category c_i, PCD_i < T_bypass) {
      Earlybypass = True;
      Exit;
    }
    If (for some banned category c_i, PCD_i > T_block) {
      Earlyblock = True;
      Exit;
    }
  } // End of If (n > Min_Scan)
} while(not end of content);
```

4 Experiments

4.1 Performance Metrics

The F1 measure, initially introduced by Van Rijsbergen [6], takes the harmonic average of the recall and the precision as the measure of accuracy. To measure the acceleration, the average scan ratio (ASR) and the average throughput are defined by

$$Average\ scan\ ratio\ (ASR) = \frac{Total\ bytes\ scanned}{Total\ bytes\ in\ the\ content}, \tag{5}$$

$$Average\ throughput = \frac{Total\ bits\ in\ the\ ontent\ being\ filtered}{Total\ execution\ time\ of\ the\ filtering\ (sec)}. \tag{6}$$

4.2 Experimental Results

Totally 300 Web pages are randomly collected from the YAHOO directory services (*http://www.yahoo.com*) for the experiment in four typically banned categories: Pornography, Game, Online-Shopping and Finance. Another 300 pages are also randomly collected from other categories as the allowable content. The

extracted keywords in the training stage are searched through the Web content with a multiple string matching algorithm. Since short patterns are not uncommon in natural languages, a sub-linear time algorithm, such as the Wu-Manber algorithm [7], can hardly take any advantages. The filtering algorithm is implemented with Lex [8], which is based on the Aho-Corasick algorithm [9], so the performance is less sensitive to short patterns.

The accuracy of the original Bayesian classifier, which scans the entire content, is compared with that of the early decision algorithm for the four banned categories in Table 1. Only the shopping category suffers noticeable accuracy degradation whereas the other categories remain fairly good accuracy. A careful examination reveals it is because the keywords in the shopping category include many ambiguous words that also appear in allowable content. If this is the case, more other examples from the category can be trained until better keywords that lead to higher accuracy are derived. The filtering accuracy by averaging the accuracy of the four banned categories and that of allowable content are presented in Table 2. The filtering accuracy of both types of content with the early decision keeps fairly close to that when scanning the entire content, but only 17.22% of content in the banned categories and 26.51% in the allowable categories on average are scanned. This means a large portion of the Web content can be bypassed in Web filtering, and the execution time can be significantly shorter.

Table 1. Comparison of classification accuracy

Algorithm	Porn			Games			Shopping			Finance		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Original Bayesian classifier	1.00	.993	.996	1.00	.971	.985	1.00	.975	.987	.896	1.00	.945
Early decision	.977	.918	.947	.958	.819	.883	.866	.750	.804	.964	.090	.931

Table 2. Filtering accuracy and average scan ratio of the early decision algorithm

Algorithm	Banned			Allowable			ASR in	ASR in
	Pr	Re	F1	Pr	Re	F1	banned content	allowable content
Early decision	.941	.847	.892	.947	.920	.934	17.22%	26.51%

False positives of allowable traffic are usually unacceptable in a practical environment and a higher threshold T_{block} would be better. By lifting the threshold T_{block} to 1.0, false positives in the allowable categories can be almost avoided. Table 3 presents that a higher threshold also results in more false negatives in the banned categories because some banned content cannot reach such a high threshold. Choosing a proper threshold is a tradeoff in a practical environment.

The execution time and throughput of the original Bayesian classifier and the early decision algorithm are compared on a PC with Intel Pentium III 700 MHz and 64MB of RAM. Table 4 presents the average execution time and the

throughput of filtering the banned and allowable content. The results show significant improvement in throughput, about five times higher than that of the original Bayesian classifier for banned content and nearly four times higher for allowable content.

Table 3. Accuracy in the setting of no false positives in allowable content

Algorithm	Porn			Games			Shopping			Finance		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
Original Bayesian classifier	1.00	.993	.996	1.00	.971	.985	1.00	.975	.987	.896	1.00	.945
Early decision	.1.00	.733	.871	.1.00	.623	.767	.1.00	.550	.709	.1.00	.730	.843

Table 4. Comparison of the throughput of the early decision algorithm and the original Bayesian classifier

Algorithm	Execution time (μs)	Throughput (Mb/s)
Original Bayesian classifier	1,333,772	41.05
Early decision for banned content	241,887	226.36
Early decision for allowable content	239,895	156.68

Many commercial products and open source packages in our investigation, such as DansGuardian (*http://dansguardian.org*), can block a page as the score accumulation achieves the given threshold configured arbitrarily by the users. The early decision algorithm compares the threshold with the probability estimation of the classification, rather than the score itself. The advantages of the early decision algorithm over the method in DansGuardian are two points. (1) The two parameters, T_{bypass} and T_{block}, have stronger association with the accuracy than the threshold of score in DansGuardian. The filtering can then be better tuned directly according to the desired accuracy. The choice of a proper threshold of score in DansGuardian to get the desired accuracy needs to take more efforts by trial and error. (2) The early decision algorithm accelerates not only filtering blocked Web pages, but also filtering allowable pages. The acceleration is particularly significant when the Web accesses are mostly allowable content.

The early decision algorithm is also implemented by modifying the filtering code in DansGuardian. The throughput is enhanced by about three times on average than that in DansGuardian in our testing samples because of the acceleration from the allowable content and the better criterion to decide the blocking. This algorithm can also be implemented into other Web filtering products to accelerate the filtering process.

4.3 Practice Considerations in Deployment

With the increasing number of categories to be classified, ambiguity between these categories may increase. In our opinion, the proper place to perform Web content filtering is restricted to the edge devices for performance reason. Such edge devices usually require fewer banned categories. The problem with increasing number of categories is not that serious.

The early decision algorithm is supposed to complement other Web filtering approaches, such as URL filtering, not to replace them. Some situations, such as SSL connections and content of images, video, Flash objects or Java applets, are non-trivial to analyze on line. This algorithm can work with other approaches to filter the Web content with high efficiency.

The two thresholds, T_{bypass} and T_{block}, can be tuned according to the tradeoffs between accuracy and efficiency. The accuracy can be increased at the cost of less efficiency by decreasing T_{bypass} or increasing T_{block}, and the efficiency can be increased at the cost of less accuracy by increasing T_{bypass} or decreasing T_{block}. The tuning depends on which is more important for an organization: accuracy or efficiency.

5 Conclusions

This work addresses the problem of possibly long delay from text classification algorithms to perform run-time content analysis of Web content. An early decision algorithm to decide to either block or pass the content as soon as the decision can be made is presented. A significant performance improvement is observed. The throughput is increased by about five times higher for banned content and nearly four times higher for allowable content while the accuracy remains fairly good. In the F1 measure, the accuracy can achieve about 89% for filtering banned content, and about 93% for allowable content.

The early decision algorithm is simple but effective. The same rationale behind this algorithm can be applied to other content filtering applications as well, such as anti-spam. The algorithm can be also combined with more features other than keywords from the text to further increase the overall accuracy of the content filter. Besides, the filtering can be further accelerated by combining the URL-based method with the cached results. That is, by caching the URLs of the filtered Web pages, duplicate filtering on the same Web page can be avoided. Content analysis can be skipped if the cached URL is matched. The maintenance of the URL list is also facilitated.

Acknowledgement

This work was supported by National Science Council under the Grants NSC 94-2752-E-009-004-PAE.

References

1. Internet Filter Review 2005. Available at *http://internet-filter-review.toptenreviews.com/*
2. Y. Yang and X. Liu, A re-examination of text categorization methods, Proc. of SIGIR' 99, 22nd ACM International Conference on Research and Development in Information Re-trieval (1999) 42–49

3. F. Sebastiani, Machine learning in automated text categorization, ACM Computing Survey, vol. 34, No. 1 March (2002) 1-47
4. Tom Mitchell. Machine Learning, McGraw Hill (1996)
5. The Bow library and Rainbow. Available at *http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/*
6. C.J. van Rijsbergen. Information Retrieval, Butterworths, London (1979)
7. S. Wu and U. Manber, A fast algorithm for multi-pattern searching, Technical Report TR-94-17, University of Arizona (1994)
8. M.E. Lesk, Lex — A lexical analyzer generator, Comp. Sci. Tech. Rep. No. 39. Bell Laboratories (1975)
9. Aho, A. V., and M. J. Corasick, Efficient string matching: an aid to bibliographic search, Comm. of the ACM, 18 (1975) 333-340

Near-Duplicate Mail Detection Based on URL Information for Spam Filtering

Chun-Chao Yeh and Chia-Hui Lin

Department of Computer Science, National Taiwan Ocean University, Taiwan
{ccyeh, m9257007}@mail.ntou.edu.tw

Abstract. Due to fast changing of spam techniques to evade being detected, we argue that multiple spam detection strategies should be developed to effectively against spam. In literature, many proposed spam detection schemes used similar strategies based on supervised classification techniques such as naive Baysian, SVM, and K-NN. But only few works were on the strategy using detection of duplicate copies. In this paper, we propose a new duplicate-mail detection scheme based on similarity of mail context between incoming mails, especially the context of URL information. We discuss different design strategies to against possible spam tricks to avoid being detected. Also, We compared our approaches with four different approaches available in literature: Octet-based histogram method, I-Mach, Winnowing, and identical matching. With over thousands of real mails we collected as testing data, our experiment results show that the proposed strategy outperforms the others. Without considering compulsory miss, over 97% of near duplicate mails can be detected correctly.

1 Introduction

Today, Internet email service, one of the most important and successful Internet applications, is threatened by spam. According to a report in 2003 [1], more than one half of Internet emails are spam. The problem seems more serious today. Without a spam filter, one Internet user might receive over one hundred mails a day and find that most of them are spam. Spam generates unnecessary traffics to Internet backbone, and as a result, wastes network bandwidth and induces addition delay to normal packets. Moreover, receiving spam is a nuisance for Internet users. It costs Internet users more time and money to download their emails from their ISPs, and additional time to find out "real" mails from lots of spam mails they downloaded [2].

In literature, many spam filters use word terms in mail context as features for classification. From a set of well-classified mails (called training samples, including both spam and non-spam mails), two sets of words are selected to present both spam and non-spam mails. Each word in both sets is associated with different degree of correlation to the classes (spam or non-spam). A binary classifier(spam or non-spam) can be constructed from the training sample using

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 842–851, 2006.

different statistical classification models such as naive Bayesian [3,4], K-NN [5], SVM [6,7], or boosting tree [8]. While statistics-based spam filters are popular, they are subject to be attacked. More and more anti-filtering techniques are used by spammers to elude spam-filters, especially for those based on spam/non-spam words appear in the mail context [9,10,11,12].

Another simple and intuitive strategy is to detect duplicate copies of a spam message. An example system is such as DCC [13]. By its nature, spam will be sent to a large group of unknown victims. As a result, whenever duplicate copies of a mail are received by a group of Internet users, the mail is very likely to be a spam mail. Intuitively, a simple approach is to detect identical copies based on some message digest functions such as MD5 or SHA1. However, one possible countermeasure against the above duplicate-mail detection technique is using personalized spam mail, in which spammers customize their spam messages for different receivers. For example, a common approach is to add spam victim's email account or few random characters to the spam context. Such a personalized spam mail can be done quite easily by software.

A common criticism on the spam-detection technique based on duplicate mail detection is that it could raise false alarms when a normal mail is sent to multiple receivers. A possible approach to relieve the false alarm is using white list. Usually, senders of a normal multi-addressee mail one might receive are somewhat predictable, either from some specified persons (e.g. one's friends) or from some specified organizations (e.g. the organization one works with, or the mail-list one subscribed). These specified persons/organizations can be added to white list to effectively reduce the number of false alarms.

While many spam detection schemes have been developed, spam systems become more sophisticate to against spam filters. To successfully send their spam message to Internet users, spammers would change their spam behaviors to avoid being detected. Very likely, each spam detection technique has its weakness. It seems not possible to design a detection technique to against all possible spam tricks. At least, the argument is correct today as we have not seen such an omnipotent technique existed. Combining different strategies to form an effective detection system sounds more practical. Currently, most active researches on spam detection are those based on statistic classification schemes. Most of them provide similar spam-indicating information (for example based on context terms or structure), while at same time they present similar weakness. Contrastively, mail duplication provides another view point of spam characteristics. We believe combining different degrees of spam-indicating information could be more effective and more robust to against spammer's tricks. To our best knowledge, few research results were reported in literature on duplicate mail detection for spam filtering.

The rest of the paper is organized as follows. In the following section, we give a more detailed discussion on the problems we deal with. In Section 3, we provide our solution schemes. Section 4 presents the experiment results. Conclusions are made in Section 5.

2 Problem Description and Design Strategies

2.1 The Needs for Detection of Near-Duplicate Mails

It is clear a spam should be disseminated to Internet users as many as possible, from a spammer point of view. Consequently, mass copies of the spam will be sent to a group of pre-selected (or random-selected) Internet users. Clever spammers would know that a spam message would be easily detected if all the copies of the spam are identical copies. In general, there are two main reasons to customize the same spam message for different receivers. First, the spammer wants to personalize the message for different receivers in salutations, or to indicate the time the mail being sent. Second, the spammer intends to add some random contexts to avoid being detected through message fingerprints. Some examples about how spammers can play the trick to avoid sending identical copies are:(1)making difference in salutations, (2)adding some random text in the begining or the end of the message, (3)inserting some special symbols, and (4)utilizing the tricks with HTML. Some special ASCII codes can be used to insert in mail context while not to make too much trouble for receivers to understand the message. Examples are such as Space, CR, LF, Tab, to name a few. Also, For HTML mails, many tricks can be played to change the context stream while maintaining semantics of the message. These tricks are such as using fonts, layout, or breaking the words.

2.2 Why Use URL Information

According to our observation, in recent years, more and more spam mails include hypertext in the message body. Parts of them might include some hyperlinks with simple titles. Different potential benefits encourage spammers using hypertext in their spam. First, comparing with plain text, hypertext can provide more colorful and vivid message. Using URL hyperlinks embedded in mails to link multimeda contents (such as picture, audio or video) is easier than to send the whole content files with the mail, in terms of size. Meanwhile, it is easier to gather the information of how many times the spam message is opened, which is truly important for spammers to show to their customers how effective their spam systems can achieve. On the other hand, using hypertext can avoid the spam message being detected. Many spam detection schemes rely on checking mail context for some targeted spam-related terms. To hide these hot terms, spammers can play varieties of hypertext tricks such as using formatting text, graphic, images, hyperlinks, to name a few.

Meanwhile, there are two folds to detect spam with URL information. First, it is hard to fake. URL information corresponding to the Internet resources needed by the mail should be correctly stated to have the browser access the resources successfully. Two copies of same spam would very likely contain same URL information. Second, it provides useful information about which websites are closely related to spam. By collecting URL information from those spam mails, one can easy find which websites provide the resources pointed by the spam. These suspicious websites can be put into a black-list to be blocked.

3 Proposed Approach

3.1 Spam Detection Process

Figure 1 shows the block diagram of the proposed spam detection process. When a new mail (M_x) comes in, the system first checks whether it is in white list or black list. If yes, immediately add a black-list/white-list mark into this mail. If not, do the duplicate detection to check whether or not this mail is a duplicate mail.

The first step to do the detection in the proposed scheme is feature extraction.We use selected context information as features for comparison. For each incoming mail, we first try to extract all URL information included in the mail context. If the mail contains enough URL information with respect to the message size, we use the extracted URL information to form a set of token streams. Then, the token streams are partitions into a set of substreams. For fast comparison, each of the extracted substreams are map to a hash value using some message digesting functions such as MD5 or SHA1. (We used MD5 in our system prototype.) Detailed discussion on how to extract the token streams related to URL information is presented in next section.

On the other hand, if the mail did not contain enough URL information, we treat whole context as a long character stream. Then, apply a predefined *Stop List* to partition the character stream into multiple substreams. Among all the substrams we keep at most K (a system parameter, defined to be 20-30 in our experiments) of longest substreams. Again, we use hash value of each selected substream to represent the fingerprint (the feature set) of the mail.

After the features (that is a set of hash value) of the incoming mail is extracted, we compare the features with those feature sets collected before. Two feature databases are maintained. One is for mails containing enough URL information; the other is for those containing less (or none) URL information. For two mails (M_i, M_j) , the similarity between the two mails is measured by a similarity function $S(M_i, M_j) = 2 * number\_of\_matched()/(dim(M_i) + dim(M_j))$, where $number\_of\_matched()$ is the number of features matched in both of the mails M_i and M_j. And, the function $\dim(M_i)$ represents the number of features in M_i.

The proposed spam detection process can be deployed in mail servers and coworks with other existing spam detection strategies to form an effective multi-layered spam filtering subsystem. Also, it can work along as a server to providing duplicate-mail detection services for mail servers (or user clients).

3.2 Token Streams Processing for URL Information

For mails containing URL information, we use the URL information presented in the mail as features. Basically, a URL uniquely defines an Internet resource. It is hard to impersonate a URL X for an existing Internet resource with another URL Y which points to another available Internet resource. However, the flexibility of URL mechanisms enables different URLs (in terms of the character strings

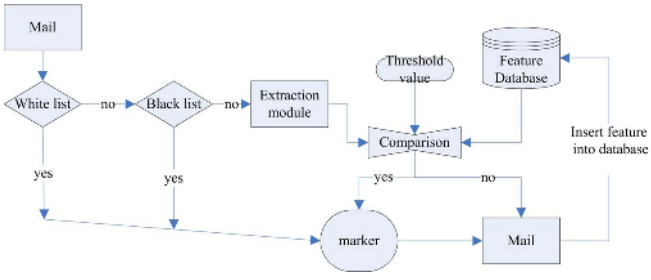


Fig. 1. Spam detection process

to present the URLs) point to same Internet resource. Spammers could play different tricks to make two URLs look different but actually point to same Internet resource. In general, two different approaches can be used to play URL tricks: URL resolving and dynamic webpage. To against spammers play tricks to invalidate the detection, we propose following different mechanisms to handle URL information.

- Full-length (FL): Preserve all the original URL information.
- Ignore-CGI (ICGI): Same as URL-FL but ignore all the CGI parameters.
- IP-only (IP-only): Use only the IP address corresponding to the domain name in the URL.
- Truncated domain-name (TDN): Same as URL-ICGI but with a truncated domain-name.

The FL mechanism preserves original URL strings, which provides a baseline for comparison with other complex mechanisms. The ICGI mechanism ignores the CGI parameters to against spammers playing trcks on CGI. The IP-only mechanism converts each URL to the host (IP address) pointed by the URL, by applying DNS query process. It ignores all possible fake CGI parameters and URL names, while takes a risk to cause false alarm. The TDN mechanism takes a compromise between the ICGI and the IP-only mechanisms. With TDN, we skip all subdomain information belonging to internal domain structure of an individual such as a company or an organization. It can be done by paring the full qualified domain name (FQDN) in a URL downward from top level domain name. If the domain name belongs to specified domains or well known domains such as those belonging to gTLD (generic Top Level Domain)or ccTLD (country-code Top Level Domain), check the domain name at next level; otherwise stop at this level and ignore all the domain names in the remaining levels. Figure 2 gives an example about the mechanisms.

As we discussed before, spammers might play different tricks to make two same URLs look different. These artificial noises imposed by spammers can be reduced to some levels with the proposed handling schemes discussed above. However, spammers still get some chances to avoid detection if the imposed noises cannot be removed completely, due to sensitivity of the hash function. For example,

assume two different URLs, "http://x1.y1.z1.aaa.bbb.ccc/ad1/p.php?e=123 q=z uid=12345" and "http://x2.aaa.bbb.ccc/ad2/p.php?e=123 q=z uid=xyztuv", refereeing to a same Internet resources actually. After Truncated domain-name processing, they become "aaa.bbb.ccc/ad1/p.php" and "aaa.bbb.ccc/ad2/p.php". The two are almost the same except one difference in directory part: "ad1" v.s. "ad2". Nonetheless, the two URL strings result in different hash values. One possible stategy to reduce the effects (a small part of errors dominates all the parts) is to divide a URL string into multiple substrings. Then, use substrings as bases for comparison. We refer this processing scheme as *subtoken* mechanism in the following context. Following above example, if we use "/" as a separator, we will get two feature sets, {"aaa.bbb.ccc", "ad1", "p.php"} and {"aaa.bbb.ccc", "ad2", "p.php"}, for the two URL strings.

URL link	http://x1.y1.z1.aaa.com.tw/ad1/p.php?e=123 q=z uid=12345
FL	"x1.y1.z1.aaa.com.tw/ad1/p.php?e=123q=zuid=12345"
ICGI	"x1.y1.z1.aaa.com.tw/ad1/p.php"
IP-only	"111.222.333.444"
TDN	"aaa.com.tw/ad1/p.php"
TDN-subtoken	{ "aaa.com.tw" , "ad1" , "p.php" }

Fig. 2. Examples for the URL processing mechanisms

4 Experiment Results

4.1 Test Data Set and Performance Measurement

In this research study, we did intensive experiments based on real data. Although there are some spam collections available in public, most of these spam collections are not for studies on duplicate mail detection. Usually, the spam mails are contributed by individual Internet users. If the number of the contributors is not large enough, it does not provide enough samples of (near)-duplicate mails, and thus hardly be useful in this research study. Possible solutions are to use multiple spam archives. How, duplicate reports/collections in these archives would bias the test results, if we simply aggregate all mails form multiple spam archives without proper pre-screening. Consequently, we collected the test data from a mail server, which provided mail services for over hundreds of users. The test data consist of over seven thousands of mails (7,750 in total), which we collected during November 2004 from the server. For privacy concerns, we collected only those mails classified as spam by a mail filter (SpamAssassin). Again for privacy concerns, we performed a random process to collect the spam randomly, instead of collecting all the spam. The collected data sets are summarized in Table 1, which show most of the collected (spam) mails contain URL information.

In our performance evaluation, we focus on the accuracy of the detection system. The accuracy is measured by error rate. For a mail set M and a detection scheme S, the error rate corresponding to the detection scheme S applying on the data set M, denoted as $R^S_{error}(M)$, is defined to be $R^S_{error}(M) = C^S_{error}(M)/C(M)$, where $C^S_{error}(M)$ is the number of mails which are classified incorrectly (either false positive or false negative), and $C(M)$ is the number of mails in the data set M.

Table 1. Test data sets

Data set	Total nails	URL_info	No_URL	% of URL_info
data_set1	2557	2534	23	99.1%
data_set2	2872	2856	16	99.4%
data_set3	2321	2289	32	98.6%

4.2 Results on Different Proposed Mechanisms

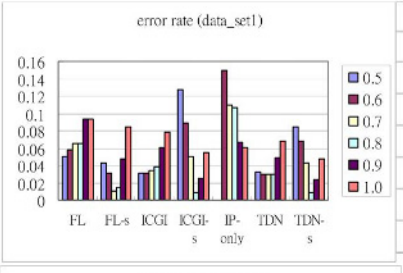
In this section we report the experiment results on the proposed scheme. Due to space limitation, we only discuss the results on those mails with URL information (containing over 98% of all test data). The results for the four test data sets are shown in Figures 3. We evaluated the error rate (y-axle) under different proposed mechanisms (x-axle) with different threshold value (0.1 - 1.0). The mechanism with "-s" appended to the name stands for the mechanism with the subtoken preprocessing during the feature extraction process. In general, small threshold value would cause more false negative, while large threshold value would cause more false positive. For all the evaluated mechanisms the best threshold value is between 0.5-1.0. Consequently, in the figure, we show only the cases for the threshold value from 0.5 to 1.0.

We found that the results on the three data sets are quite consistent. The results show the strategies with TDN and ICGI outperform others, with TDN slightly better than ICGI. With subtoken mechanism, the performance can be improved futher (again ICGI and TDN strategies get better results), while it makes selection of a proper threshold value more sensitive. The three data sets show that there is a good consensus between them for the selection of proper threshold value. Under the case of subtoken mechanism, the proper setting of the threshold value for both of ICGI and TDN is about 08-09. Under the setting, they can achieve an error rate as low as around 0.01 for these three data sets.

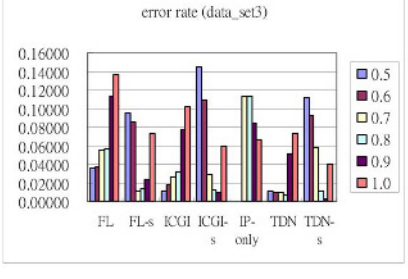
Besides, something are worthy of notice on e-papers, which are sent periodically to potential users. For two e-papers from same publisher but for different contents (for example published on different weeks or months, or for promoting different products), their URL information embeded in the mails are likely very similar (to share same framework for the presentation style and common information). Consequently, the IP-only mechanism would misclassify this two e-papers as same one. The ICGI and TDN mechanisms encounter same situation, but not so stringent as the IP-only mechanism. However, from different points

of view, if we treat all e-papers as spam, both of the ICGI and TDN mechanisms could successfully filter these e-papers (since they are treat as duplicate mails). Moreover, as we state before, to achieve a good performance, we suggest the detection schemes should work with white/black list. For those e-papers a user has an interest to receive, he/she can put the publisher information of the e-papers to white list to avoid being misclassified.

In the experiments, we classify those e-papers with different contents as different mails, which causes more errors on the mechanisms with the ICGI, TDN, and IP-only strategies. Nonetheless,the results show the mechanisms with ICGI and TDN still outperform the FL mechanism. From spam detection point of, we can treat all the unexpected e-papers are same. Under such a setting, we can expect that the ICGI and TDN mechanisms can behavior even better.



data_set 1	FL	FL-s	ICGI	ICGI-s	IP-only	TDN	TDN-s
0.5	0.05084	0.04302	0.03168	0.12671	0.20375	0.03285	0.08447
0.6	0.05710	0.03050	0.03050	0.08838	0.15018	0.02972	0.06883
0.7	0.06492	0.01017	0.03402	0.05084	0.10950	0.03011	0.04224
0.8	0.06492	0.01486	0.03911	0.00821	0.10677	0.02894	0.00939
0.9	0.09386	0.04771	0.06023	0.02464	0.06609	0.04849	0.02386
1.0	0.09347	0.08447	0.07822	0.05553	0.06023	0.06766	0.04693



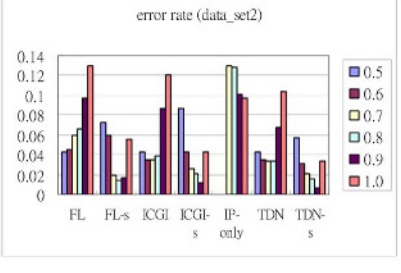


Fig. 3. Performance evaluation on different proposed mechanisms (detail data is shown only for data_set1)

4.3 Comparison with Other Approaches

To evaluate the strength of the proposed schemes, we compared performance between our approach with other four approaches: I-match, Winnowing, Octet histogram-based, and identical-matching-based. The I-match [14] and Winnowing [15] systems are two recently developed systems for near duplicate document detection. The Octet histogram-based approach [16] is a detection scheme designed especially for near duplicate mail detection. The identical-matching approach was evaluated as a baseline approach for comparison.

We carefully chose different system parameters for each of the schemes. Three different system parameters were chosen for performance evaluation. In general, we chose a best parameter setting for each of them (R2). From the best

setting, we choose two additional settings (R1 and R3) around it. Results of the performance comparison between the four mechanisms are shown in Figures 4. The results show that our proposed approach performs best. Without considering compulsory miss, over 97% of near duplicate mails can be detected correctly.

data_set 1	ICGI-s	TDN-s	FL-s	I-match	Winn.	Histo.	Ident.
R1	0.05084	0.04224	0.0305	0.13766	0.15252	0.04028	0.16504
R2	0.00821	0.00939	0.01017	0.12984	0.13414	0.03129	0.16504
R3	0.02464	0.02386	0.01486	0.14157	0.22057	0.03872	0.16504
data_set 2	ICGI-s	TDN-s	FL-s	I-match	Winn.	Histo.	Ident.
R1	0.02054	0.01602	0.01915	0.1163	0.10759	0.04248	0.14728
R2	0.01114	0.00592	0.01393	0.10829	0.06302	0.02577	0.14728
R3	0.04283	0.03343	0.01741	0.11769	0.07591	0.05292	0.14728
data_set 3	ICGI-s	TDN-s	FL-s	I-match	Winn.	Histo.	Ident.
R1	0.01206	0.0112	0.08531	0.11762	0.15381	0.03274	0.16372
R2	0.00905	0.00388	0.01077	0.11202	0.12452	0.01896	0.16372
R3	0.05903	0.0405	0.01422	0.14347	0.13873	0.02456	0.16372

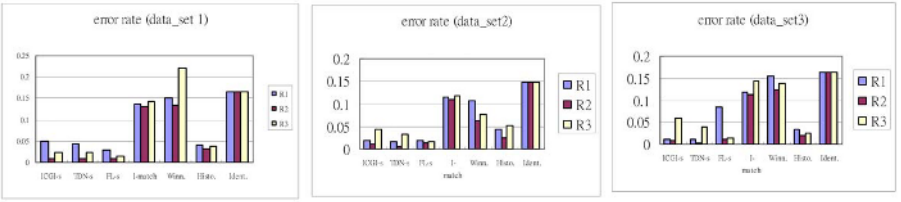


Fig. 4. Performance comparison between different approaches

5 Conclusion

Currently, spam is one of the most critical problems for Internet. The war between spam and anti-spam is still going. Multiple countermeasure, including spam detection, are investigated by researchers to stop spam. Each spam detection technique has its pros and cons. It seems not possible to utilize single detection strategy to against all possible spam tricks. Combining different strategies, especially for those complementary to each other, to form an effective detection system sounds more practical. In this paper we present our recent results on the spam detection techniques. We focus on the detecting technique based on duplicate-mail detection. From our collected data, we found that many of spam utilize URL hyperlink in the spam message, and we explained possible reasons and its impacts on spam detection. We expect the trend will last. Consequently, we proposed different mechanisms to deal with such a spam message. Based on thousands of real mails as testing data, we evaluated the proposed design strategies and made a comparison with other four different approaches available in literature. Results show that our proposed approach outperforms the others. Hopefully, this research study provides timely results on the spam detecting techniques based on duplicate mail detection, especially for spam message contenting URL information as presented in current and near future.

References

1. Weinstein, L.: Inside risks: Spam wars. Communication of ACM, Vol. 46, No. 8 (2003) 136–136.
2. Corbato, F.J.: On computer system challenges. Journal of ACM, vol. 50, No. 1 (2003) 30–31.
3. Sahami, M., Dumaisy, S., Heckermany, D., Horvitzy, E.: A Bayesian approach to filtering junk E-Mail. In Proc. Of AAAI Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin, (1998) 55–62.
4. Graham, P.: A plan for spam. Aug 2002, available at http://www.paulgraham.com/spam.html.
5. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a naive bayesian and a memorybased approach. In Proc. of the PKDD workshop on Machine Learning and Textual Information Access, (2000) 1–13.
6. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, vol. 2, (2001) 45–66.
7. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Trans. on Neural Networks, Vol. 10, No. 5,(1999) 1048–1054.
8. Carreras, X., Marquez, L.,: Boosting trees for anti-Spam email filtering. In Proc. of Euro Conference on Recent Advances in Natural Language Processing (RANLP 2001), Sep. 2001.
9. Hulten, G., Penta A., Seshadrinathan G., Mishra, M.:Trends in spam products and methods. In Proc. of First Conference on Email and Anti-Spam (CEAS), 2004.
10. Machlis, S.: Uh-oh: spam's getting more sophisticated. Computerworld, Jan 17 2003, available at http://www.computerworld.com.
11. Graham-Cumming,J.: How to beat an adaptive spam filter. In Proc. of MIT Spam Conference, 2004.
12. Wittel, G.L., Wu, S.F.: On attacking statistical spam filters. In Proc. of First Conference on Email and Anti-Spam (CEAS), 2004.
13. Distributed Checksum Clearinghouse (DCC). Available at: http:// www.rhyolite.com/anti-spam/dcc/.
14. Chowdhury, A., Frieder, Grossman, O.D., McCabe,M.C.: Collection statistics for fast duplicate document detection. ACM Trans. on Information Systems, Vol. 20, No. 2, (2002) 171–191.
15. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: local algorithms for document fingerprinting. In Proc. of SIGMOD 2003, (2003) 76–85.
16. Yeh, C.-C., Yeh, N.-W.,: Octet histogram-based near duplicate mail detection for spam filtering. In Proc. of IEEE-EEE05-MEM, 2005, Hong Kong, (2005) 14–20.

Two-Level Proxy: The Media Streaming Cache Architecture for GPRS Mobile Network*

Bo Yang, Jianxin Liao, and Xiaomin Zhu

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
yangbo@ebupt.com

Abstract. Based on the technical background of media streaming over GPRS (General Packet Radio Service) mobile network, a novel media streaming cache architecture of two-level proxy is proposed for streaming service in GPRS network. CCP (Central Cache Proxy) and UECP (User-End Cache Proxy) are applied to improve the performance of streaming service and to decrease the load of GPRS core network. Key technical details correlated to this architecture are researched, such as: reliability design of proxy server, caching policy, and strategy of charging and billing. Analyses and simulation experiments prove that this two-level caching proxy architecture can effectively optimize the performance of streaming service in GRPS network without upgrading the original GPRS network entities.

1 Introduction

GPRS (General Packet Radio Service)[1], brought forward in the specification of GSM (Global System for Mobile communications) Phase 2.1, is a high-speed data communication technology, it provides an end-to-end, high speed (data rate from 9.05 to 171.2 Kbps per user), large-area wireless IP connection to mobile users. More than 100 telecommunication carriers in the world have deployed their commercial GPRS system.

Media streaming[2] is the technology of combining multimedia with computer network. What is different from traditional Internet multimedia is that media streaming employs streaming method to transmit digital multimedia content. Instead of downloading the entire content locally before playing, media streaming allows the user playing while downloading, decreases the memory requirement

* This work is jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) Program for New Century Excellent Talents in University (No. NCET-04-0111); (3) Specialized Research Fund for the Doctoral Program of Higher Education (No. 20030013006); (4) Development Fund Key Project for Electronic and Information Industry (Core Service Platform for Next Generation Network); (5) Development Fund Project for Electronic and Information Industry (Value-added Service Platform and Application System for Mobile Communications); (6) National Specific Project for Hi-tech Industrialization and Information Equipments (Mobile Intelligent Network Supporting Value-added Data Services).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 852–861, 2006.

of terminal device. These characteristics make media streaming technology fit to GPRS mobile network and handset[3]. However, there are still some performance problems to be solved before streaming service is commercially used. These problems result from the current GPRS architecture: (1) the protocol stack of GPRS core network is not designed for continuous real-time streaming traffic, the GTP (GPRS Tunneling Protocol) introduces an additional time delay in data packet encapsulation and transmission; (2) the streaming is bandwidth-sensitive, with the increase of concurrent user number, the quality of service drops and the impact on GPRS core network raises; (3) core network does not support IP multicasting, which makes effective streaming scheduling policies (such as batching, patching) unavailable.

In order to solve the above problems, based on the GPRS mobile network architecture and the character of VoD (Video on-demand) service, a novel media streaming cache architecture of two-level proxy for GPRS mobile streaming is proposed. The principle of two-level caching proxy is to add a central cache proxy (CCP) and a user-end cache proxy (UECP) as media content caching server, connecting CCP and UECPs by high speed IP direct links. By using CCP, we store and access most of media programs locally; by applying prefix caching on UECP, we decrease the playback delay of media streaming service; by employing the IP connection between CCP and UECP, we decrease the media data transmission delay and free the core GPRS network from heavy multimedia data traffic load.

This paper is structured as follows. Section 2 introduces the background of GPRS and streaming technology, presents the two-level proxy media streaming cache architecture. Section 3 discusses the key technical details of this architecture, such as: reliability and robustness design, strategy of charging, mobile handset adaptability, live video service support, the deployment costs and so on. Simulation experiments are carried out to evaluate the performance of our two-level proxy architecture in section 4. Section 5 summarizes the characters of the two-level proxy media streaming cache architecture of GPRS mobile network.

2 Two-Level Proxy Media Streaming Cache Architecture

2.1 Basis

GPRS[1] is achieved by overlaying a new packet switching network on the GSM base framework. The new added parts include: SGSN (Serving GPRS Support Node), GGSN (Gateway GPRS Support Node) and G-Series interfaces. SGSN provides MS (Mobile Station) with such services as mobility management, routing selection and so on. GGSN is the access gateway to services and PDN (Public Data Networks).

The GPRS network protocol includes GTP (GPRS Tunneling Protocol), IP (Internet Protocol), LLC (Logical Link Control) and RLC (Radio Link Control). MS (Mobile Station) accesses BSS (Base Station System) by U_m interface; BSS interacts with SGSN by using frame relay technology on G_b Interface; SGSN

connects GGSN on G_n Interface, data and signaling messages between them are encapsulated in GTP (GPRS Tunneling Protocol) Tunnel.

In GPRS, the core network bandwidth is a rare resource, the streaming service is of bandwidth consumptive; by employing stream caching, we can save the GPRS network bandwidth by avoiding access to the streaming content server (SCS) for each individual customer request, hence reduces the data flow between GGSN and PDN, and shortens user's VCR-like operation (Such as: playback, pause, fast forward) delay.

2.2 Architecture of Two-Level Proxy Media Steaming Cache

Based on the GPRS network architecture, a novel media streaming cache proxy architecture is proposed as follows.

A CCP is placed between GGSN and PDN. The main function of CCP is: forwarding the RTP (Real-time Transport Protocol) video data packet from the SCS (Streaming Content Server), caching those stream contents that satisfy certain caching strategy in the form of media file; maintaining a content cache information table (CCIT) of cached files and their URL (Uniform Resource Locator) on the PDN; checking the RTSP (Real Time Streaming Protocol) messages between GPRS network and SCS. When a RTSP request from MS reaches the CCP, CCP gets the required URL and matches it in the CCIT, if the URL is matched successfully, CCP then accesses the local cached copy of the media content, packets and sends it to the terminal user; if the match fails, CCP simply forwards the RTSP request to SCS.

By adding a CCP, we decrease the repeated access of same program from the SCS, improve the service capability of streaming content server, and avoid the possibility of bottleneck between GGSN and SCS. If cache matches, CCP will provide streaming service with local cached copy, which is network cost saving and of low latency.

CCP optimizes the streaming service to a certain degree, but there are still three problems to be solved: (1)the path from GGSN to MS may still be in heavy burden in transmitting media streaming data packets; (2)along this path, packets may encounter many times of format encapsulation and decapsulation according to different protocols requirement, this consumes a lot of process time and network bandwidth; (3)the GPRS core network does not support multicast, which makes classical scheduling policies (such as: patching, batching, and merging) inapplicable. As the service users keep on increasing, these problems may lead to new bottleneck. So we introduce a new entity named UECP in each BSS, UECP is placed between BTS (Base Transceiver Station) and BSC (Base Station Controller). The main function of UECP is: adopting adaptive cache replacement strategy and prefetching the prefix of some popular media programs from CCP; retrieving the remained part from CCP on user's demand; interacting with CCP to maintain the media content cache information table; monitoring the RTSP message between MS and streaming content server, providing streaming service for the cached content and forwarding RTSP message for cache missed request.

The IP direct link is used to transmit data and control messages between CCP and UECP, which alleviates the burden of transmitting media streaming data on the GPRS core network from GGSN to BSC. Without GTP encapsulation and decapsulation, the data traffic efficiency is improved.

Fig.1 shows our design of two-level proxy media streaming cache architecture. A UECP is arranged for each BSS, a CCP is deployed between each GGSN and PDN, all the UECPs and CCPs connect to each other by IP direct link. Two protocol stacks run on UECP: by GPRS stack, UECP connects to BTS and BSC via A_{bis} interface, forwards GPRS signaling and data; by IP/Ethernet stack, UECP directly connects to CCP via G_i interface, providing direct channel for media data rapid transmission. In the application layer, UECP supports WAP (Wireless Application Protocol), RTSP and RTP/RTCP (RTP Control Protocol). CCP runs IP stack, connects GGSN and PDN together via G_i interface, connects UECP via TCP/IP protocol.

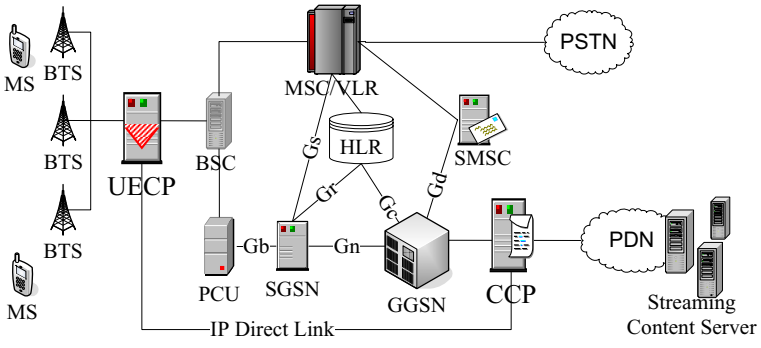


Fig. 1. Two-level Caching Media Streaming Proxy Architecture in GPRS Network, with two cache proxies added: UECP and CCP. UECP and CCP are directly connected by an IP direct link.

2.3 Mechanism of Two-Level Proxy Media Steaming Cache

According to the process of VoD service initiated by MS, mechanism of the two-level proxy media streaming cache architecture is described as follows.

Using a WAP (Wireless Application Protocol) enabled mobile handset, the customer browses the program provider's WAP page and searches interesting programs. After the selected program's hyperlink is clicked, the terminal initiates a RTSP request to the corresponding streaming content server. This RTSP message includes a URL, which is used to locate the content server and the required program. By monitoring application layer messages, UECP obtains the RTSP message from MS, gets the URL and looks up the content cache information table. If the URL is matched successfully, UECP terminates the RTSP message, maintains the state machine and sends corresponding RTSP response message back to the MS, accesses the cached file and packets it in RTP data

flow and sends it to the mobile user. At the same time, UECP schedules a task to download the suffix part of the program from CCP through IP direct link, the download task must be carried out in time to guarantee continuous video playback at the GPRS handset. If the URL is not matched, UECP forwards the RTSP message to CCP, the MS then obtains the media content from CCP or SCS.

After CCP receives the RTSP message forwarded by UECP, CCP picks up the URL and tries to match its CCIT. If cache hits, CCP terminates the RTSP message, maintains the state machine and sends corresponding RTSP response message to the MS, acquires the cached content, packs it into RTP packets and sends it to the end user through the link of {CCP – IP Direct Link – UECP – BTS}. If cache misses, CCP forwards the RTSP request message to SCS, SCS will send the RTP stream to MS through the link of {SCS – CCP – IP Direct Link – UECP – BTS}. In the course of CCP providing media content, UECP acts as a terminal, collects the content and caches the prefix to local disk according to the caching strategy, dynamically adjusts its cache information table.

By using CCP, the service load of streaming content server is alleviated; the consumption of PDN ingress traffic is decreased. With properly selected cache allocation and replacement algorithm, UECP can serve most of mobile terminals' request with cached media prefix. This rapid and localized respond improves the quality of VoD service. Bypassing streaming RTP flows between GGSN and BSC, the high-speed IP direct link cuts down the bandwidth requirement and process consumption, avoids the impact of streaming service on the GPRS core network as well.

What make mobile streaming service differ from Internet streaming service is that we must take the mobile users' mobility character into consideration. Some method should be taken to guarantee continuous service under terminal moving condition. In our two-level proxy architecture, each UECP serves one BSS. When user moves within one BSC area, the user's streaming service is provided by the same UECP, so that service discontinuity will not occur; when a user moves between different BSC areas, the inter-BSC handover messages between MS and BSC can be trapped and applied to trigger the correlated UECP handover and cache duplication so that streaming service can be provided smoothly. Due to limited space, the handover mechanism will be discussed in other works.

Advantages of the two-level proxy architecture can be summed up as follows: (1) with no requirement of modification to the standard GPRS network entity and protocol, the two-level proxy architecture is transparent to the former GPRS network, can work seamlessly in the original network; (2) bypassing the streaming traffic from the GPRS core network, the two-level proxy architecture relieves the impact on other GPRS services; (3) it improves the quality of VoD service; (4) IP multicast and application layer mutlicast can be supported, which makes multicast-based stream scheduling policies available; (5) it can support service continuity under moving environment.

3 Key Technologies

3.1 Design of Reliability and Performance

By introducing UECP and CCP, two more entities are added to the GPRS network, which implies two additional potential fault points; so the reliability of the two proxies is of great importance. We employ hot backup and load sharing method to improve the system reliability. Each cache proxy consists of two hosts at least, with the same hardware and software configuration. All these hosts work independently and share the same disk array to store and retrieve the cache files. In case one of these hosts crashed, the other host would take over its work.

In software designing, we take the robustness and the reliability of communication module into consideration; make sure that the control and data messages can be forwarded correctly even when cache function failure occurs on the cache proxy server. Thus, even in the worst condition the quality of streaming service will only degrade to the original degree of no-cache, with no effect to other GPRS services.

The process of filtering and forwarding messages may bring some additional delay. By using a programmable network processor, we can improve the performance of proxy server, limit the additional delay to an acceptable degree.

3.2 Caching Policies in Two-Level Proxy Architecture

The cache replacement policy we used in CCP server is LFU (Least Frequently Used): for each video object, CCP maintains the access number of the object during a certain time period; when the CCP cache space is full, the objects with minimal access number is evicted to make room for the new one. The cache establishment mechanism uses the same policy: at startup time, the cache space is empty and the access information table is set to null, objects are cached and evicted according the LFU policy.

For the following reasons, caching the whole program in CCP is feasible and effective: (1) only a limited number of CCPs are needed for a GPRS network, equipping CCP with powerful storage device costs a trivial additional investment; (2) the video content for GPRS environment is specially coded (such as H.264), the full video file size is relatively small; (3) the more content cached, the more network bandwidth between GGSN and PDN is saved. But for UECP, things are different: (1) UECP is required by each BSS, a large amount of UECP is needed in the two-level proxy architecture; (2) in each BSS domain, the number of service users is relatively small, full content caching is a waste of resource and processing time in low rate access environment. Therefore, we apply prefix-caching[5] method in UECP. Together with the prefetching method, the prefix of each programs cached in CCP are downloaded to the UECP in advance, the playback startup delay is decreased by local accessing of media program prefix from UECP.

3.3 Charging Function in Two-Level Proxy Architecture

GPRS is not a circuit switched network, its charging scheme always bases on data traffic size statistics. GGSN and SGSN are the standard charging function entities in the GPRS network, they produce the charge data record (CDR). In the two-level proxy architecture, not all streaming request and data messages reach GGSN and/or SGSN. For example, in the case of UECP cache hits, UECP provides streaming data service instead of SCS, GGSN is not aware of the data traffic. So we need to extend the charging function entities to include proxy servers. For the cached object, UECP and CCP supervise the service and traffic, generate the charging information CDR, and upload CDR to the charging system periodically.

Implementing charge function on cache proxy makes the GPRS charging system be more complicated and increases the cost of proxy server. But the additional investment is not a waste, by introducing UECP and CCP, we decrease GSNs' charge burden, and provide more flexible charge strategy, such as: based on where the content the user required located, UECP, CCP or SCS; based on popularity rate of the video program; based on quality of service and so on.

3.4 Adaptation of Mobile Terminal

Mobile video streaming service is terminal sensitive, different handsets may result in different level of media service quality. In streaming service, we mainly concern about terminal characters such as size of screen, number of available colors, resolution, video format and so on. To fit the diverse kinds of handset, we have three candidate methods: (1) For each video program, the cache proxy stores several copies for different type of handsets, sends the appropriate copy to the fitting terminal; (2) Encoding the media content using an adaptive encoding format and requiring the terminal to decode the content according to its own capability; (3) The cache proxy stores a high quality file and transcodes according to the terminal's capability before sending content to the end user. We prefer the third method because it can adaptively support many kinds of terminals, and with no additional terminal process ability and additional cache proxy storage requirement. In our system, UECP is used to adapt user handset and transcode streaming content. Transcoding requires a great deal of processor resources. Fortunately, there are many kinds of DSP (Digital Signal Processing) boards available for high-speed video format transcoding.

UAProf (User Agent Profile)[6] is used to support the handset's capability negotiation between cache proxy and handset. UAProf is one of WAP Forum's standard of describing the mobile terminal's capability. WAP Gateway can obtain the UAProf by interacting with the terminal. The UECP gets the user handset's capability from the WAP gateway and sends the appropriate transcoded media streaming to corresponding terminal user.

3.5 Live Video Service Support Using Streaming Agent Function

Generally, there are two kinds of video streaming services: video on-demand and live video. From above discussion, we can see the two-level proxy architecture optimizes performance of the video on-demand services by caching hot programs in local cache server. To optimize the quality of real-time live video services, we can further extend UECP's ability to support streaming agent (SA)[7] function. According to the character of transmission medium, GPRS network are separated into two parts: wired network and wireless link. Placing on the junction of the wired network and wireless link, the SA periodically generates partial path RTCP message and sends it to the server and the end user. Having obtained the RTCP messages sent by the client and the SA, video streaming server and proxy can distinguish where the traffic is congested, the wired network or the radio link, and take the feasible method to optimize the network traffic.

Two characteristics make it suitable for UECP to act as SA: (1) UECP lies between wired and radio network, the very place for SA; (2) working as a content proxy, UECP has been designed to forward RTP messages, can be easily extended to support RTP packet statistics and timely RTCP messages feedback without extra cost. To act as SA, the UECP functional extension includes: generate RTP packet statistic information, send partial path RTCP message to user and server periodically, analysis the RTCP message from mobile users and apply traffic control policies.

By extending UECP to support SA function, our two-level proxy architecture effectively improves the quality of service for both live video and on-demand video as well.

3.6 Deployment Cost of Two-Level Proxy Media Streaming Cache Architecture

Deploying two-level proxy architecture needs no modification to original network entity and interface, the deployment investment is divided into two parts: proxy server cost and IP network cost.

One CCP is equipped for each GGSN, a UECP is needed by each BSC area. Take a middle-sized city into consideration (such as Hang Zhou City, China), there are 72 BSC areas and 2 GGSNs. With such a small number, the proxy servers' investment is inexpensive.

For IP network, the bandwidth a mobile user needed to run streaming service is about 20Kbps[9], suppose there are 5000 concurrent users in each BSC area and 10% is using streaming service, so the network bandwidth is no more than 10Mbps. Taking security into consideration, the public MAN (Metropolitan Area Network) can be used to satisfy our system requirement, the cost nowadays is trivial.

4 Simulation

In section 2, the advantage of two-level caching proxy architecture is analyzed in detail. In this section, we verify the validity of two-level caching architecture by using simulation experiments.

The simulation environment is the same as Fig.1. Under the precondition of no crucial effect to the simulation result, we simplified the simulation network model with the following assumption: (1) all the users handsets are of the same type, with the same resolution and buffer size; (2) there are 3 BTS controlled by the same BSC, there are at most 50 concurrent users in each BTS; (3) the access rate of VoD service follows the Poisson Distribution of 5 user requests per minute; (4) there are 200 programs stored in the SCS, the popular programs have been cached both in UECP and CCP according to the object cache rate argument parameter; (5) the user request pattern follows the Zipf distribution[4].

The simulation was conducted under 3 different scenarios: scenario 1 (NO–CACHE) means the original GPRS network with no cache; scenario 2 (CCP–ONLY) indicates that only cache proxy CCP is used; while in scenario 3 (UECP +CCP) the two-level proxy architecture is deployed. In each scenario, the simulation ran 5 simulation hours. The relationship between the average initial playback delay and video object cache hit rate is plotted in Fig.2. The X-axis stands for percentage of video objects cached in proxy servers, while the Y-axis means the average initial playback delay. The simulation result shows that, compared with scenario 1, scenario 2 partly decreases the playback delay by reducing some access to SCS. For scenario 3, some of VoD requests is served by UECP with local cached content, dramatically decreases the startup delay by canceling the transmit delay in the path between BSC and CCP.

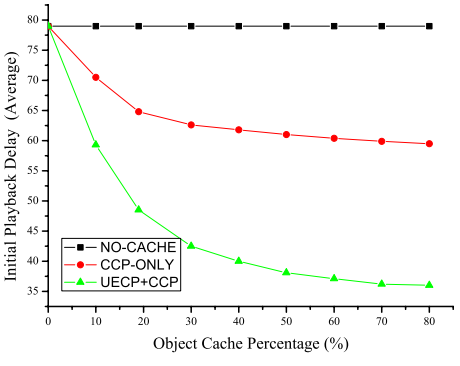


Fig. 2. Simulation Result: average initial playback delay vs object cache percentage under different scenarios (NO-Cache/CCP-Only/CCP+UECP)

From Fig.2, we find that the playback delay dose not decrease in proportion with the increase of object cache percentage, after the video object cache rate exceed 30%, the decrease rate of playback delay dropped sharply. This can be explained by the Zipf distribution property of VoD service: with about 30% of hot video objects are cached locally, most user's VoD access requirement are satisfied by local cache proxy, therefore, the startup delay close to constant value, the increment of object cache percentage has little contribution to the decrease of average playback delay.

5 Summary

The two-level caching proxy architecture can be summarized into three main characteristics: First, by using cache proxy and IP direct link, the performance of on-demand services is considerably optimized; second, by providing streaming agent function on UECP, the traffic control function for live video streaming service is effectively improved; last and the most important, the two-level proxy media streaming cache system can work transparently with the original GPRS network, no entity is need to be upgraded.

In the future, we plan to study the location-based streaming, streaming schedule algorithm, more effective cache lookup algorithm and cache replacement strategy for two-level proxy architecture.

References

1. Binjie Han: Theory and Optimization of GPRS Network. China Machinery Press (2004) 26-53
2. Yuzhuo Zhong, Zhe Xiang, Hong Shen: Media Streaming and Video Server. Tsinghua University Press (2003) 122-161
3. Christian Hoymann, Peter Stuckmann: On the Feasibility of Video Streaming Application over GPRS/EGPRS. IEEE GLOBECOM (2002) 2478-2482
4. Asit Dan, Dinkar Sitaram: Scheduling Policies for an On-Demand Video Server with Batching. ACM Multimedia (1994)
5. Subhabrata Sen, Jennifer Rexford, Don Towsley: Proxy Prefix Caching for Multimedia Streams. IEEE INFOCOM (1999)
6. WAP Forum: WAP UAProf. Version 20-Oct-2001 (2001)
7. Gene Cheung, Takeshi Yoshinura: Streaming Agent: A Network Proxy for Media Streaming in 3G Wireless Network. IEEE Packet Video Workshop (2002)
8. Bin Wang, Subhabrata Sen, Micah Adler, Don Towsley: Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution. IEEE Transactions on Multimedia (2002) 366-374
9. Miikka Lundan, Igor D.D. Curcio: 3GPP Streaming over GPRS Rel.'97. The 12th International Conference on Computer Communications and Networks (2003) 101-106

A Protocol Switching Scheme for Developing Network Management Applications

Hyeokchan Kwon, Jaehoon Nah, and Jongsoo Jang

Electronics and Telecommunications Research Institute
161 Gajeong-Dong, Yuseong-gu, Daejeon, South Korea
{hckwon, jhnah, jsjang}@etri.re.kr
http://etri.re.kr/

Abstract. In this paper, we present a Protocol Switching(PS) Scheme to find efficient agent migration strategy for developing network management application. The purpose of the proposed scheme is to set up the best migration plan of mobile agent in order to minimize network execution time. To verify the effectiveness of the proposed scheme, we designed a network traffic estimation model for three distributed paradigms i.e RPC(Remote Procedure Call), mobile agent and mobile agent with locker pattern, and we evaluated PS scheme by simulation

1 Introduction

A mobile agent is an executing program that can migrate during execution from machine to machine. In RPC(Remote Procedure Call) and multi agent, the tasks are performed by global communication with remote site, whereas mobile agent performs the task by migrating a whole computational component, together with its state, the code, and some resources[1][2].

Many researchers suggest that a major benefit provided by mobile code is the capability to reduce network communication by moving client's knowledge close to server's resources, thus accessing them locally. By moving to the location of information resource, the agent can search the resource locally, eliminating the transfer of intermediate results across the network and reducing end-to-end delay. Recently, for this benefit of mobile agent, the demand of applying mobile agent to distributed systems such as information retrieval, network management, and electronic commerce has been increased.

But the question of whether the system using mobile agent is bringing significant benefits to the performance of distributed applications against traditional approach is an open one. Various parameters must be considered to evaluate performance of a paradigm used in developing distributed application.

In this paper, we present a Protocol Switching(PS) Scheme to find efficient agent migration strategy for developing network management application. The purpose of the proposed scheme is to set up the best migration plan of mobile agent in order to minimize network execution time.

In order to verify the effectiveness of the proposed scheme, we designed a network traffic estimation model for three distributed paradigms i.e RPC(Remote Procedure

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 862 – 872, 2006.

Call), mobile agent and mobile agent with locker pattern. In case of locker pattern, mobile agent temporarily stores data in private. In this way it can avoid bringing data that at the moment are not needed. On a later occasion, the stored data can be sent to client node. The three distributed paradigms considered in this paper are shown in fig. 1.

To evaluate performance of distributed paradigms, parameters such as CPU costs, memory usage and network traffic etc, should be considered. But in this paper we only concern network traffic. We are going to extend the model to consider additional parameters in the future.

In the previous researches such as [3], they developed simple network management system using mobile agent. But they did not precisely compare and evaluate performance of centralized approach based on SNMP protocol with distributed approach based mobile agent in network management applications. They only compared those performances by measuring response times of specific request. [3],[4],[5] proposed mobile agent based network management application, but they use only mobile agent for developing network management application. They do not consider other paradigms such as RPC, locker pattern.

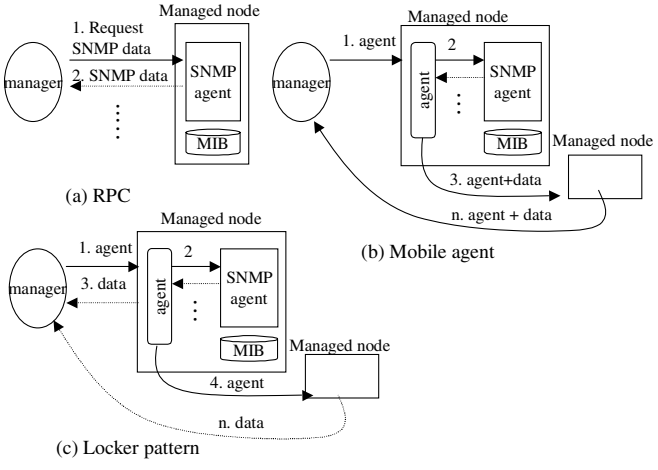


Fig. 1. Three distributed paradigms for developing network management application

The paper organized as follows. Section 2 introduces network management systems. Section 3 presents network traffic estimation model and simulation results for uniform network environment. The refined model that takes into account non-uniform network is presented in section 4. Section 5 presents a PS(Protocol Switching) scheme and simulation results. Finally conclusion is given in section 6.

2 Overview of Network Management System

Network Management can be classified into mainly two separate worlds: IETF management which relies on the Simple Network Management Protocol(SNMP), and one

of its derivatives, ISO management which relies on the Common Management Information Protocol(CMIP)[1],[2]. Both protocols assume to adapt the centralized management architecture based on a client-server paradigm. Currently SNMP Protocol based on RPC(Remote Procedure Call) is widely used because its architecture is simple and easy to be implemented. However the centralized approach can increase overall network traffic as well as the network manager's work in that most processing load converges into the network manager. To solve this problem, the demand for mobile agent technology to network management system has been actively attempted [3],[4],[5],[8],[9],[10].

A typical network management system is shown in Fig. 2[6],[7],[8]. A typical network management system comprises of one or more Network Management Stations(NMSs) which interact with the SNMP agents located at the network component such as host, server, router, gateway, bridge and hub etc. utilizing a particular protocol such as SNMP, CMIP etc. The information communicated between the NMS and SNMP agents is defined by a Management Information Base(MIB). The NMS kernel obtains the management information from either the SNMP agent or DB and provides it to the management applications. The NMS can be logically sub divided into a manager and a server. The manager comprises of the GUI and applications that perform network management activity, while the server comprises of the NMS kernel that obtain, saves and provides the information to the manager part of the NMS. NMS overlooks the management. It queries the network components on a timely basis to determine the health of the network. In this case the DB stays with the NMS and it is the central NMS that communicates with the SNMP agents.

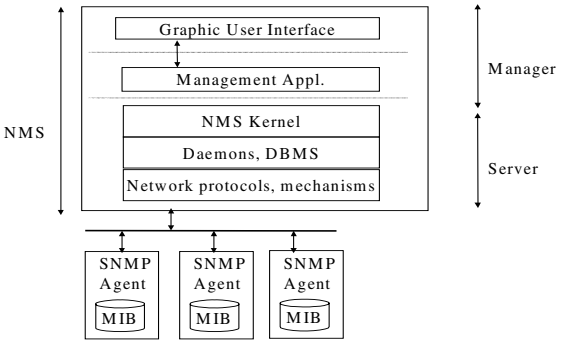


Fig. 2. The typical Network Management System

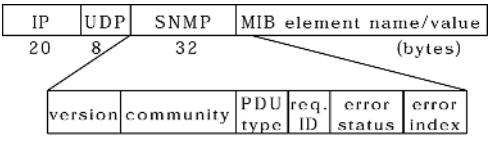


Fig. 3. SNMP message format

The network manager communicates with the SNMP agents using *get-request*, *get-response*, *get-next-request* and *set-request* primitives. They interact each other through RPC mechanism based on client-server paradigm. *get-request* message is used when network manager request management information to SNMP agent. And *get_response* message is used when SNMP agent return requested management information to network manager. *set_request* message is used when network manager altering management information of managed node. The message format of SNMP protocol is shown in Fig. 3.

3 Network Traffic Estimation Model for Uniform Network

3.1 Network Traffic Estimation Models

In this section, we present a network traffic estimation model for a uniform network. The equations presented in this paper are used for processing *get_request* and *get_response* messages of SNMP protocol. In network management system, the most important part is the processing of *get_request* and *get_response* messages [5],[6], [7],[11]. We do not consider multicast. Parameters are shown in Table 1.

Table 1. Parameters

Prm.	Meaning
N	The number of managed node
R_i	The number of request to node I
S_{nr}	The number of SNMP variables for r'th request of node n.
L_V	The size of SNMP variable name
L_S	The size of message header for transferring SNMP variables (IP+UDP+SNMP header)
L_T	The size of message header for transferring TCP data (IP+TCP header)
L_{VB}	The size of variable binding(SNMP variable name + value)
δ	Network delay
β	Network bandwidth
S_A	The size of Mobile Agent

In case of RPC, representative client-server model, the network manager send list of SNMP variables to the managed node for determine the health of the network. Then the SNMP agent of managed node return values of requested SNMP variables. Equation 1 shows overall network load of RPC.

$$L_{RPC} = \sum_{n=1}^{N} \sum_{r=1}^{Rn} (2L_S + (L_V + L_{VB})S_{nr}) \tag{1}$$

The amount of $L_S + L_V S_{nr}$ network load is required for r'th request of SNMP variables from network manager to managed node N. And the amount of $L_S + L_{VB} S_{nr}$ network load is required for reply requested information back to the network manager at node N.

Equation (2) shows total network execution time of RPC.

$$T_{RPC} = 2\sum_{n=1}^{N}\sum_{r=1}^{Rn}\delta + \frac{L_{RPC}}{\beta} \tag{2}$$

It requires $2\sum_{n=1}^{N}\sum_{r=1}^{Rn}\delta$ delays for Rn request and response of SNMP variables for each node. Mobile agent moves itself with collected data such as SNMP variable name, value etc. Equation 3 and 4 shows network load and network execution time for migrating a mobile agent respectively.

$$L_{MA} = \sum_{n=0}^{N}(S_A + \sum_{i=1}^{n}\sum_{r=1}^{Rn} S_{nr}L_{VB}) \tag{3}$$

$$T_{MA} = (N+1)\delta + \frac{L_{MA}}{\beta} \tag{4}$$

N+1 network delay are required for agent migration. In the equation 3, $\sum_{i=1}^{n}\sum_{r=1}^{Rn} S_{nr}L_{VB}$ stands for the size of previously accumulated SNMP variables. The size of mobile agent to be transferred , SA in equation 3, is shown in equation 5 and fig. 4.

$$\begin{aligned} S_A &= M_A + (M_A / payload) * (IPheader + TCPheader) \\ M_A &= Agent\_header + M \\ M &= M\_state + M\_code + M\_data \end{aligned} \tag{5}$$

M denotes the size of mobile agent itself that contains agent code, data and state. For transferring agent, an agent is attached to the next of an agent header, and M_A is segmented into TCP messages. The payload in equation 5 is TCP payload. Each TCP message is attached to the next of a TCP header and encapsulated into an IP packet[11].

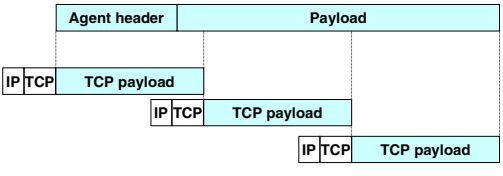


Fig. 4. Transferring Agent Format

In case of locker pattern, agent moves itself without accumulating SNMP data. For each node, collected SNMP data is sent to network manager directly. Equation 6 and 7 shows network load and network execution time for locker pattern respectively.

$$L_{LOC} = \sum_{n=0}^{N}(S_A + L_T + \sum_{r=1}^{Rn} S_{nr}L_{VB}) \tag{6}$$

$$T_{LOC} = 2N\delta + \frac{L_{LOC}}{\beta} \tag{7}$$

In the equation 6, $\sum_{r=1}^{Rn} S_{nr}L_{VB}$ stands for the size of transferred data to the network manager at node n. It requires 2 delays for agent migration and response of SNMP variables for each node.

3.2 Simulation

In this section, we present simulation results. The parameter values are shown in Table 2. We assume R_i , the number of request to node i, and S_{nr} ,the number of SNMP variables for r'th request of node n, has same average value for each node. UDP protocol is less reliable than TCP, however UDP consumes less network times than that of TCP[12]. Locker pattern and mobile agent uses TCP protocol, so the IP and TCP header is attached to the message sent by locker pattern and mobile agent. However, RPC approach uses SNMP and UDP protocol, so SNMP, IP and UDP header is attached to the message sent by SNMP protocol. For this reason, SNMP message is about 20bytes bigger than TCP message[6][12].

Table 2. Parameter values for Simulation

Prm.	N	R_i	S_{nr}	L_v	L_s
Value	20	5	15	5B	60B
L_T	L_{VB}	δ		β	S_A
40B	10B	20ms(TCP), 2ms(UDP)		400KB/s	6KB

Fig. 5 shows the network execution time while varying the number of network node N between 5 and 95. Fig. 5 shows that locker pattern has relatively less network execution time than RPC and mobile agent. The usage of mobile agent is better then that of RPC in case that the number of node is less than 65. In mobile agent, it consumes a little time to make the network connection, whereas it consumes much more time to transfer mobile agent and its accumulated data.

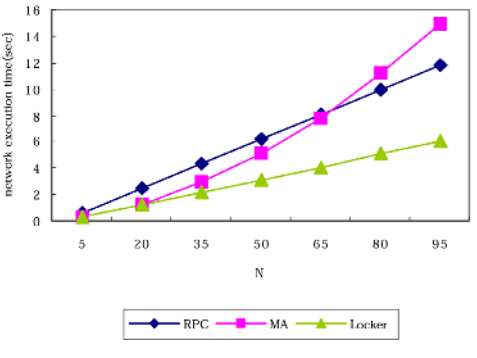


Fig. 5. Network execution time vs. number of nodes

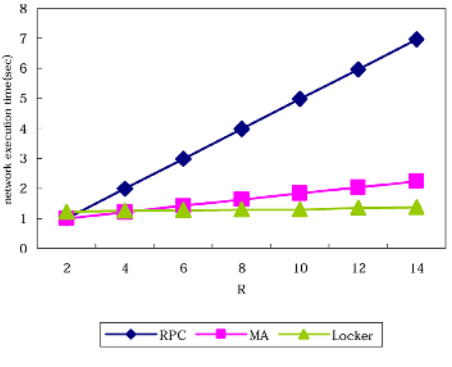


Fig. 6. Network execution times vs. R

Fig. 6 compares the network execution time while varying average number of request R between 2 and 14. Fig. 6 shows that the usage of mobile agent and locker pattern is better than that of RPC. In RPC, It normally needs to make many remote communications, which produces a great deal of network delay, whereas it needs far less time in migrating over network.

Fig. 7 compares the network execution time while varying the number of requested SNMP variables, for one request, S between 5 and 65. Fig. 7 shows that mobile agent has least network execution time in case that S is smaller than 14, but in case that S is better than 54, mobile agent has biggest network execution time.

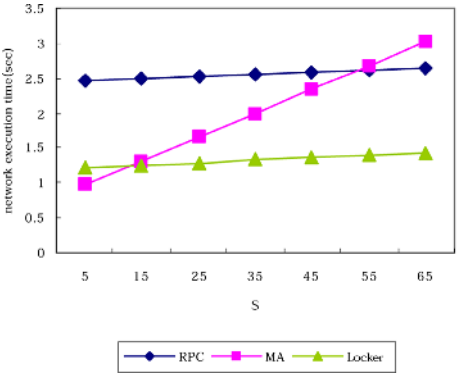


Fig. 7. Network execution times vs. S

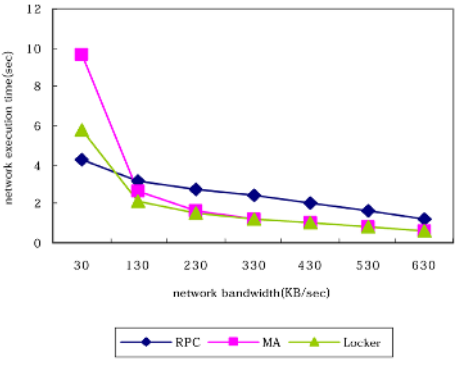


Fig. 8. Network execution times vs. δ

Fig. 8 compares the network execution time while varying the network bandwidth between 30 and 630KB/sec. In case that network bandwidth is 630KB/sec, we consider network delay of mobile agent and locker pattern that use TCP protocol is 11ms, and it is increased 3ms while network bandwidth is increased 100KB/sec. In case of RPC which uses UDP protocol, we consider the network delay is 60% of that of mobile agent and locker pattern.

Fig. 8 shows that locker pattern has relatively less network execution time those of RPC and mobile agent. The usage of mobile agent is better than those of locker pattern and RPC when network bandwidth is higher than 430KB/s. The usage of RPC is better than those of mobile agent and locker pattern when network bandwidth is less than 90KB/s. In mobile agent and locker pattern, the network execution time increased rapidly while network bandwidth increased from 30KB/sec to 130KB. The accumulated data of mobile agent and locker pattern is higher than that of RPC. So the performance of mobile agent and locker pattern is much influenced by network bandwidth. But, in this simulation, when network bandwidth is higher than 130KB/sec, the network delay has much influence on total network execution time than network bandwidth. That is because, in case of network management application, the size of collected data is relatively small in comparison with other applications such as data mining, information retrieval etc.

In this simulation, locker pattern generally performs better than mobile agent and RPC. And mobile agent shows better performance than RPC does when it requires frequent remote communications or when the condition of given network is good as well.

We can estimate network execution time of each paradigms by applying various parameter values to the network traffic estimation model presented in this section.

4 Network Traffic Estimation Model for Non-uniform Network

In this section, we present network traffic estimation model for non-uniform network. Fig. 9 shows the examples of non-uniform network constructed by three sub networks. For each sub network, the network bandwidth is β_1, β_2, β_3 and the network delay is δ_1, δ_2, δ_3 respectively. The β_{01} stands for network bandwidth between network manager and sub network whose network bandwidth is β_1. The additional parameter for the non-uniform network is shown in table 3.

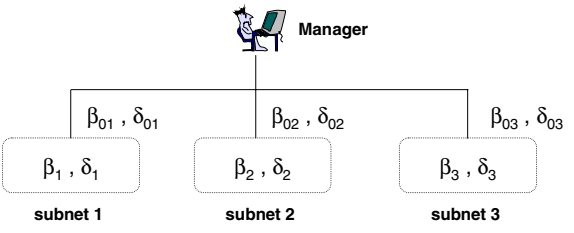


Fig. 9. Non-uniform network

Table 3. Additional Parameters

N_N	The number of node for each sub network
N_S	The number of sub network

Equation 8 shows overall network execution time for RPC.

$$T_{RPC} = \sum_{j=1}^{N_S}(((\sum_{n=1}^{N_N}\sum_{r=1}^{Rn}(2L_S + (L_V + L_{VB})S_{nr})(1/(\min(\beta_{0j}, \beta_j)) + 2\sum_{n=1}^{N_N}\sum_{r=1}^{Rn}\max(\delta_{0j}, \delta_j)) \quad (8)$$

min is the function of minimum value to be returned from its parameters, and *max* is the function of maximum value to be returned from its parameters. The network bandwidth between two nodes located in different sub network is minimum value of the two nodes[12]. In the equation 8, $L_S + L_{VB}S_{nr}$ is the size of transferred data to network manager for r'th request at node n.

Equation 9 shows overall network execution time for mobile agent. In the equation 9, $\sum_{i=1}^{j*N+n}\sum_{r=1}^{Rn}(S_{nr}L_{VB})$ stands for the size of previously accumulated SNMP variables at node n of sub network j.

$$T_{MA} = \sum_{j=0}^{N_S-1}(\sum_{n=1}^{N_N}(S_A + \sum_{i=1}^{j*N_N+n}\sum_{r=1}^{Rn}(S_{nr}L_{VB}))(1/\min(\beta_{0j}, \beta_j)) + \max(\delta_{0j}, \delta_j)) \quad (9)$$

Equation 10 shows overall network execution time for locker pattern. The network delay for each sub network is $2N_N \max(\delta_{0j}, \delta_j)$,and $N_S \times 2N_N \max(\delta_{0j}, \delta_j)$ is total network delay.

$$T_{MA(L)} = \sum_{j=1}^{N_S}(\sum_{n=1}^{N_N}(S_A + L_T + \sum_{r=1}^{Rn}S_{nr} * L_{VB})(1/\min(\beta_{0j}, \beta_j)) + 2\max(\delta_{0j}, \delta_j))) \quad (10)$$

5 Protocol Switching Scheme

In this section we propose a PS(Protocol Switching) scheme that determines efficient interaction patterns out of a few paradigms. The purpose of PS scheme is to set up the best interaction patterns with regard to minimizing network execution time. The PS scheme is shown below. The network traffic estimation model in PS scheme is that of section 3 and 4

PS Scheme

Input: Network Parameters for each subnet – table 1,2

Output: Selected_Paradigm_List (List of Interaction Pattern for each subnet)

PS_{time}, RPC_{time}, MA_{time}, $LOCKER_{time}$ = 0;

For each sub network i

RPCi = Network execution times for RPC that is calculated by equation 1 and 2

MAi = Network execution times for Mobile Agent that is calculated by equation 3 and 4

LOCKERi = Network execution times for Locker pattern that is calculated by equation 5 and 6

Selected_Paradigm_List[i] = *Min*(RPCi, MAi, LOCKERi)

PS_{time} = PS_{time} + Selected_Paradigm_List[i] $_{time}$

RPC_{time} = Total network execution times for RPC that is calculated by equation 8

MA_{time}, = Total network execution times for RPC that is calculated by equation 9

$LOCKER_{time}$ = Total network execution times for RPC that is calculated by equation 10

Return *Min*(PS_{time} , MA_{time}, RPC_{time} , $LOCKER_{time}$)

We evaluated the scheme by simulation. The parameter values are shown in table 2 in section 3.2. We considered that the network consists of three sub networks is shown in table 4, and the network manager is located in sub network 1. In table 4, network delay is for TCP connection, and in case of UDP 10,13 and 17ms for each sub network.

The interaction pattern determined by PS scheme is shown in table 5. In this simulation, we consider mobile agent visit node by decreasing order of network bandwidth to reduce network execution time.

Table 4. Considered network environment

Sub net.	Network bandwidth	Network delay	Managed node
1	1.3MB/sec	12ms	15
2	230KB/sec	23ms	10
3	30KB/sec	29ms	7

Table 5. Interaction pattern from PS scheme

Sub net1	Sub net2	Sub net3
Mobile agent	Locker pattern	Mobile Agent

Fig. 10 compares network execution times to each other calculated by network traffic estimation model for non-uniform network presented in section 4. We can see that the interaction pattern determined by PS scheme has the smallest network execution time.

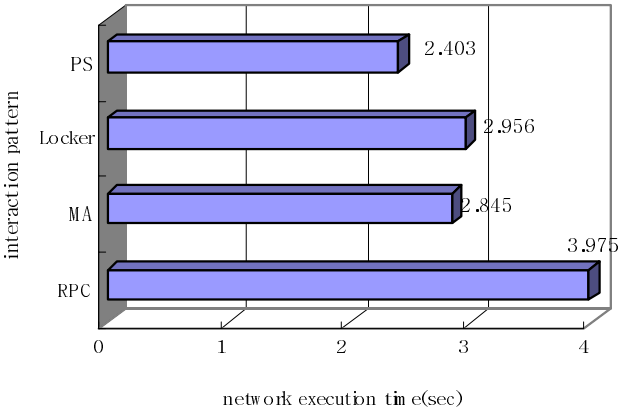


Fig. 10. Simulation Result

6 Conclusions

In this paper, we present a Protocol Switching(PS) Scheme to find efficient agent migration strategy for developing network management application. From this research, we can see that the network execution time varies according to interaction patterns from the given paradigms. This research helps us to decide convenient interaction pattern in specific application domain for developing distributed applications. In the future work we will extend the model to consider additional parameters such as CPU costs, memory usage and so on. And we will extend the model so that it can be applied to other applications such as information retrieval, electronic commerce and so on.

References

1. Bic, L. F., fukuda, M., and Dillencourt. Distributed computing using autonomous objects. IEEE Computer, Aug. 1996.
2. Wooldridge, M. and N. R. Jennings. Agent Theories, Architectures and Languages: A Survey. Intelligent Agent, pp.1-39, Springer-Verlag, Germany, 1995

3. M. G.Rubinstein, O.C.M.B.Duarte, G.Pujolle, “Improving Management Performance by Using Multiple Mobile Agents,” Proc. 4th International Conference on Autonomous Agents, pp.165-166, 2000
4. Theodore Kotsilieris, Angelos Michalas, Stylianos Kalogeropoulos, George Karetsos , Moshe Sidi and Vassilios Loumos, “Performance optimization of Network Management Applications based on Mobile Agents”, Informatik Forum Journal Special Issue on Mobile Agent Technology, December 2002, p.p. 82-90, ISSN 1010-6111
5. T.White, B.Pagurek and A. Biezcza, “Network Modeling for Management Applications Using Intelligent Mobile Agents,” Journal of Network and Systems Management, Sep. 1999
6. W. Stallings, SNMP, SNMPv2, SNMPv3 and RMON 1 and 2, Addison Wesley, 1999
7. SNMPv3, http://www.snmplink.org
8. A. Bieszczad, T. White and B. Pagurek, "Mobile Agents for Network Management,” IEEE Communications Surveys, Sep. 1998
9. G.P. Picco, M. Baldi, “Evaluating tradeoffs of Mobile Code Design Paradigms in Network Management Applications,” Proc. of the 20th International Conference on Software Engineering(ICSE98), Japan, 1998
10. Stylianos Kalogeropoulos, George Karetsos, Angelos Anagnostopoulos, Theodore Kotsilieris, Angelos Michalas and Vassilis Loumos, "A Methodology for Improving the Performance of Agent-Based Applications Through the Identification of the Optimal Number of Mobile Agents", Autonomous Agents and Multi-Agent Systems, 2002
11. D. Griffin, G. Pavlou, P. Georgatsos, Providing Customisable Network Management Services Through Mobile Agents, Proceedings of the 7th International Conference on Intelligence in Services and Networks (IS&N 2000).
12. D. B. Lange, M. Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998
13. W. Stallings, Data and Computer Communications (6th edition), Macmillan Publishing Company, 2000

Multimedia Traffic Load Distribution in Massively Multiplayer Online Games

Hyungjune Im, Hyunchul Kim, and Kilnam Chon

Department of Computer Science,
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

Abstract. Most Massively Multiplayer Online Games use client-server architecture and thereby bottlenecks in a single server have been a hot issue. One feasible option for the streaming scenario is to consider the federated peer-to-peer model, where multiple reflectors are put between clients to deliver packets. In this paper, we find that although the reflectors may handle multimedia stream to a certain degree, they often become bottlenecks, particularly when client distribution is skewed. To solve the reflector bandwidth bottleneck problem of the federated peer-to-peer model, we propose a traffic distribution model, and then show that the proposed scheme successfully reduces the peak traffic of an overloaded reflector to less than twice the average traffic size. To evaluate the efficiency of the proposed scheme, we also measured the duration of the distribution process and the additional load of the peak, which is caused at the initial buffering process just before the distribution.

1 Introduction

A Massively Multiplayer Online Game (MMOG), or simply an online game, is a system related to a networked virtual environment and a network computer game [6]. Online gaming has become one of the most popular applications on the Internet. Famous titles such as Ultima Online, EverQuest, Lineage, and Final Fantasy XI have more than 100,000 subscriptions [1].

A typical MMOG system has a server farm and the scalability problem of game servers has been addressed [3], [8]. One factor that obstructs scalability and flexibility is the crowd. For the last decade, the crowded environment problem has been often addressed in area of Distributed Virtual Environment (DVE) [5], [9]. In a crowded environment, many players receive packets from and send packets to one another, and resource usage increases quickly. Lineage, for example, periodically holds a siege attack event, and participants from various clans number more than a hundred. In such cases, the region obviously becomes a hot spot. The number of messages generated by interactions of neighboring players is from tens to hundreds times, or much higher, in a crowded environment than in a sparse environment.

With advancement in game technology and greater demands from players, audio streaming was realized in conventional video games and computer games. These games enable players to use voice chatting, and players no longer need

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 873–882, 2006.

to type on keyboards to chat. We expect multimedia streaming, including video streaming, to soon be supported in this area, thereby boosting the potential for immersive communication, which is similar to video conferencing. MMOG systems with attached audio and video streaming have already been introduced [7], [11].

The federated peer-to-peer model [3] is a recently proposed alternative to the client-server model in the multimedia streaming scenario, where multiple reflectors - that is, hosts put between clients and the server - split up the total network traffic. The client-server architecture moves the entire stream traffic through a dedicated server and puts a severe overhead on the server. In the federated peer-to-peer architecture, the entire game world is subdivided into multiple groups, and clients only exchange packets with members in the same group through a reflector. Members in each reflector are much fewer than the population of the entire game.

Our contributions in this paper are twofold. First, we find that even the federated peer-to-peer architecture is not scalable in a crowded environment, particularly in multimedia-enabled online games. A reflector that handles a crowded group faces a similar problem to that of the client-server model and the reflector's network bandwidth is saturated due to the large stream volume. Second, therefore, to solve the scalability problem, we propose a traffic distribution model for the federated peer-to-peer architecture and evaluate it. Our distribution model effectively reduces the peak traffic load in the overloaded reflector with reasonable costs.

The rest of the paper is organized as follows: In section II, we introduce the federated peer-to-peer model and discuss its scalability problem in handling multiple concurrent streams generated by users. In section III, we suggest the requirements for analyzing the system and explain our solution to the problem in detail. In section IV, we present and analyze the simulation results. Finally, in section V, we offer our conclusions.

2 Problem Definition

2.1 Federated Peer-to-Peer Model

Rooney et al. suggested the federated peer-to-peer model as a flexible game architecture [3]. They claimed that the problem with the client-server model is due to computation power of the server, and that the problem with the pure peer-to-peer model is due to the network bandwidth of each peer. Their solution was to put reflectors between the client and the server, as shown in Fig. 1, enabling the reflectors to act like proxy servers.

An entire game is divided into many groups, and the control server assigns each group to a reflector. In Fig. 1, there are two groups, B1 and B2. The control server associates the group B1 to Reflector 1 and the group B2 to Reflector 2, respectively. Instead of a central server, a client computes game decisions in a similar way to thepure peer-to-peer model and sends messages, including

computed results, through a responsible reflector to other clients in the same group - for example, its neighbors or party members. The reflector forwards replicated packets to destined clients.

The federated peer-to-peer model is a sound architecture to support multimedia streaming and preferable to either client-server model or pure peer-to-peer model for the following reasons. First, the federated peer-to-peer model is more scalable than client-server model. If a company puts a communication server beside the game servers to support all the users, the bandwidth bottleneck occurs quickly because of the heavy stream traffic. Multiple reflectors disseminate stream traffic and therefore are more scalable. Second, the federated peer-to-peer model is more reliable than pure peer-to-peer model. Peers have less resource than reflectors, and they are not as reliable as reflectors. Peers are not appropriate to forward all the stream packets to other peers by themselves because it needs much more bandwidth than they have.

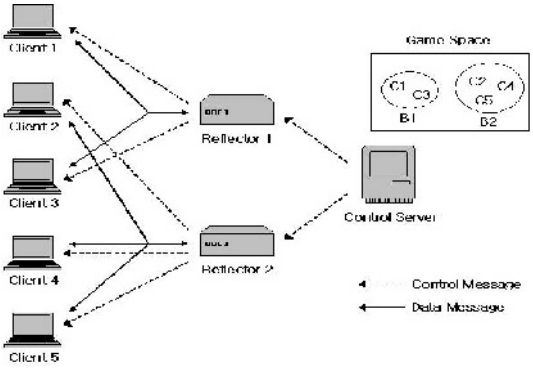


Fig. 1. Federated peer-to-peer model [4]

2.2 Traffic Overload in Crowded Environment

In the federated peer-to-peer model, a group refers to N players that gather in a small region. When everyone receives packets from every member in the same group, the theoretical message complexity of the system is $O(N^2)$ [12]. In a crowded environment, N increases quickly and the complexity becomes so high that a responsible reflector cannot support the multimedia streaming of all the players.

Fig. 2 shows the simulation results of the multimedia stream traffic generated in a reflector with various crowded environments. The graph shows that the maximum traffic load generated in a reflector that responds to a crowded group is much higher than the average stream traffic rate. In the simulation, the average number of clients in a group is ten. Without a skewed distribution, the probability of a client joining group P becomes $1/(totalnumberofgroups)$ – P = 0.01 with 100 groups and 1000 users, P = 0.005 with 200 groups and 2000 users, and so on.

We tested various probabilities for a client joining a predefined group that acts as crowded region. The number of clients was 3000 and the crowded probability P varied from 0.1 to 0.3. We expected the size of the crowded group to be 300, 600, and 900, respectively. In this test, no clients received more than ten streams concurrently from other members. We expect that clients have less bandwidth than the servers, and clients cannot receive streams from everyone in the same region when they belong to the crowded group. This paper proposes a solution to this problem by distributing the stream traffic from the overloaded reflector to relatively free hosts.

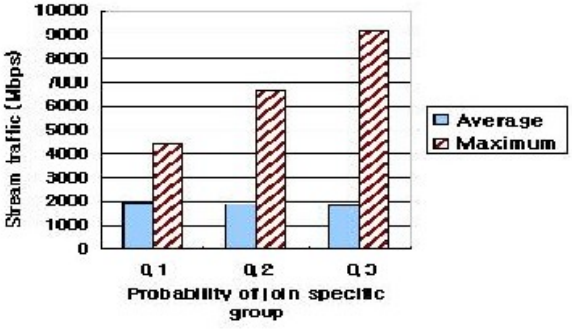


Fig. 2. Stream traffic comparison with various probabilities of crowd

3 Proposed Traffic Distribution Model

In traffic distribution of a crowded environment, some factors must be investigated for efficient achievement. They give criteria to measure the performance of the traffic distribution model. Previous work on load balancing in a virtual environment indicates some parameters for this subject [5], [9]. The major performance requirements of this work are the distribution performance, time, and cost.

Fig. 3 is a simple diagram of the system and approach scenario. The entire procedure has the following five steps, which are illustrated in the diagram with numbers and arrows.

1) Each reflector continuously measures the load rate whenever a client joins or leaves and it periodically reports the results to the server. The load rate of one reflector, L, is calculated as,

$$L = (size\ of\ group\ 1)^2 + (size\ of\ group\ 2)^2 + ... + (size\ of\ group\ N)^2\ . \quad (1)$$

2) The reflector notifies the server when the stream traffic rate exceeds a given threshold, which must be under the available bandwidth of the host. Once players swarm in a specific, the rests of the players are likely to join there and the population keeps growing.

3) The server compares the load rate of the reflectors collected at step 1 to decide how to redistribute the traffic and to determine who takes over the task. Another

reflector called a helper is a new participant that reports the least traffic load recently.
4) In the buffering step, the stream packet sender is the overloaded reflector and the receiver is nominated helper server. Unless the packets are copied to a new destination before the clients arrive, the users experience a pause in their reception of voice and video data.
5) The excessive clients leave the sender side and join the receiver side. If a crowded group is divided and simultaneously supported by the sender and the receiver, the group's members can still communicate with each other.

The sender checks whether the stream traffic diminishes below a given value (which differs from the threshold value) after the entire process ends. If not, it repeats the process from step 2 until the condition is satisfied.

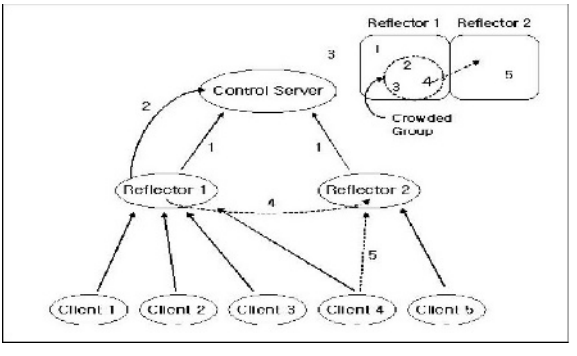


Fig. 3. System diagram and traffic distribution process

4 Performance Evaluation

4.1 Traffic Overload in Crowded Environment

For the simulation, we use a Java-based network simulation tool called Scalable Simulation Framework Network Models (SSFNet) [10] to implement MMOG environment on a Linux host. To configure the simulation, we specified several parameters to realize an MMOG-like experiment. Some of the parameters explicitly refer to previous measurements and papers [2], [8]. The rest of the parameters are inducted from apparent features of online game.

We put ten reflectors in this virtual system. The system has two main independent variables - the number of clients and the probability of crowding. The number of clients varied from 100 to 5000. Probability of crowding varies from 0.1, to 0.2, to 0.3, refers to the number of clients likely to join a predefined group. With a probability is 0.2, the size of the crowd is expected to be one fifth of the total population.

In addition, we set the overload threshold, at which the reflector raises an alarm, to three times the average stream rate. We calculated the average rate

from preliminary experiments with various population sizes. The release value where the overloaded reflector judges that it is no longer overloaded was 70 percent of the threshold traffic.

The multimedia stream rate for one client is 1Mbps and the simulation run for 300 seconds. In the period from 0 to 60 seconds, clients randomly start running and connecting to the server. We repeated the simulation five times under specific conditions (namely, the number of users, probability of crowding).

4.2 Simulation Results

Fig. 4 is a representative example of traffic dissemination. Each line indicates the outbound stream traffic from each reflector. In the graph, the number of clients is 5000 and the probability of crowding, P, is 0.3. In this example, the threshold is 9 Gbps. The higher line (solid line) is a typical feature of an overloaded reflector, and the lower line (dashed line) is the average traffic of all the reflectors. The distribution phase occurs repeatedly, but the first occurrence (around 40 seconds) is important because the first one scatters most of the excess clients and charge. The measured data were all extracted from the first occurrence. The discrepancy between the peak point and the threshold is an additional cost, and it depends on the number of users and the probability of crowding. The overload does not happen when the probability, P, is 0.1.

Fig. 5 shows the traffic ratio of the highest traffic rate and the average traffic rate, which were measured immediately after the reflector entered the overload status and was released. As this rate is bigger, the gap between the overloaded reflector and the other reflectors becomes severe. The performance of the traffic distribution can be evaluated by how much the ratio decreases through the redistribution. The ratio at the point of the overload ranges from about 5.0 to 5.2 when P is 0.3, and about 3.2 to 3.4 when P is 0.2. When the redistribution is completed, the ratio shrinks to less than 1.5. It is relevant to the threshold and the release value. If a reflector perceives the released status at a much lower point than the simulation, the ratio will drops to less than 1.5, but it is more expensive and needs a longer time.

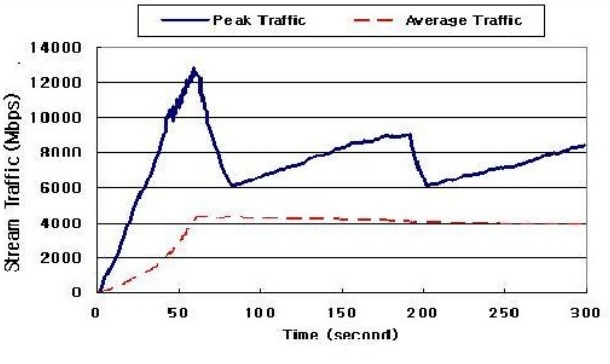


Fig. 4. Stream traffic line with traffic distribution

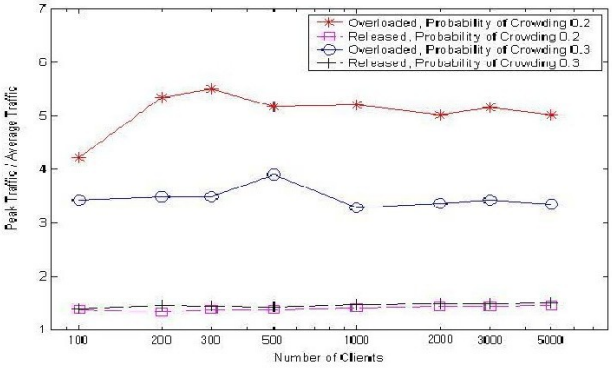


Fig. 5. Traffic distribution performance

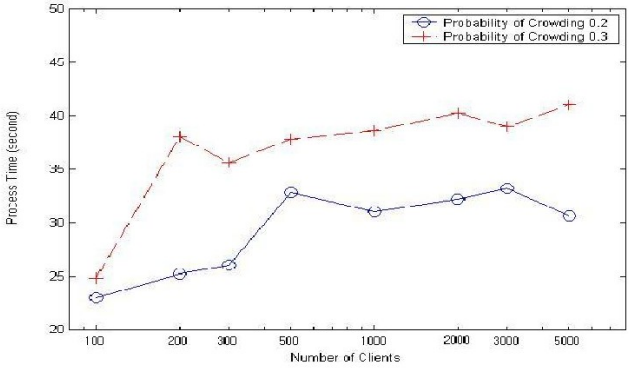


Fig. 6. Process time consumption

A reflector cannot serve the streaming process well if it stays overloaded for a long time. A short processing time is essential for a stable and effective load distribution. Process times taken for the proposed traffic distribution task differ only a few seconds for various population sizes, though it depends on the probability of crowding (Fig. 6). When P is 0.2, it takes around 30 - 33 seconds for any number of clients. As P increases to 0.3, it takes about 40 seconds in every case. An exception occurs when the number of clients is 100, in which case there is a large variation in the measured data.

The cost is measured and recorded in two ways. The first parameter is the number of control messages generated during the traffic distribution process. Fig. 7 shows the number of inbound and outbound control messages in overloaded reflector. This number grows proportionally as the number of clients increases. The reflectors continuously communicate with the control server and clients, especially when they change groups or reflector connections. Most of these messages were sent to and from clients who were in the crowded group but migrated to other reflectors.

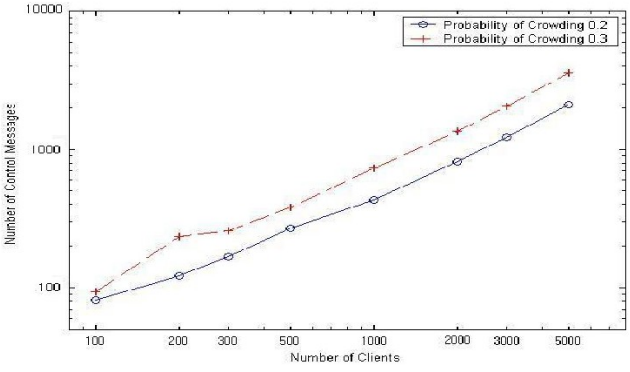


Fig. 7. Number of control messages

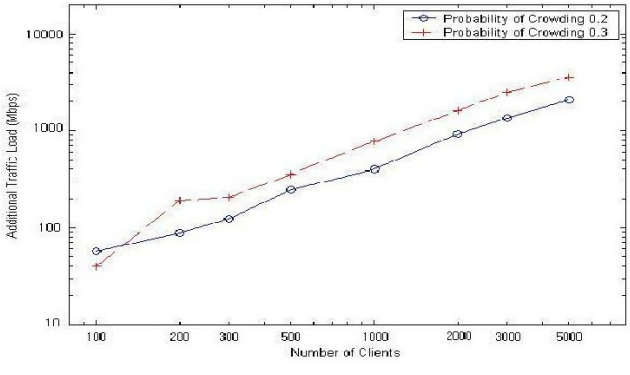


Fig. 8. Additional traffic load

Fig. 8 shows the traffic load that exceeds the threshold - that is the gap between the maximum value and the threshold traffic value. The overloaded reflector starts buffer-ing before the clients migrate and it generates additional outbound traffic. The popula-tion increases for the first 60 seconds, thereby expanding the additional traffic load. It also depends on the population and the probability, for instance, the number of control messages. Furthermore, it increases almost linearly in terms of the number of clients. As with the process time measurement, the cost parameters have a larger variation when the population is smaller. Tens of clients in a group is meaningless in the traffic distribution model.

4.3 Performance Analysis

The graph in Fig. 4 shows a continuous increment in the peak traffic even after the overload alarm has been raised. The increment indicates that the threshold value should not be the system's "Maginot Line." In the graph, the threshold is 9

Gbps but the peak traffic rises to 12.5Gbps before enough clients are delegated. If developers dynamically adjusted the threshold value, the system would be flexibly maintained.

A drop in the peak traffic is efficient when many users migrate at once. When a client receives a maximum of c streams, as in our simulation, and if the number of clients in a group, N, is larger than c, the total number of stream connections is cN. Further-more, if m clients migrate from a crowded group at the same time, the total number of outbound streams from the reflector shrinks. We also assume that any two clients in the same group can exchange streams even if they connect to different reflectors. After m clients migrate, the number of clients in the original reflector (that is, clients who belong to the crowded group) is $N-m$. Moreover, because the reflector now sends $c(N-m)$ streams to clients and $N-m$ streams to a helper reflector, the total number of streams becomes $(c+1)(N-m)$ and the condition for making the term smaller than cN is,

$$(c+1)(N-m) = cN + N - cm - m < cN \tag{2}$$

$$m > N/(c+1)\ . \tag{3}$$

The result of this equation indicates that at least $N/(c+1)$ clients must be moved to reduce the traffic in an overloaded reflector. Preferably as many clients as possible should be moved in order to efficiently reduce the traffic. However, it is prohibited by the high distribution cost in the simulation.

Most of the measured parameters are related to the number of users and the prob-ability of crowding. The same trend occurs in the traffic load that exceeds the thresh-old. If operators collect enough measurement data, they can foretell the maximum number of concurrent users in a region or in the entire world. We could then construct a plausible strategy to balance the overhead. Generally cost and time show a consistent pattern in relation to the growth in the number of clients (linearly grow or relevant), though this phenomenon does not occur if the number of clients is as small as a hundred.

Indiscrete concentration in a crowded group triggers a repetitive phase shift be-tween the overload status and the released status. In the real environment of an online game, users can arrive at different times. Because group changes seldom occur in the operation, the traffic concentration after the first overload occurrence is slower than at the beginning. However the traffic concentration is just as fast as the initial phase when the population size itself increases and new members join the group.

5 Conclusion

In this paper, we found that there is a scalability problem in the federated peer-to-peer model for multimedia streaming in MMOGs. The federated peer-to-peer model is an efficient model for multimedia streaming, though the model is still problematic when a game is overcrowded. A reflector suffering from a network traffic overload renders unsatisfactory service to users.

To solve this problem, we propose a load balancing feature in the system, and evaluate its performance, time consumption and cost. The results show that the peak, which is about five times greater than the average traffic load before the distribution process, decreases to less than one and half times the average traffic load after the task has been completed. Our simulation shows that excessive traffic over given threshold is not ignorable in an overloaded reflector. A sophisticated algorithm which subdivides a crowded group is necessary to remedy the situation by further research.

References

1. B. S. Woodcock : An analysis of MMOG subscription growth, http://www.mmogchart.com
2. J. Kim, E. Hong, and Y. Choi : Measurement and analysis of a Massively Multiplayer Online Role Playing Game traffic, in Proc. 1st APAN Network Research Workshop, Busan, South Korea (2003) 99-102
3. S. Rooney, D. Bauer, and R. Deydier : A federated peer-to-peer network game architecture, IEEE Communications Magazine, vol.42, no. 5. (2004) 114-122
4. D. Bauer and S. Rooney : The performance of software multicast-reflector implementations for multi-player online games, in Proc. 5th Networked Group Communication, Munich, Germany (2003) 214-225
5. Y. Choi : Load balancing for networked virtual environment servers using area split and merge, M.S. thesis, Dept. CS, KAIST, South Korea (2002)
6. J. Smed, T. Kaukoranta, and H. Hakonen : Aspects of networking in multiplayer computer games, in Proc. 1st International Conference on Application and Development of Computer Games in the 21st Century, Hong Kong, China. (2001) 74-81
7. P. Quax, T. Jehaes, P. Jorissen, and W. Lamotte : A multi-user framework supporting video-based avatars, in Proc. 2nd Workshop on Network and System Support for Games, Redwood City, CA. (2003) 137-147
8. B. Knutsson, H. Lu, W. Xu, and B. Hopkins : Peer-to-peer support for massively multi-player games, in Proc. 23rd Conference of the IEEE Communications Society, Hong Kong, China (2004) 96-107
9. T. Carneiro and J. Arabe : Load balancing for distributed virtual reality systems, in Proc. International Symposium on Computer Graphics, Image Processing and Vision, Rio de Ja-neiro, Brazil. (1998) 158-165
10. A. Ogielski, D. Nicol, and J. Cowie : Scalable Simulation Framework Network Models, http://www.ssfnet.org/
11. P. Boustead and F. Safaei : Comparison of delivery architecture for immersive audio in crowded networked games, in Proc. 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (2004) 22-27
12. M. R. Macedonia and M. J. Zyda : A taxonomy for networked virtual environments, IEEE Multimedia, vol.4, no. 1. (1997) 48-56.

A Realization Method of Voice over IP System Passing Through Firewall and Its Implementation

Masashi Ito and Akira Watanabe

Graduate School of Science and Technology, Meijo University, Japan
m0432004@ccmailg.meijo-u.ac.jp, wtnbakr@ccmfs.meijo-u.ac.jp

Abstract. In recent years, IP telephone has achieved remarkable progress on the Internet through "low price ", "continuous connection", and "high speed communication band". However, it is not easy to use IP telephone over firewall and NAT because of its restrictions of the communication. We have proposed the system called SoFW (SIP over Firewall) that suppresses the problem. In this paper, detailed functions and its implementation method of SoFW are described.

1 Introduction

Due to spread of broad band communications and development of backbone networks between ISPs, the transmission capacity of the network has been considerably increased. Therefore, the quality assurance of Voice over IP (VoIP, hereinafter) becomes a level of practical use, and it has become popular among enterprise networks and home networks.

However, in case of enterprise networks, Firewall (FW, hereinafter) are located between the enterprise network and the Internet, prevent the communications of VoIP between the terminals [1]. It is expected that expansion of the VoIP is further promoted if it can safely pass through FW.

There is a protocol based on the existing telephone specification referred to as H.323 [2] which is standardized in an early stage by ITU-T (International Telecommunication Union Telecommunication) as the session initiation protocol of VoIP. However, now, SIP (Session Initiation Protocol) [3] [4] standardized by IETF (Internet Engineering Task Force) owing to easy implementation and expandability, is being paid attention that it can be used for various kinds of multi-media services. SIP has been employed to most of VoIP systems provided by ISP [5] [6]. A SIP system consists of user agents and a SIP server, and provides functions of registering user locations for the SIP server, and of relaying dial messages based on its location. However, in the SIP system, it is needed that IP address of Callee terminal, or IP address of the SIP server to which the Callee terminal belongs to can be identified by caller when dial starts. For that reason, dialing can not be performed in the environment in which NAT (Network Address Translator) [7] exists between the communication terminals. Further, in most cases, FW limits the communications to the applications such as mail and Web server access from an inside of the enterprise network to the

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 883 – 891, 2006.

Internet, and blocks other communications. If VoIP is to be introduced in a network under such limitations, a security policy of the enterprise network must be changed, and degradation of security accompanied thereby possibly occurs.

Some systems have been proposed, in which VoIP can pass through the barrier of FW and NAT. They are, for example, HCAP [8], Skype [9], and SoftEther [10]. SoftEther enables any applications to path through FW and NAT, not limited to VoIP.

HCAP and Skype provide an HTTP tunnel between terminals in an enterprise network and a relay server on the Internet. Special applications are used for dialing, and voice streams are relayed with packets embedded in HTTP GET and POST messages. Thus, VoIP can pass through FW and NAT if the environment is capable of accessing web site on the Internet. However, there are problems that special functions are required for the terminals, and there is wasteful traffic caused by constant connections of HTTP flows on FW. In case of SoftEther, software referred to as Virtual LAN Card is implemented in a PC on a private address side, and software referred to as Virtual HUB is implemented in a PC on a global address side. Virtual IP address and MAC address are allocated in Virtual LAN Card. Virtual LAN Card and Virtual HUB construct a virtual Ethernet by embedding Ethernet frames in a protocol capable that can pass through FW and NAT, such as HTTPS, SSH or the like. Terminals connected to the virtual Ethernet can freely communicate across FW and NAT. In order to construct a VoIP system on the virtual Ethernet, a SIP server and VoIP terminals are to be connected to the virtual Ethernet. However, in this system, it raises problems that a network originally protected by FW is exposed to danger, and further, an integrated control of IP addresses in the virtual Ethernet is needed.

Thus, we have been proposed the system called SoFW (SIP over Firewall) to solve the problems. In SoFW, two types of relay agents are placed inside and outside of FW/NAT, one by one, and all messages of SIP and voice streams from terminals are passed through in HTTP tunnel made by the relay agents. Since SoFW realizes passages over FW and NAT only by adding relay agents, it does not affect existing systems. This becomes very effective in case that people in the company are already using the SIP based VoIP system. In this paper, outline of SoFW and its realization method are described in 2, implementation method is shown in 3, and conclusion is described in 4.

2 Outline of SoFW and Its Implementation

2.1 Outline of SoFW

Fig. 1 shows the configuration of SoFW. In SoFW, HRAC (Half Relay Agent Client) is placed in an enterprise network and HRAS (Half Relay Agent Server) is placed on the Internet. Prior to the telephone communication, an HTTP tunnel is generated between HRAC and HRAS, and the two devices are functioned as a virtual SIP server having interfaces of a global and a private IP address. Voice streams, are also relayed by the HTTP tunnel.

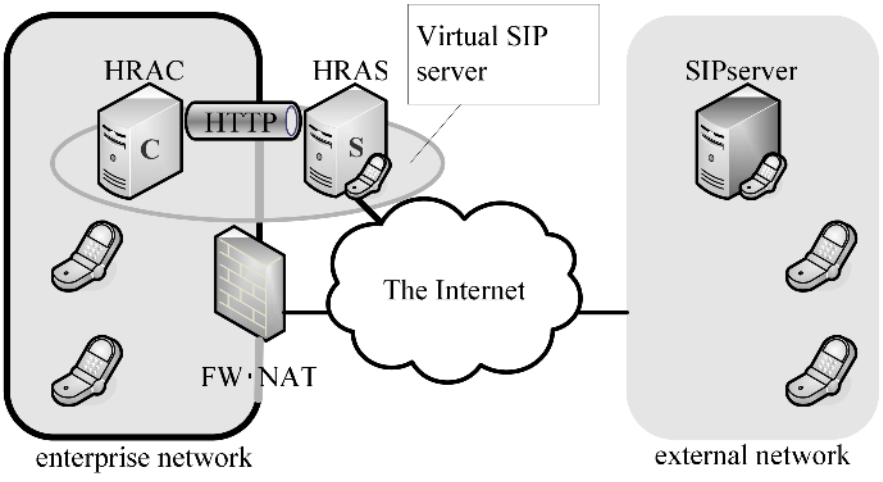


Fig. 1. Structure of SoFW

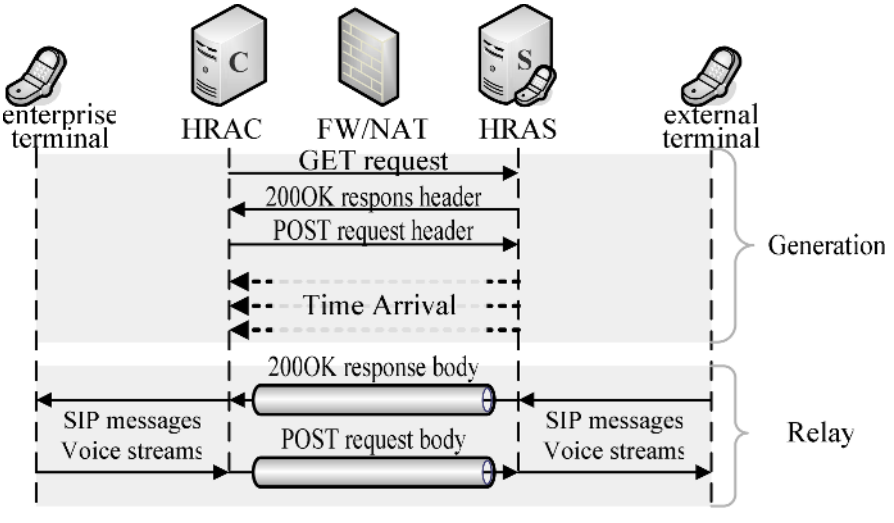


Fig. 2. Sequence of a generation of HTTP tunnel

2.2 HTTP Tunnel

Fig. 2 shows sequence of a generation of HTTP tunnel. HRAC establishes two TCP connections for a GET request and a POST request, defined by HTTP. When HRAS receives a GET request, it returns a header part of 200OK response. When the process

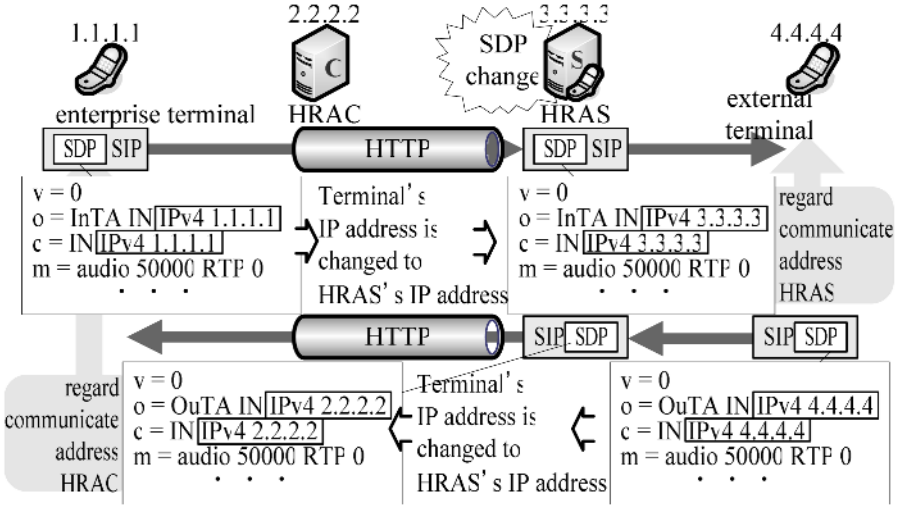


Fig. 3. Procedure of changing SDP contents

is completed, HRAC and HRAS wait for SIP messages from end terminals. HRAC embeds a receiving SIP message in a body part of a POST request and transmits it to HRAS. HRAS embeds a receiving message in a body part of a 200OK response, and transmits it to HRAC. HRAS sends to HRAC messages referred to as Time Arrival at every predetermined time during waiting time in order to keep the TCP connections alive.

2.3 Voice Stream Guidance by the Change of SDP Contents

SoFW relays not only SIP messages but also voice streams to the HTTP tunnels between HRAC and HRAS. However, in normal SIP specifications, voice streams are directly exchanged between end terminals. In SoFW, to guide the voice streams to the HTTP tunnel, when SIP messages reach HRAS in dialing phase, HRAS changes type values described in SDP [11], body part of SIP messages. Fig. 3 shows the procedure of changing SDP contents. Various kinds of information required for a voice communication is described in SDP as type values. The type values include IP addresses, port numbers and codec type which will be used for a voice communication by the terminals. In HRAS, an IP address of caller in SDP transmitted from an enterprise terminal is changed into the IP address of HRAS, and an IP address of callee in SDP transmitted from an external terminal is changed into the IP address of HRAC, respectively. The enterprise terminal, that receives the changed SDP, recognizes that the correspondent node is HRAC, and the external terminal recognizes that the correspondent node is HRAS, and thus, the voice streams are guided to the HTTP tunnel.

Table 1. Contents of RAT

Contents	Explanation
To	Information of destination terminal
From	Information of source terminal
Call-ID	Session descriptor
IIP	IP address of an enterprise terminal
IPort	Port number of an enterprise terminal
OIP	IP address of an external terminal
OPort	Port number of an external terminal

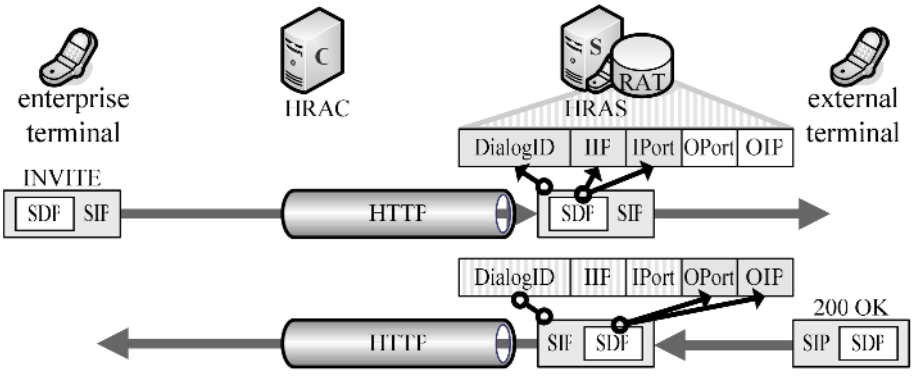


Fig. 4. A flow of RAT generation

2.4 Determination of a Routing Path of Voice Streams Using RAT(Relay Agent Table)

As described in 2.2, end terminals send voice streams toward HRAC or HRAS, thus HRAC and HRAS have to determine the right path of voice streams to the end terminals. In SoFW, RAT (Relay Agent Table) specific to SoFW, is generated in HRAS from the information of SIP header and SDP contents during dialing operations. The paths of voice streams are determined with reference to RAT during voice communications. Table. 1 shows contents of RAT. To, From, and Call-ID are obtained from SIP headers and they form a dialog ID which identifies the communication. Others are obtained from SDP contents, and IIP and IPort show an IP address and a port number of an enterprise terminal, and OIP and OPort show an IP address and a port number of an external terminal. Fig. 4 shows a flow of RAT generation when the dialing is started from an enterprise terminal. SDP is contained in INVITE message which is a start message of a caller and in 200OK which is the response of INVITE. When HRAS receives INVITE, it writes down the dialog ID, IIP and IPort in a RAT record. Next, when HRAS receives 200OK it retrieves the same communication of the RAT record from the dialog ID in the message, and adds OIP and OPort in the RAT record.

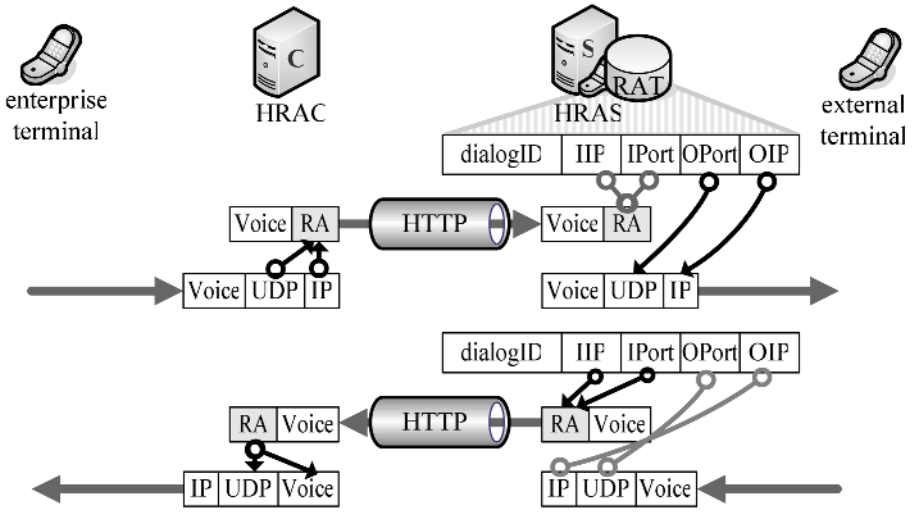


Fig. 5. Process flows of voice streams

When dialing operation is finished, voice communications start, and the path of the voice streams is determined by RAT in HRAS. Fig. 5 shows process flows of voice streams. When HRAC receives voice streams from an enterprise terminal, an IP address and a port number of the enterprise terminal are added to the voice data as an RA header, and the packet is relayed to HRAS. HRAS retrieves the corresponding RAT record from the information in the RA header, and changes the destination of the voice stream to the external terminal indicated in RAT and transmits the voice stream. When HRAS receives voice streams from an external terminal, HRAS retrieves the corresponding RAT record from a source IP address and a port number. The IP address and the port number of the external terminal are added to the voice data as an RA header, and the packet is relayed to HRAC. HRAC changes the destination address of the voice streams to the enterprise terminal indicated in the RA header, and transmits the voice streams.

When HRAS receives a BYE message which is a request for disconnection, a corresponding RAT record is retrieved from the dialog ID, and contents of the record is deleted.

3 Implementation Method

HRAC and HRAS have been implemented as applications on FedoraCore30 (linux2.6.9), and the function of HRAS has been realized by a cooperation with SER [12] which is free software of SIP server. Fig. 6 shows the function of HRAS and HRAC, and its data flow. Table 2 indicates functions of HRAC and HRAS. On the portion of dialing process in HRAS, other functions than SER are referred to as SIP Relay Server module. In dialing phase, SIP messages dialed from an external terminal to an enterprise terminal are processed in HRAS by SER at first, and then processed

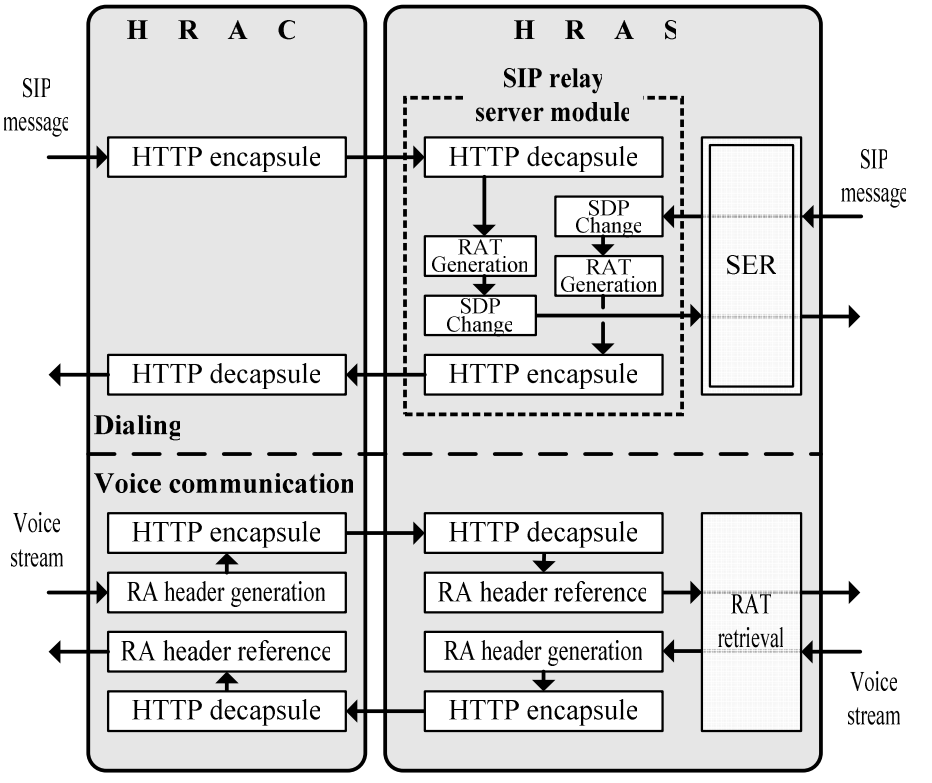


Fig. 6. Function and data flow

Table 2. Function of HRAC and HRAS

	Achievement method	Function
HRAS	HTTP tunnel	HTTP encapsule HTTP decapsule
	Voice stream guidance	SDP change
	Determination of a routing path of voice streams	RAT generation RAT retrieval RA header reference RA header generation
	Cooperation with SIP server	SER
HRAC	HTTP tunnel	HTTP encapsule HTTP decapsule
	Determination of a routing path of voice stream	RA header reference RA header generation

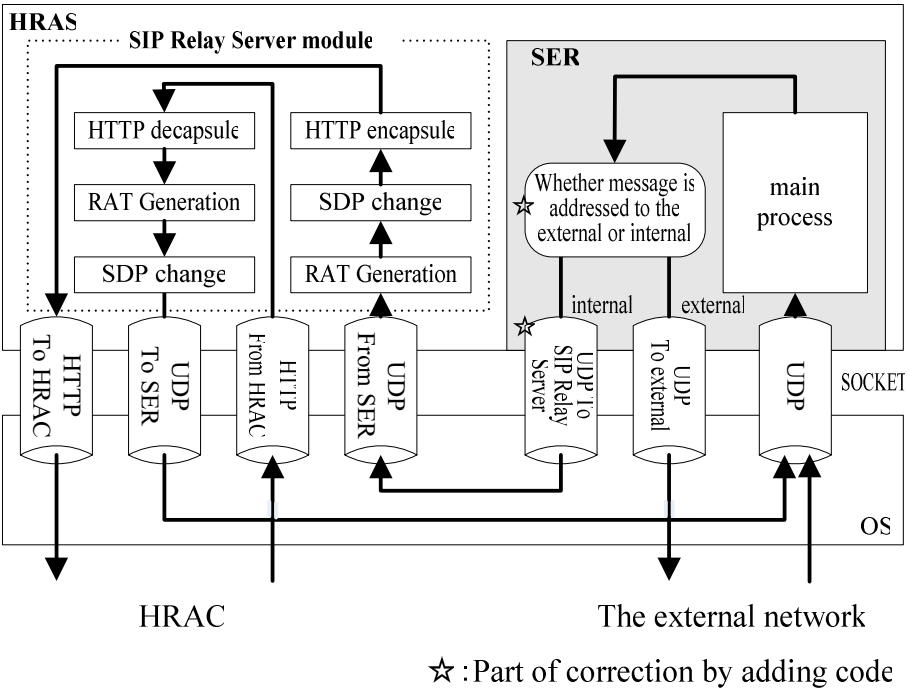


Fig. 7. The connection of SER and SIP relay server modules

by SIP Relay Server, in which SDP change, RAT generation, and HTTP encapsule, and relayed to HRAC. In HRAC, the SIP messages are transmitted to an enterprise terminal after processed by HTTP decapsule. SIP messages dialed from an enterprise terminal to an external terminal are relayed to HRAS after processed by HTTP encapsule in HRAC, and processed in HRAS by SIP Relay Server at first, and then SER, and transmitted to an external terminal. In voice communication phase, voice streams transmitted from the external terminal to the enterprise terminal are processed in HRAS in a order of RAT retrieval, RA header generation, and HTTP encapsule, and relayed to HRAC. In HRAC, voice streams are processed in a order of HTTP decapsule and RA header reference, and transmitted to the enterprise terminal. Voice streams transmitted from the enterprise terminal to the external terminal are processed in HRAC in a order of RA header generation and HTTP encapsule, and relayed to HRAS. In HRAS, voice streams are processed in a order of HTTP decapsule, RA header reference, and RAT retrieval, and transmitted to the external terminal. SOCKET is used for a connection between SIP Relay Server and SER, and its connection method is shown in Fig. 7. Fig. 7 shows the boundary portion between SER and SIP Relay Server in Fig. 6 in detail. SIP messages transmitted to HRAS from an external terminal reach SOCKET generated by SER, then after processed by SER, reach SOCKET generated by SIP Relay Server, and then relayed to HRAC. On the other hand, SIP messages relayed to HRAS through HRAC reach SOCKET generated by SIP Relay Server, and after processed by SIP Relay Server, they reach SOCKET

generated by SER. Then after processed by SER, they are transmitted to an external terminal. In case of replying a response like SIP registration message transmitted from an enterprise terminal, a reply message (200OK) is made by SER and relayed back to HRAC through SOCKET generated by SIP Relay Server again. To achieve the above-described flow, a simple modification is made to SER. As shown in Fig. 7, before SIP messages completing a series of processing by SER is about to leave for the SOCKET, a judgement function whether the message is addressed to an external terminal or to an enterprise terminal is added.

4 Conclusion

In this paper we have described the realization method of SoFW and its implementation. We are now implementing the system, and are going to evaluate delay of voice communications near future. This time, we have limited the application of SIP to voice communications, however, SIP have been paid many attentions to various kinds of applications. Studies for applications other than IP telephone should be considered hereafter.

References

1. N.Freed : Behavior of and Requirements for Internet Firewalls, IETF RFC 2979 (200.10)
2. H.323, Packe Based Multimedia Communications Systems, ITU-T Recommendation, 1998
3. Handley, M., Schulzrinne, H., Schooler, E. and Rosenberg, J.: SIP : Session Initiation Protocol, RFC2543(1999)
4. J. Rosenberg,et all “SIP: Session Initiation Protocol” IETF RFC3261(2002.6)
5. Petri Koskelainen, Henning Schulzrinnc, Xiaotao Wu : VoIP : A SIP-based conference control framework, ACM press 53-61 (2002.5)
6. Stcfan Bcrgcr, Honning Schulzrinnc, Stylianos Sidiroglou, Xiaotao Wu : Conferencing : Ubiquitous computing using SIP, ACM press 82-89(2003.6)
7. K. Egevang, P. Francis : The IP Network Address Translator (NAT), IETF RFC 1631 (1994.5)
8. Shinji Kunai.: Allround Internet Telephony Protocol -HTTP-based Conference Application Protocol ,IPSJ –JNK4403007
9. Skype.: “http://www.skype.com/home.html”
10. SoftEhter.: ” http://www.softether.com/jp/”
11. Handley, M. and Jacobson, V.: SDP : Session Description Protocol, IEETF RFC2327(1998)
12. SER : http://www.iptel.org/ser/

Voice Logging and Search Technology in IP Telephony Call Center

Kohta Ohshima[1], Eiji Muramatsu[1], Yasutaka Otake[1], Kimihiko Ando[1], Hiroki Ohno[2], and Matsuaki Terada[3]

[1] Graduate School of Technology, Tokyo University of Aguriculture and Technology
kohta@tela.cs.tuat.ac.jp
http://www.tela.cs.tuat.ac.jp/
[2] Intellectual Property Department Legal Affairs Division, Toppan Forms CO.,LTD.
[3] Institute of Simbiotic Science and Technology,
Tokyo University of Agriculture and Technology
m-tera@cc.tuat.ac.jp

Abstract. The Computer Telephony Integration system using VoIP has various usefullness. In particular, the demand for preservation and practical use of real-time voice information for telephone applications, is high in recent years. In the present paper, we investigate the voice logging system using a mid-scale IP phone call center having approximately 20 seats. The proposed system is characterized by the following four points: (1) reduction of packet loss and the impact thereof, (2) speech recognition technology to generate text information, (3) technology for dividing voice streams into paragraphs, and (4) the ability to treat voice and text seamlessly. The proposed system is evaluated by developing a prototype system that is equipped with the above features. Based on the results, a high recognition rate and telephone call preservation was achieved for 20 seats.

1 Introduction

Voice over IP (VoIP) is a technology that enables audio communication over an IP network. VoIP technology has grown in recent years, to the extent that it has replaced the Public Switched Telephone Network (PSTN). VoIP has the advantages of inexpensive equipment and low communication fees compared to the PSTN.

The Computer Telephony Integration (CTI) system using VoIP is effective in terms of both extendibility and cost, and effective use of saved information is important for Customer Relationship Management (CRM) applications, data mining, and perpetuation of evidence. In particular, preservation and practical use of real-time voice information, such as in telephone applications, are in high demand[2].

The present study investigates the voice logging system of a mid-scale IP phone call center having approximately 20 seats. Approximately 60% of Japanese call centers are mid-scale call centers having 50 seats or less [2]. Voice information is saved for use by the call center, and the saved information is not used in real time.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 892–901, 2006.

The proposed logging system is intended to treat voice information and text information bi-directionally in a seamless manner. Text conversion is performed on the saved voice information using speech recognition technology, and the converted text is associated with time information, making it possible to search the text of the voice information and playback the desired portion of the recorded voice information.

Developing a voice logging system with a speech recognition function requires examination of the system configuration and the recognition rate. The system configuration differs depending on whether a logging system is applied to the call center. The call center examined in the present study is assumed to use Session Initiation Protocol (SIP)[1] for the signaling protocol. Furthermore, the speech recognition rate is low for an unspecified speaker. Call centers deal with unspecified speakers, and so the search technology must be able to search for the desired information from an incorrectly recognized text. The proposed system is characterized by the following four points: (1) reduction of packet loss and the impact thereof, (2) speech recognition technology to generate text information, (3) technology for dividing voice streams into paragraphs, and (4) the ability to treat voice and text seamlessly. Since the purpose of the present study is not an examination of speech recognition engines, rather than improving the speech recognition engine, the hit ratio is increased only by using the recognition results.

2 Saving Voice Streams

2.1 Items for Consideration

Voice packet preservation involves reducing or corresponding a packet loss. Generally, in VoIP traffic, the signaling information and the voice stream are divided. Information concerning the media (target IP address and port number, kind of codec, etc.) is included in the signaling information. When a signaling packet is lost, media information cannot be acquired and differentiating voice streams may be impossible.

Saving a stream completely requires consideration of (1) reducing packet loss and (2) corresponding a packet loss.

2.2 System Structure

Figure 1 illustrates the target IP telephony call center. All of the telephones at the call center are connected to an IP network, and telephone calls that connect to external telephones are saved at the call center.

Based on differences in the logging node, the system configuration can consider the following three methods:

- Relay server method
- Sniffing method
- Client logging method

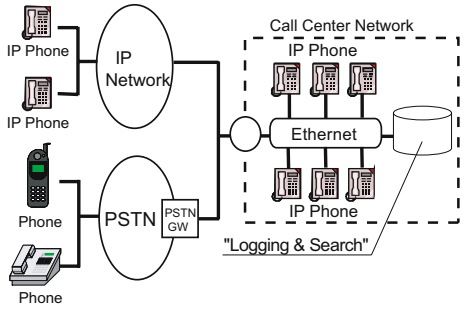


Fig. 1. IP telephony call center

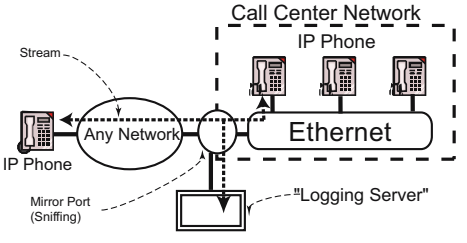


Fig. 2. Sniffing method

The relay server method always sends and saves packets via a server, such as the media gateway. This method has the risk of negatively affecting voice QoS due to delay delivery because of the load on the media gateway.

The sniffing method performs logging of the packet via a logging server and saves the copied data using the port mirroring function of the network apparatus (a switch and a router) (Fig. 2). Although an apparatus with a port mirroring function is required for this method, this is not a problem because the network most apparatuses used by trunk-line data service networks are equipped with a port mirroring function. In addition, permeability, whereby the user is not aware of the logging, is realizable. However, the load on the logging server becomes a bottleneck, and there is a risk that the packet that is received by the client is not receivable by the logging server

The client logging method logs the voice data at the client, and therefore no specific apparatus becomes a bottleneck. However, since a client equipped with the preservation function is needed, the cost in applying this method to a call center is high.

The relay server method has low voice QoS, and, since there is a bottleneck, the number of clients that can be handled is limited. Although the sniffing method has the possibility of packet loss, this method is easy to introduce and voice QoS is high. Although the client logging method does not have a bottleneck and voice QoS is high, the cost of installation at a call center is high because a special client is required.

We selected the sniffing method for the system configuration. The present study has as a prerequisite applicability at a call center. Therefore, it is necessary to consider ease of installation. The relay server method was not selected because a media gateway is required, and although the client logging method has various advantages, it was not selected because the cost of a special client is high.

2.3 Logging Processing

Both real-time processing flow and non-real-time processing flow can be considered in the sniffing method.

Real-time processing flow saves required packets alternately. The received packet is analyzed to determine whether it is a VoIP packet and is saved if this is the case. This flow can minimize the required amount of storage. However, the packet loss rate becomes high due to the load of distinction processing when receiving packets. Moreover, when a signaling packet is lost, there is the risk that a voice packet will not be saved.

Non-real-time processing flow temporarily saves all of the packets received by the logging server and distinguishes and acquires a required packet at off-peak times, for example, after the end of call center operation. In this flow, saved data cannot be used in real time. However, since all of the packets are saved, when a signaling packet is lost, it is possible to retrieve voice packets from the saved packets. Moreover, the packet loss rate by load becomes lower than real-time processing flow because the received packet is saved without performing distinction processing.

Although real-time processing flow has an advantage in that the time until data can be used and the required storage quantity are small, scalability is small and the impact of packet loss is great. Although non-real-time processing flow requires longer until preserved data can be used, its scalability is large and the impact of packet loss can be minimized. We chose non-real-time processing flow in consideration of the impact of packet loss.

2.4 Correspondence to Packet Loss

When a signaling packet is lost, the information in a voice packet cannot be obtained. Primarily, four kinds of information are included in the signaling packet: (a) Caller/Callee ID, (b) Call-ID, (c) Voice Codec, and (d) target port number. Here, (a) is the telephone number rather than the IP address, (b) is a unique ID that is designated for each telephone call. Calls having the same ID will be regarded as the same telephone call. Moreover, (c) is the voice codec of a voice packet, and (d) is the destination port number of a voice packet. This information is used when fetching a voice packet from a packet saved by non-real-time processing flow. Therefore, when a signaling packet is lost, it is necessary to guess this information in order to fetch a voice packet.

The following three features of a voice stream are used in fetching a voice packet without a signaling packet:

- All of the voice packets are transmitted to the destination specified by the signaling packet.

– Each voice packet are transmitted by a fixed cycle (ex. 20ms).
– The header of the transmission protocol, such as RTP[3], is attached.

When the packet group that fills these three conditions exists in the saved packet group, it can be concluded that the packet that fills these conditions is a voice packet. Here, (a) and (b) are not acquirable from a voice packet. Therefore, an IP address is used as (a) and a suitable value is generated as (b). In addition, (C) can be obtained from the transmission protocol header, and (D) can be obtained from a UDP header.

3 Reference of a Voice Stream

Reference of preserved information is performed via retrieval by keyword using the text generated using speech recognition technology. The voice stream and the text are associated by a time-axis.

3.1 Items for Consideration

Since telephone calls may target numerous unspecified persons in the call center, it is necessary to use a speech recognition engine that can handle an unspecified speaker. The speech recognition engine for an unspecified speaker has a low recognition rate compared with the speech recognition engine for a specified speaker. In addition, since the sound quality of the voice in telephone communications is poor, the recognition rate is further reduced. Therefore, improved voice quality and the development of technology that can enable low recognition rate searching are needed.

3.2 Speech Recognition Engine

Speech recognition may be either speaker-dependent or speaker-independent. Speaker-dependent speech recognition uses the wave pattern of the user, which is registered beforehand. Speaker-independent speech recognition is a method that compares a voice model and linguistic knowledge that models the statistical information on the features of the input voice and outputs the candidate that is most similar to the input voice as a recognition result.

We selected Julius[4], which is a speaker-independent speech recognition engine, because wave pattern registration of a user is impossible. Julius is a free open-source recognition engine that has a recognition rate of 90% or above with 20,000 vocabulary entries.

3.3 Association of Voice and Text

A technique for associating a voice stream and a recognition text on a time-axis is proposed. Since it is difficult to associate a voice stream and a text on a word-per-word basis, the voice stream and the text are associated by paragraph. The voice stream is divided by paragraph beforehand, and association is achieved by

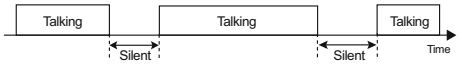


Fig. 3. Paragraph division by RTP Silence Suppression

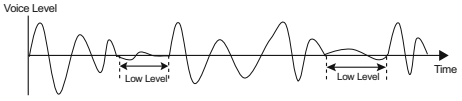


Fig. 4. Paragraph division by low voice level section detection

performing recognition processing by paragraph. An increase in the recognition rate can be expected by performing speech recognition by paragraph.

Two methods are proposed as paragraph division techniques. The first divides paragraphs using RTP Silence Suppression, where sections that are quieter than a certain threshold mark new paragraphs (Fig. 3). RTP Silence Suppression is a control that does not send voice packets when the input level from the microphone is very small. This method cannot be used when RTP Silence Suppression is not performed. The second method divides the voice stream into sections that exceed the threshold and low voice level sections as paragraphs (Fig. 4).

3.4 Search Technique

In this article, searching is performed using an ambiguous retrieval dictionary so that the hit ratio can be increased, even when incorrect recognition results are obtained. In speech recognition processing, the recognition rate does not necessarily become 100% and the correctness must be judged by the user. However, it is not realistic for a human operator to distinguish a large number of recognition texts. Therefore, an ambiguous retrieval dictionary is drawn up using the incorrect-recognized results and words that may have been recognized by mistake.

An ambiguous retrieval dictionary is created using that used by recognition processing. First, a voice stream is input using a microphone, and a corresponding word is input manually and set as an index word. Next, a speech recognition engine generates a recognition text from the input voice stream, and associates the generated text and the index. Then, the input voice stream is adjusted (level adjustment, noise grant, tempo adjustment), and a recognition text is again generated and associates with an index. Similarly, association with audio adjustment, and recognition and indexing are repeated, and the word list of the dictionary is increased.

Figure 5 illustrates the search flow using an ambiguous retrieval dictionary. First, the user inputs a keyword into the search interface and transmits the input keyword to a search engine. Next, the search engine receives a word list considered to be incorrect recognition results of the keyword retrieved from an ambiguous retrieval dictionary. Then, the database in which the voice stream

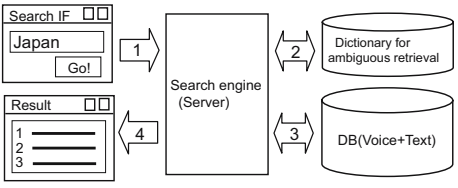


Fig. 5. Search using an ambiguous retrieval dictionary

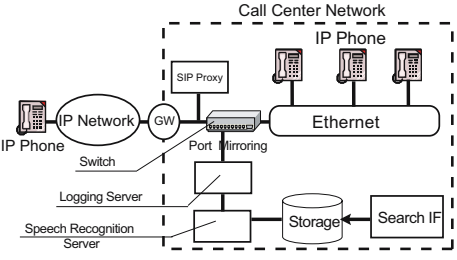


Fig. 6. Implemented system

and the recognition text were saved is searched using the word list. Finally, the results that match the correct recognition results are assigned a high score and displayed.

4 Implementation

Figure 6 illustrates the implemented system. All of the packets to an IP phone are copied by a switch having a port mirroring function and are saved by a logging server. After the saved data performs selection and retrieval of a voice packet, and after paragraph division is performed, recognized is performed by a speech recognition server. The recognition result text is associated with the voice stream and is saved at storage. The logging server (3GHz Pentium4 processor) and a speech recognition server (1.2GHz Pentium3 processor) are implemented on PC-UNIX, and developed by gcc. Storage is implemented on PC-UNIX, and PostgreSQL and Apache are run on the server.

Figure 7 shows the search interface. The search can be conducted using general browser software, and the interface is displayed by CGI[6]. Retrieval by keyword, Callee/Caller search, and date search are possible.

5 Performance Evaluation

5.1 Packet Loss Rate

The packet loss rate was measured by measuring the number of 64-byte packets received after transmission. Evaluation was performed by comparing real-time

Fig. 7. Search interface

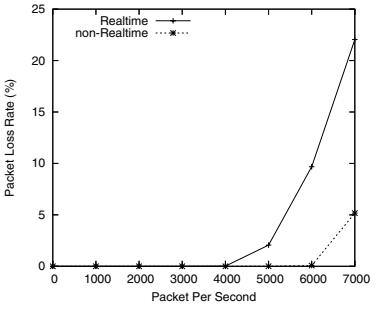


Fig. 8. Real-time and non-real-time packet loss rate comparison

processing flow with non-real-time processing flow. Transmitting packets in real time reduces the number of packets required for the selection target, including signaling packets.

The relationship between the number of packets received in one second (pps) and the packet loss rate are shown in Fig. 8. For the real-time processing flow, the loss rate was above 5000 pps, and at a rate of 7,000 pps, 23% of the packets was lossed. For non-real-time processing flow, the loss rate was above 6,000 pps, and at a rate of 7000 pps, 5.2% of the packets was lossed. The loss rate for a non-real-time processing flow is smaller compared to real-time processing flow because packets are saved without the distinction.

Figure 9 shows the CPU load factor at the time of logging processing. After the CPU load reaches 80%, the packet loss rate begins to rise. Therefore, in order to lower the packet loss rate, it is necessary to reduce the CPU load.

When the CPU load threshold is set to 80% of the packets to be saved, the real-time processing flow is set to 4000 pps and non-real-time processing flow is set to 5000 pps. Since approximately 60 voice packets are exchanged in one second, approximately 40 telephone calls can be saved by a non-real-time processing flow without packet loss.

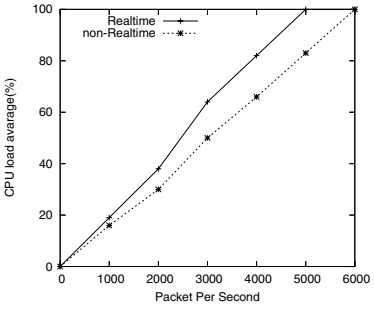


Fig. 9. CPU load factor comparison between real-time processing and non-real-time processing

Table 1. Comparison of speech recognition rates

	Sample Text #1	Sample Text #2
Paragraph #1	61.5%	87.0%
Paragraph #2	28.6%	81.3%
Paragraph #3	83.3%	70.0%
Paragraph #4	72.2%	75.0%
Recognize all paragraphs	66.1%	72.2%

5.2 Rate of Speech Recognition

The rate of speech recognition was compared before and after division of the input voice stream into paragraphs.

Table 1 compares the recognition rates with respect to paragraph division for two sample texts made up of four paragraphs. The recognition rates for individual paragraphs appear to be higher, in general, than for the non-divided rates. Although the recognition rate varies with the text with respect to the vocabulary of the speech recognition engine, the recognition rate appears to be higher when the text is divided into paragraphs.

6 Conclusions

In the present study, we proposed a voice logging system for an IP telephony call center. Methods by which to reduce the number of lost voice packets, to respond at the time of packet loss, in order to refer to preserved information, and to divide the voice information into paragraphs, and to generate text information from saved voice information by speech recognition technology, and to treat voice and text seamlessly are proposed. In addition, a system by which to implement the proposed method was developed. Measurement evaluation of the developed system revealed the performance of the system to be adequate for use in a call center having approximately 20 seats.

Acknowledgements

This work was supported in part by MEXT, KAKENHI (No. 17560330).

References

1. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler: SIP: Session Initiation Protocol. IETF, RFC3261 (2002)
2. Computer Telephony Magazine: CallCenter REPORT2003 in Japan. RIC TELECOM (2003)
3. H. Schulzrinne, S. Casner, R. Frederic, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications. IETF, RFC1889 (1996)
4. Julius - an Open-Source Large Vocabulary CSR Engine, http://julius.sourceforge.jp/
5. K. Singh, X. Wu, J. Lennox, H. Schulzrinne: Comprehensive Multi-platform Collaboration. Multimedia Computing and Networking, San Jose, California, USA (2004)
6. D. Robinson, L. Coar: The common gateway interface (CGI) version 1.1. Internet Draft draft-coar-cgi-v11-04.txt, IETF (2003)
7. H. Sinnreich, A. Johnston: Internet Communications Using SIP. John Whiley & Sons, Inc., New York (2001)
8. W. Jiang, J. Lennox, S. Narayanan, H. Schulzrinne, K. Singh, X. Wu: Integrating Internet telephony services. IEEE Internet Computing 6 (2002) 64–72
9. B. Milner, S. Semnani: Robust Speech Recognition over IP Networks. Proc. ICASSP 2000, Istanbul, Turkey (2000)
10. T. Salonidis, V. Digalakis: Robust Speech Recognition for Multiple Topological Scenarios of the GSM Mobile Phone System. Proc. ICASSP 98, Washington (1998)
11. F. Kubala, S. Colbath, D. Liu, A. Srivastava, J. Makhoul: Integrated technologies for indexing spoken language. Communications of the ACM (2000) 43–48

Integration of Ontologies and Semantic Annotations with Resource Description Framework in Eclipse-Based Platforms with Editing Features for Semantic Web

Rui G. Pereira and Mário M. Freire

Department of Informatics, University of Beira Interior,
Rua Marquês d'Ávila e Bolama,
6201-001 Covilhã, Portugal
{rpereira, mario}@di.ubi.pt

Abstract. Over the last three years, the number of developed Semantic Web tools has fastly increased. This huge arrival of new tools reveals the acceptance and credibility of the Semantic Web architecture. Nevertheless, currently available Semantic Web tools are limited to work only with few layers of the Semantic Web architecture. Therefore, there is a need for a new generation of Semantic Web tools integrating the functionality of several layers, in order to be easily used and tested by every web programmer/user. In this paper, we propose the architecture and describe the development of a new Semantic Web tool, named Integrated Editor of Semantic Annotations, Ontologies and Resource Description Framework for the Semantic Web (SWedt). This tool integrates the majority of standardized Semantic Web technologies and allows the easy creation of Semantic Web documents by web programmers/users.

1 Introduction

By Tim Berners-Lee *et.al.* [1], the Web is defined as a simple and excellent international platform of interconnection and transferring of documents between heterogeneous computer systems. However, is it extremely inefficient where the goal is the interconnection or the transferring, in an easy and autonomous way, of the concepts defined inside the documents. Another limitation is the fact that the existing information in the Web is being developed and prepared seeing exclusively its understanding by human beings. This way, we can summarize that the root of the main limitations in the web lays on its documents, more specifically in the inexistence of a universal structure and definition according to the kind of inside informations. Therefore, the way nowadays the information is defined and structured in the web may be considered as chaotic. The necessity of development of a new web architecture arises from the conscientiousness of these limitations. This new architecture will turn the web into a mean of collaboration by excellence, among people or between simple computer programs or another kind of non-human entity. By Tim Berners-Lee [2] and Marja-Riitta

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 902–911, 2006.

Koivunen *et.al.* [3], the materialization of this architecture depends on the use of a semantic that attributes a well-defined meaning to the available information in the present Web. One of the key-conditions to the success of this architecture is that the information classification can become universal, independently from the existing races, cultures, languages and communities. To this new revolution of Web functioning, Tim Berners-Lee has called Semantic Web.

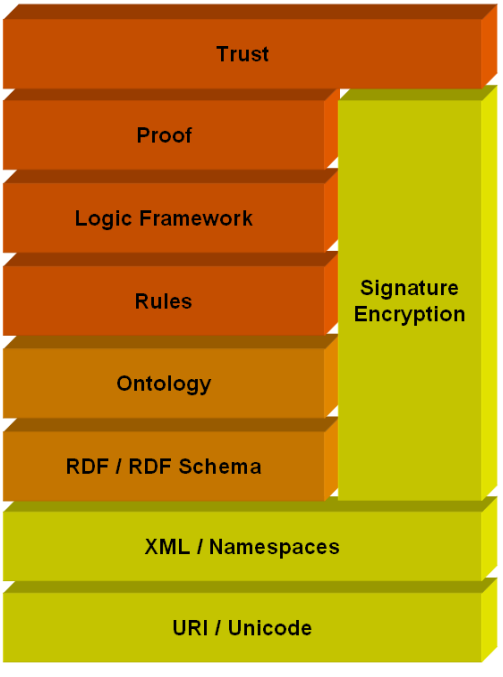


Fig. 1. The Semantic Web stack (adapted from [4])

In the present proposal of the developing web semantic presented by Tim Berners-Lee and by the consortium W3C [5] it is suggested a stratified architecture of technologies, as shown in figure 1. Nevertheless, the technologies of the Web semantic architecture, like the Uniform Resource Identifier (URI[1]) [6], the Extensible Mark-up Language (XML) [7], the Model and the Syntax of Resource Description Framework (RDF) [8], the RDF Schema[2] (RDFS) [9], the Ontologies, and so on, are presented in various layers, it doesn't mean that they are isolated entities. On the contrary, they are so interconnected that sometimes it is difficult to analyze them separately. Technologies are presented in a stratified architecture because, in general, technologies of the lower layers define the basis to the upper layers technologies. Thus, as we go ascending through the layers

[1] Also known as Universal Resource Identifier.
[2] RDF vocabulary description language.

of the architecture, technologies are representing the information in such a way, more and more expressive and rich in significance.

Despite being based on previous information retrieval and knowledge representation projects, Semantic Web goes far beyond them. It presents a way to define, add, extract and contextualize real-world concepts without ambiguity for several areas of knowledge in a non-centralized system. Machines will manage and interpret information in the same way as humans do, carrying them to our communication level. Its ability to interpret reality is still clearly very primitive. Nevertheless, in Semantic Web they will easily interact, learn and evolve. [1, 10]

Semantic Web is neither fiction nor reality; it is shown as a useful possibility and challenge. Semantic Web will integrate, interact and bring benefits to all human activities. Its full potential will profit beyond the Web, to real-world machines, providing increased interaction between machine to machine and machine to human, like phones, radios and other electronic devices. Semantic Web will be another step in a human-like form of the machines approach to our reality and in the evolution of the human knowledge. [10]

2 Architecture of the SWedt

Over the last three years, the number of developed Semantic Web tools had a very fast grow. Nevertheless, currently available Semantic Web tools are limited to work only with few levels of the Semantic Web architecture, just like RDF Editors [11, 12] or Ontologies Editors [13, 14]. In fact, the development of integrated tools for several levels of the Semantic Web architecture, in a way to be easily used and tested by every web user, is still not a reality. Here, we present the development of a new Semantic Web tool, named SWedt that integrates main standardized Semantic Web technologies and allows the easy creation of Semantic Web documents by web users.

The idea of creating a Semantic Web Editor when Semantic Web architecture, implementation and functionalities are still under development is justified by the urgent need to establish a link between Semantic Web developers/researches and the actual Web users. Thus, one of the main objectives in the SWedt development is to draw attention of current Web users to the Semantic Web. This interest is vital to the viability of Semantic Web itself, because it is known that even a very promising technology will not become a reality without the acceptance of its end-users. A new technology will only be accepted if it is useful and powerful and of extremely simple to use. SWedt tool is expected to reach this objective allowing the creation of Semantic Web pages to those users who are unfamiliar with these concepts.

The developed tool comprises and integrates an extremely diverse set of technologies, many of them still under development. Two of the main advantages in using tools previously available with licences free of utilization is due, by one hand, to the needless of constantly reinventing the wheel and, by other hand, with the stoutness in terms of functioning that usually it present. From among

all used tools, the a Eclipse Platform [15, 16] and the API Jena [17] were the ones that have contributed more to the SWedt development.

The need of information exchange from different kind of tools is increasing every day and will prevail in the Semantic Web. The majority of the current tools, besides that presenting different form of use interaction, they constantly need to change (import and export) information among them. By the other hand, the Eclipse Platform [15, 16], implemented in open code, is compatible with most operation system[3], can be easily extended, can allow the use of any language[4] and its main goal is the interconnection of different tools through the use of an universal and pleasant IDE. In this platform, tools are developed in the form of Plug-ins. Plug-ins are structured code packages that can be coupled to an application/platform in particular, and this way contributing to the addiction of new functionalities. Besides, Plug-ins can define Extension Points, which are well-defined localization in a Plug-in code that may be extended to other Plug-ins.

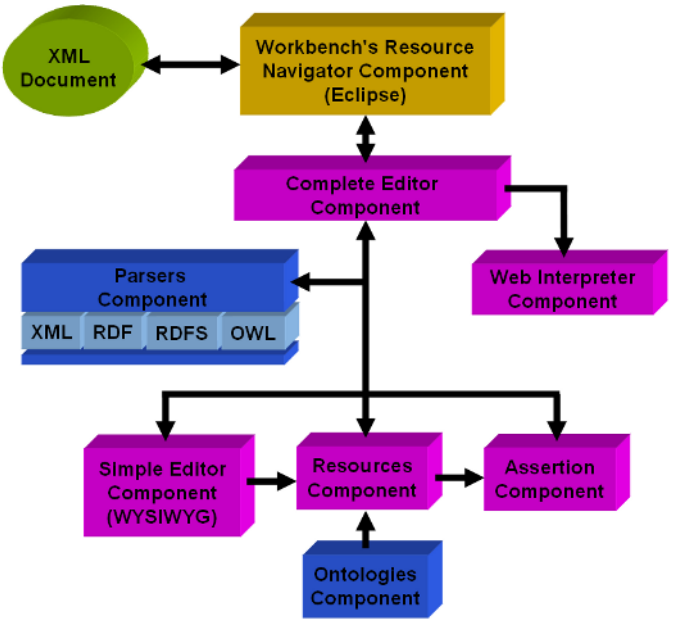


Fig. 2. Architecture of SWedt tool

In the figure 2 it is presented, through the unidirectional and bi-directional arrows, the existing interactions between the main components of the SWedt tool architecture, represented by the eight colored blocks. The oval shaped green

[3] Microsoft Windows, Linux, and so on.

[4] Example of languages that can be used: HTML, Java, C, JSP, *Enterprise JavaBeans* (EJB), XML, etc.

block placed in the upper left part of the image represents information in XML format (XML Document), the orange block placed in the upper part of the image represents a Eclipse Platform component (more exactly the Document Manager), the remaining seven violet and blue blocks existing in the image represent components that were developed from its root to the SWedt tool. The violet blocks represent tool components and the user has a practical perception of its existence. That is he can interact with them. The blue blocks represent tool components and the user doesn't have practical perception of its existence.

The XML format files are the raw material to the SWedt tool. The XML syntax is known by the tool and well as the RDF, RDFS e OWL [18] syntaxes whenever that are inserted in XML format files.

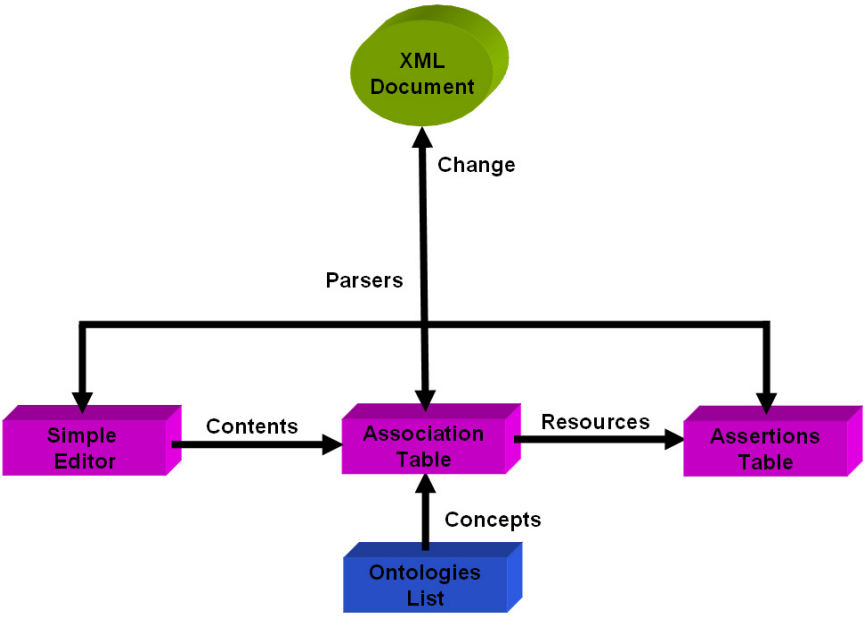


Fig. 3. Entities and Actions associated to development and implementation of SWedt tool

The SWedt tool development was characterized by the definition, and implementation in Java [19] language, of five main entities and by five main actions, that connect the entities between them. The defined and implemented entities are shown in figure 3 through the use of the five colored blocks. The main actions are presented by the arrow subtitles that define the existing relations between the main entities. The violet blocks represent entities that are directly related with the interaction done by the user. The remaining green and blue blocks represent entities that are not directly related to the interaction done by the users.

The SWedt tool uses the API Jena to represent internally through RDF Model any XML Document that the user may make available to the tool. This RDF Model of a XML Document represented internally by the SWedt tool is also called XML Document entity. The XML Document entity possesses a double objective. Firstly, to allow internally a virtual representation of a XML file and, therefore, to facilitate the existing interaction between the internal entities of the SWedt tool and the existing data in the XML file. Secondly, to allow an easy manipulation of the existing data in the XML file through the use of an RDF Model that is available in memory. The creation and management of the XML Document entity is the unique responsibility of the SWedt tool and this entity is associated to three other entities of the tool: the Simple Editor Entity, Association Table Entity and Assertion Table Entity. The existing association between the XML Document and the three entities referred in the former sentence is bi-directional and presents two kinds of actions: Change Action and Parsers Action. The Change Action is bi-directional and allows the establishment of an actualization between the content existing on the XML Document entity and the three entities referred in the previous paragraph. Whenever the XML Document entity is changed, the SWedt tool will automatically update the content of the other three entities referred through the Change Action so it can establish the actuality and integrity of the content presented in the various tool components. The opposite is true, whenever the content of the three referred entity is changed, the SWedt tool will automatically update the content of the XML Document entity through the Change Action. The Parsers Action is also bi-directional. Using API Jena, it allows to interpret and to manipulate the RDF Model of the XML Document entity. This action is carried out whenever the SWedt tool needs to update the XML Document entity or to fulfil the data contents existing in the following entities:

- Simple Editor Entity: The data content presented in this entity is just composed by the textual content existing on the RDF Model. Therefore, all content referring to the proper instructions of the XML, RDF, RDFS e OWL syntaxes existing in the RDF Model is rejected by the Parsers Action actuation;
- Association Table Entity: The data content presented in this entity is just composed by RDF resources defined in the RDF Model and by all words existing in the Simple Editor Entity. Therefore, all remaining content existing in the RDF Model is rejected by the Assertion-Table Entity; and
- Assertion-Table Entity: The data content presented in this entity is just composed by the RDF Resources and Assertions existing in RDF Model. Therefore, the remaining content of RDF Model is rejected by the Parsers Action actuation.

The SWedt tool makes available an editor that allows the textual content manipulation existing in a XML file by the user. This editor is the nucleus of the Simple Editor Entity. The main objective of this entity is to allow the simple access and edition of the textual content existing in a XML file by the SWedt file user. This entity allows the manipulation of an XML file textual

content through the invocation of Parsers action about the XML Document entity. So, through that action and then through the API Jena use, the entity accede to the RDF Model of the tool and extract all textual content of the XML file. Presenting the textual content of a XML file by an editor is a SWedt tool responsibility. However, the manipulation of that content is a responsibility of the editor of the user. Thus, whenever the SWedt tool user changes the content of the Simple Editor entity through the Change action, he automatically updates the content of the XML Document entity. Besides these two actions, Parsers and Change, this entity also interacts with a unidirectional action called Contents. The last action is always carried out when the content belonging to the Simple Editor entity is changed. Its function is to provide a list of all words existing in the Simple Editor entity to the Association Table entity. The SWedt tool makes a table available with the declaration of all RDF Resources existing in a XML file. That table is the nucleus of the Association Table entity. It is called Association Table entity because it has a three-column table associated. Each line of that table is composed by a RDF Resource declaration and each RDF Resource declaration is composed by an association between a word belonging to the Simple Editor Entity content (first column of the table) and a URI[5] (second column of the table). The third column makes a mechanism of graphic selection available that enables the validation of associations existing between those words and these URIs by the SWedt tool users. The main objective of this entity is to allow the definition of RDF Resources through the manipulation of a simple table associated to the XML Document made by not-familiarized users with the RDF Syntax and Model. This entity fulfils the table through the following three actions:

- Parsers Action: This action allows the presentation and manipulation of RDF Resources existing in a XML file through the invocation of the Parsers Action to the XML Document entity. So, through that action and then the use of API Jena, the entity extracts all RDF Resources declared in the internal RDF Model of the tool and inserts them in the Association Table;
- Contents Action: This action provides a list of all words existing in the Simple Editor Entity to the Association Table Entity. The Association Table Entity extracts all words of that list and inserts them in the first column of its table, corresponding each word to a line of its table; and
- Concepts Action: This action creates a list of all word existing in the former table that don't have a URI associated, that is, the second column on the table is blanked. URI are acquired by concepts identification, defined in Ontologies, that are equal or similar to the words existing in the first column of the table. That similitude is verified syntactically and in the form of domain. The main function of this action is to look for concepts, existing in Ontologies; equal or similar to the words existing in the Association Table in order to create associations between words and URIs automatically. The concepts defined in Ontologies are made available by the List of Ontologies.

[5] Concept defined in an Ontology.

Besides the invocation of these three actions, whenever the table is changed, either automatically by the tool either by the user, this entity invokes two other actions: Change and Resources. The unidirectional action called Resources will whenever invoked make available to the Assertion Table a list of all Resources existing in the table of the Association Table entity.

The SWedt tool makes a list of Ontologies available that can be updated any moment. The entity responsible by the list management is called List of Ontology Entity. Its main function is to allow the search and extraction of concepts defined in a List of Ontology made by the Concepts Action. The activity of navigation through the concepts defined in Ontologies was implemented through the use of API Kazuki [20].The functioning of this entity presents a transparent behavior to SWedt tool users.

The SWedt tool lists the declaration of all RDF Assertions available that exists in a XML file into a table. That table is the nucleus of the Assertion Table Entity. The main objective of this entity is to allow the definition of RDF Assertions through the manipulation of a simple table associated to the XML Document made by not-familiarized users with the RDF Syntax and Model. This entity fulfil its table through the following two actions and by the user:

- Parsers Action: This entity allows the presentation and manipulation the RDF Assertions through the invocation of the Parsers Action about the XML Document entity. Therefore, through that action and then using API Jena, the entity extracts all RDF Assertions declared in the internal RDF Model of the tool and inserts them in the Assertion Table;
- Resources Action: This action provides a list of all RDF Resources existing in the Association Table Entity to the Assertion Table. The Assertion Table Entity extracts all RDF Resources existing in that list and inserts them in three combo-lists so that they can be selected by the users; and
- User Action: This entity provides a table that, whenever manipulated by the user, it can create or change RDF assertions. The tool makes only RDF Resources available to this table, and so the user will be creating logical relations between resources and RDF Assertions when he creates triple associations between RDF available.

3 Conclusion

The current Semantic Web research is mostly pointed towards development of specialized tools associated with few layers of the Semantic Web architecture. This tendency is clearly evident by the development of a variety of systems and software tools aiming to provide editors integrating technologies used in these specific layers. However, the true power of Semantic Web will only be felt when real benefit hits the users. In fact, there are still no tools that provide a stable support bridge between Semantic Web researchers and Web users. It is important to develop support tools that may help users to involve and to take interest in the Semantic Web. Development of these tools must observe current web-user interaction and introduce only minimal changes.

We have described a Semantic Web tool integrated in the Eclipse platform, called SWedt, that allows the easy creation of Web pages following the principles of the Semantic Web. The proposed semantic web tool integrates semantic annotations and Ontologies with RDF and can be used by programmers/users who are unfamiliar with semantic web architecture.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001). Retrieved from Scientific American.com web site: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&sc=I100322 (last access: April 28, 2005).
2. Berners-Lee, Tim. (Outubro 1998) *"Semantic Web Road map"*. W3C (MIT, ERCIM, Keio). Retrieved from W3C web site: http://www.w3.org/DesignIssues/Semantic.html (last access: June 10, 2005).
3. Koivunen, Marja-Riitta & Miller, Eric. (2001) *"W3C Semantic Web Activity"*, Semantic Web Kick-Off in Finland: Vision, Technologies, Research, and Applications, HIIT Publications, Helsinki Institute for Information Technology, Helsinki, Finland.
4. Semantic Web Architecture Image. Retrieved from W3C web site: http://www.w3.org/DesignIssues/diagrams/sw-stack-2002.png (last access: April 28, 2005).
5. W3C. Retrieved from W3C web site: http://www.w3.org/ (last access: April 28, 2005).
6. *Universal Resource Identifiers* (URI). Retrieved from W3C web site: http://www.w3.org/Addressing/URL/URI_Overview.html (last access: September 15, 2005).
7. XML 1.0 - W3C Recommendation. Retrieved from W3C web site: http://www.w3.org/TR/2004/REC-xml-20040204/ (last access: September 15, 2005).
8. *RDF/XML Syntax Specification (Revised)* - W3C Recommendation. Retrieved from W3C web site: http://www.w3.org/TR/rdf-syntax-grammar/ (last access: September 15, 2005).
9. *RDF vocabulary description language* (*RDF Schema*) (RDFS). Retrieved from W3C web site: http://www.w3.org/TR/rdf-schema/ (last access: June 11, 2005).
10. Pereira, R.G., Freire, M.M.: Semantic Web. In: Pagani, M. (ed.): Encyclopedia of Multimedia Technology and Networking. Idea Group, Inc., (2005), ISBN: 1-59140-561-0.
11. RDFedt tool. Retrieved from Jan Winkler web site: http://www.jan-winkler.de/dev/e_rdfe.htm (last access: April 28, 2005).
12. RDF Instance Creator (RIC) tool. Retrieved from Mindswap web site: http://www.mindswap.org/ mhgrove/RIC/RIC.shtml (last access: April 28, 2005).
13. Protégé tool. Retrieved from Protégé web site: http://protege.stanford.edu/ (last access: April 28, 2005).
14. OntoEdit tool. Retrieved from On-To-Knowledge web site: http://www.ontoknowledge.org/tools/ontoedit.shtml (last access: April 28, 2005).
15. Eclipse White Paper: Eclipse Platform Technical Overview. Eclipse Project (2003). Retrieved from Eclipse Project web site: http://www.eclipse.org (last access: April 28, 2005).

16. Eclipse Project Presentation. Retrieved from Eclipse Project web site: www.eclipse.org/eclipse/presentation/eclipse-slides.ppt (last access: April 28, 2005).
17. Jena2 - Semantic Web Toolkit. Retrieved from HP Labs Semantic Web Research web site: http://www.hpl.hp.com/semweb/ (last access: April 28, 2005).
18. *Web Ontology Language* (OWL). Retrieved from W3C web site: http://www.w3.org/TR/owl-ref/ (last access: June 11, 2005).
19. Linguagem de Programao Java. Retrieved from Sun web site: http://java.sun.com/docs/books/jls/ (last access: June 11, 2005).
20. Kazuki API. Retrieved from SemWebCentral web site: http://projects.semwebcentral.org/projects/kazuki/ (last access: April 28, 2005).

Analysis of Error Resilience in H.264 Video Using Slice Interleaving Technique

Amit Sood, Naveen K. Chilamkurti, and Ben Soh

Dept of Comp Sci and Comp Eng
La Trobe University
Bundoora, Melbourne, Australia-3086

Abstract. An important demand in video streaming industry today is the ability to code *error resilient* digital video and the *error concealment* techniques adopted by the video coding standards. The H.264/AVC is an emerging video standard being developed jointly by the Video *Coding Experts Group* (VCEG) and the *Moving Pictures Experts Group* (MPEG) as the *Joint Video Team* (JVT). This new video coding standard defines some very powerful features that make it promising for the next generation internet applications like video streaming. Apart from significant improvements in compression performance, this standard is also more 'network friendly' and provides more features for error resilience. In this paper, we study the error resilience techniques provided by the emerging H.264 video standard. We study the effects of *slice coding* on transmitted video and also analyze the concept of *Slice Groups*, which is an error resilience techniques adopted in the H.264 video coding standard.

Keywords: H.264, Slice interleaving, PSNR, Macro block.

1 Introduction

Video compression technology has seen significant advancements in the past two decades and a lot of research work has been done in this field. The reason for this increased significance of video compression and research based on it is because of two primary reasons. Compressed video is much smaller in size as compared to uncompressed video, thereby making it possible to store or transmit in environments that have size & bandwidth restrictions. Also, compressed video allows for a balance between channel/storage media capacity and quality. This means that we can adjust the quality of the transmitted video according to the kind of network it is being transmitted on to or the kind of storage medium it is being stored in to. Advances in network technology and video coding have resulted in such applications becoming more and more feasible and realistic by the day.

An important demand in video streaming industry today is the ability to code *error resilient* digital video and the *error concealment* techniques adopted by the corresponding decoder. This is because compressed video is highly vulnerable to errors. When we talk about streaming video over a network, this vulnerability increases further. The internet transport protocols used today are extremely reliable; however

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 912 – 921, 2006.

network congestion and competition for limited bandwidth resources in the network leads to errors in the transmitted video sequences.

Error resilience is process of designing the video compression algorithm and the encoded bit stream in such a manner that it is resilient to errors. Error concealment on the other hand, is a damage recovery mechanism, where the loss information is estimated in order to conceal the fact that an error has occurred. Retransmission is generally considered unacceptable for real time video applications because of the incurred delays.

One particular coding standard that is of interest to many researchers is the H.264/AVC video coding standard. The H.264/AVC is an emerging video standard being developed jointly by the Video *Coding Experts Group* (VCEG) and the *Moving Pictures Experts Group* (MPEG) as the *Joint Video Team* (JVT). The main goals of this team is to develop a video coding design which is simple, has a higher compression performance and, is also network adaptable and friendly. The new standard has been proven to achieve significant bit rate savings when compared to other existing video coding standards such as MPEG-2 video.

The encoder consists of two separate layers, namely, the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The VCL represents the coded video data in a video stream. The NAL on the other hand, provides formatting and header information so that it can be efficiently transmitted by various transport layer protocols or storage media. Each picture of a video can be sub-divided into macro blocks that are rectangular areas of 16 X 16 samples luma component and 8 X 8 samples of each of the two chroma components. The macro blocks can be further organized as slices, which are sub-groups within a given picture that can be independently decoded. The process of dividing pictures into slices is usually referred to as 'slice coding'.

The H.264 video coding standard employs various error resilience techniques. Slice coding is one of the techniques used in the H.264 video coding standard that provides for error resilience. Since each picture is subdivided into one or more slices and the slice is given increased importance in H.264 as the basic spatial segment that is independent from its neighbors. Thus, errors or missing data from one slice cannot propagate to any other slice within the picture. Hence if there is packet loss during transmission over the network, the decoder should be able to decode the received video with varying capacity.

The aim of our experiments is to measure this capacity of the decoder to decode the received (and possibly corrupted) video in terms of mean Peak Signal to Noise Ratio (PSNR) values. This is of significant importance to business applications based on real time video streaming like video conferencing and internet based video-on-demand, where packet losses introduced by bottleneck network links and network congestion can cause significant video quality performance degradation.

We analyze the concept of *Slice Groups* in the H.264 video coding standard. A slice group is subset of macro blocks in a coded picture which may consist of one or more slices. Within each slice group, macroblocks are encoded in raster order. Therefore if only one slice is coded per picture, all the macroblocks are arranged in raster order.

2 Slice Coding

The H.264 encoder consists of two separate layers, namely, the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The VCL represents the coded video data in a video stream, and does all the classical signal processing tasks. The NAL on the other hand, provides formatting and header information so that it can be efficiently transmitted by various transport layer protocols or stored in storage media.

A coded video sequence in H.264/AVC consists of a sequence of *coded pictures*. A coded picture can represent either an entire *frame* or a single *field*, as was also the case with MPEG-2 video. Each frame can be coded as a number of macroblocks that are rectangular areas of 16 X 16 samples luma component and 8 X 8 samples of each of the two chroma components. The macroblocks are further organized as slices, which are sub-groups within a given picture that can be independently decoded. These macroblocks per slice need not be constant within a picture. The process of dividing pictures into slices is usually referred to as 'slice coding'.

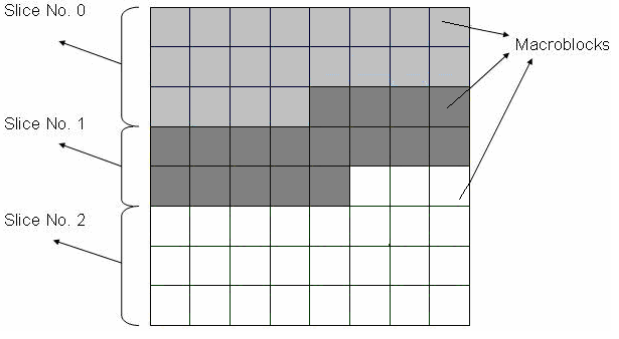


Fig. 1. A Picture Coded as Macroblocks & Slices

There are several different slice types. All three profiles support "I-slices" that contain only intra-predicted macroblocks, and "P-slices" that contain inter-predicted (motion-compensated) macroblocks. The Main and Extended profiles also support "B-slices," which implement inter-prediction from two reference frames.

The Extended profile adds "Switching I" (SI) slices and "Switching P" (SP) slices. These slices are used to facilitate features like random accesses and video stream switching, and are useful for streaming video applications. For example, SP slices can be used for seamless switching between video streams carrying the same video content but encoded at different bit rates. SI slices use only intra-frame prediction (not inter-frame prediction), and thus can be used for switching. There is minimal inter-dependency between these coded slices which can help to limit the propagation of errors.

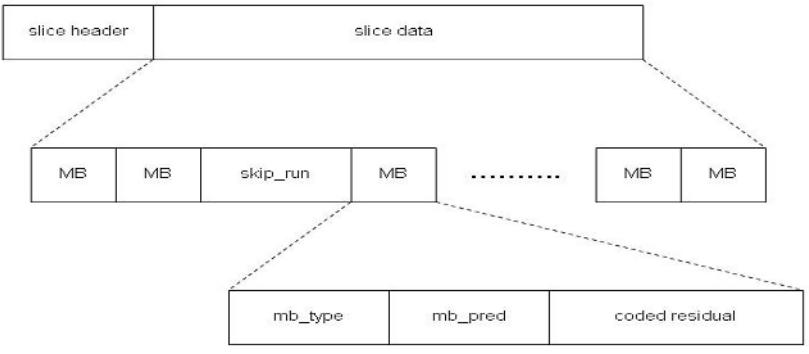


Fig. 2. Slice Syntax

The above figure shows a simplified view of a coded slice. The slice header defines the slice type and the coded picture or frame that the slice belongs to. The slice data consists of a series of coded macroblocks and/or an indication of skipped macroblocks. Each macro block on the other hand, contains a series of header elements and coded residual data.

Since each picture is subdivided into one or more slices and the slice is given increased importance in H.264 as the basic spatial segment that is independent from its neighbours, errors or missing data from one slice cannot propagate to any other slice within the picture. Simply put, it means that H.264 video coding standard does not allow for inter-slice dependency, and hence each slice can be decoded independently. Hence if there is packet loss during transmission over the network, the decoder should be able to decode the received video with varying capacity.

The aim of our experiments is to measure the capacity of the decoder to decode the received (and possibly corrupted) video in terms of mean Peak Signal to Noise Ratio (PSNR) values. This is of significant importance to commercial applications based on video streaming like video conferencing and internet based video-on-demand, where packet losses introduced by bottleneck network links and network congestion and noisy channels can cause significant video quality performance degradation.

3 Experimental Framework and Work-Flow

We analyze the situation where the number of macroblocks coded per slice is continually increased and the selected video sequence is encoded with new values. The experimental setup consists of a raw YUV CIF video sequence. The selected video sequence, 'football_cif.yuv' is encoded into an H.264 sequence using the 'The JM H.264/AVC Reference Software Encoder/Decoder'. A tool named 'tfgenout' was written in C to generate a trace file 'st' containing the frame number, frame type, frame size of each frame generated during the encoding process. This trace file acts as the input to the network simulator. A TCL script was then written to describe the

network topology that is relevant to the analysis. This script file reads the 'st' trace file, and generates another trace file called 'sd_be', containing the timestamp for UDP packet transmission, packet id and the number of bytes sent with each UDP packet sent. UDP has been chosen as the transport layer protocol for the transmission as UDP is most commonly and commercially used today with multimedia applications, such as internet phone, real-time video conferencing, and streaming of stored audio and video.

Fig. 3. Football_cif.Yuv video sequence

The two trace files 'st' and 'sd_be' together; represent the entire video transmission process, at the sender side. At, the receiver side, another trace file 'rd_be' is generated, that contains the timestamps for UDP packet reception, packet id and the number of bytes received or lost with each received or lost UDP packet. Another tool named 'et' which was written in C, then automates the process of reading the 'rd_be' file and generating the received video from the originally encoded H.264 video sequence. Once this (possibly erroneous) received video file is generated, it is fed into the H.264/AVC Reference Software Decoder, to generate the YUV CIF video sequence. Another thing to be noted here is that, with the baseline H.264 decoder, there is no need for bitstream modification. This (possibly erroneous) received video sequence is then compared with the original source YUV CIF video sequence to obtain PSNR values. The following diagram illustrates the PSNR calculation and evaluation process.

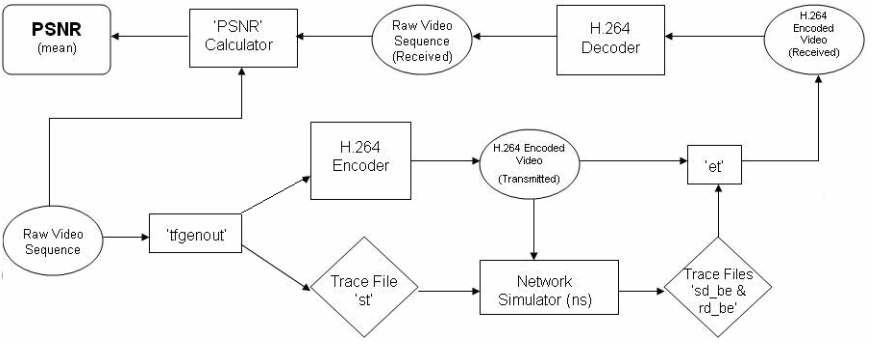


Fig. 4. Framework for PSNR calculation & video quality evaluation

The above process is repeated with several different values for the 'Number of Macroblocks coded per slice' and graphs are plotted to evaluate the result of our experiment.

4 Effect of Slice Coding

4.1 Topology 1: Heterogeneous Traffic Sources

The first topology that we have chosen for this experiment consists of three traffic sources, the primary video source (our transmitted video sequence) which is labeled UDP1, sends UDP packets from S1 to D1, another UDP traffic source labeled UDP2 that transmits UDP traffic from S2 to D2, and thirdly an FTP data source running on TCP, labeled TCP, that sends TCP packets from S3 to D3. All these traffic sources compete for bandwidth, especially in the bottle neck link R2→R3. A drop tail queue policy has been used at this link, due to which packets that arrive when the queue is full, are discarded as soon as they reach the node R2.

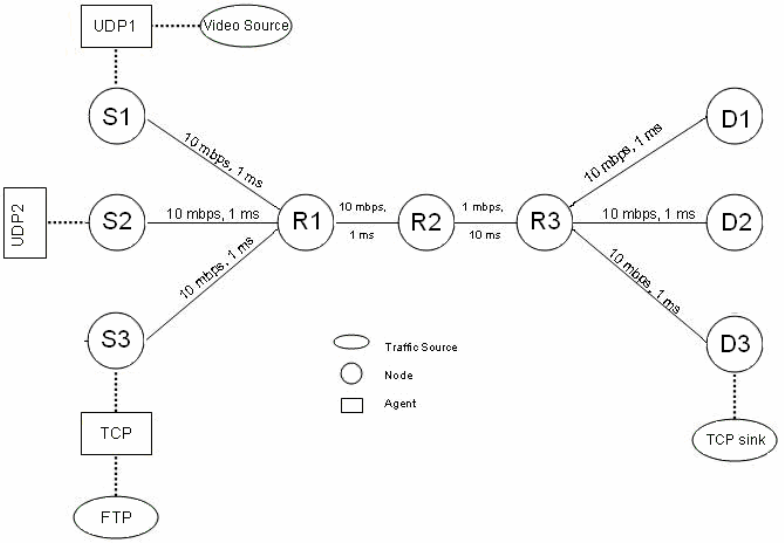


Fig. 5. Topology 1: Heterogeneous Traffic Sources

Our aim is to find out the video quality results of the chosen sequence when it is transmitted on such a network where different data sources are competing for bandwidth resources. This is a typical scenario if we consider a real world application like video conferencing over the internet, where the different networks over which the video data travels might suffer from congestion due to data from other traffic sources. We calculated PSNR values for this received video with 20 different values for the 'macroblocks per slice' per slice parameter, ranging from 5 macroblocks to 100 macroblocks per slice, and plotted these values on a graph. The results are shown below.

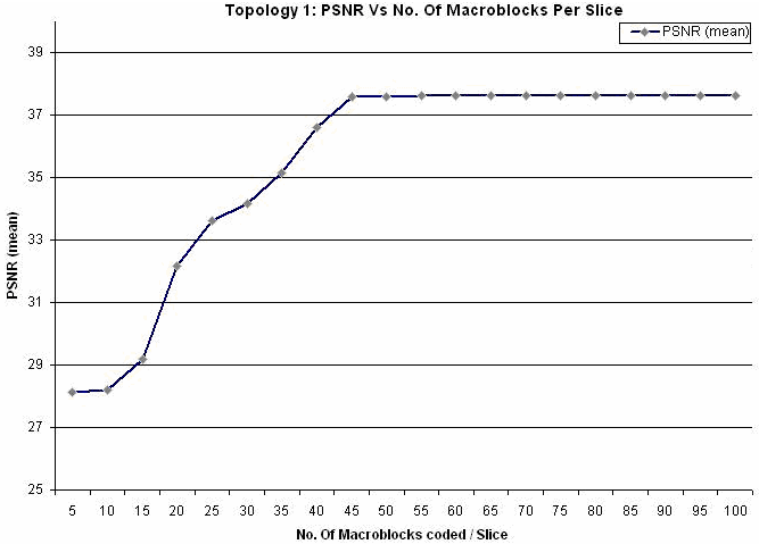


Fig. 6. Mean PSNR vs. Macroblocks per Slice Graph for Topology1

We see from the above graph that at 5 coded macroblocks per slice the calculated PSNR value is very low i.e. 28.14 dB. As we increase the number of slices coded, this PSNR value increases to 32.16 dB at 20 macroblocks, 36.6 dB at 40 macroblocks, and then 37.6 db at 50 macroblocks per slice. It then becomes stable at 37.64 dB and stays fixed at this value after that. This shows that increasing the number of coded macroblocks certainly increases the video quality, but only up to a certain limit, which in case of this network topology is 37.64 dB. This is also the best achieved quality parameter value with this specific set of conditions.

We also plotted the number of packets transmitted and lost against the number of coded macroblocks per slice. The following results were obtained.

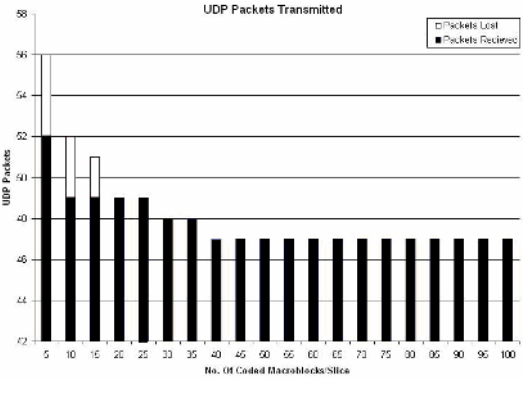


Fig. 7. Packets Transmitted, Received & Lost

From figure 6 we see that as the number of coded macroblocks per slice is increased, the number of transmitted packets decreases. This means that the increase in the number of macroblocks results in lesser number of coded slices per picture and hence the overall size of the transmitted video decreases, and therefore the number of UDP packets transmitted also decreases. Therefore the number of UDP packets transmitted is inversely proportional to the number of macroblocks coded per slice. Also the number of packets lost decreases, as the number of coded macroblocks per slice is increased, which results in better quality of the received video, as demonstrated by the previous graph.

4.2 Topology 2: Scalability

In this experiment a topology with a number of receivers, for the purpose of scalability. Here, our video source does not compete with any other traffic source for bandwidth resources. The UDP traffic source labeled UDP1 transmits UDP traffic from S1 to D1. However, there is a bottleneck link R1→R2 that can cause packet loss. A drop tail queue policy has been used at this link as well, due to which, packets that arrive when the queue is full, are discarded as soon as they reach the node R1. Since there is no competition for bandwidth resources, we expect that the received picture quality would be certainly better than the previous experiment.

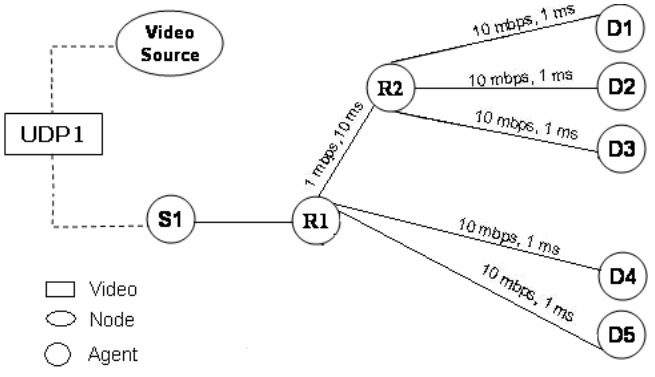


Fig. 8. Network topology for scalability

We calculated PSNR values for this received video with 20 different values for the 'macroblocks per slice' per slice parameter, ranging from 5 macroblocks to 100 macroblocks per slice, and plotted these values on a graph. The PSNR calculations for this topology are depicted, again in the form of the following graph.

From the fig. 9 again, we see again that as the number of coded macroblocks are increased per slice that is coded, the PSNR value increases. Although the achieved PSNR value is much higher with this topology, as compared to the previous one, the result achieved is the same. The higher PSNR values are because of the fact that no traffic sources compete for bandwidth resources in this setup. The PSNR increases from 37.57 dB at 5 coded macroblocks per slice to 37.62 dB at 20 macroblocks per slice, and the trend continues till the threshold value of 37.65 dB is reached,

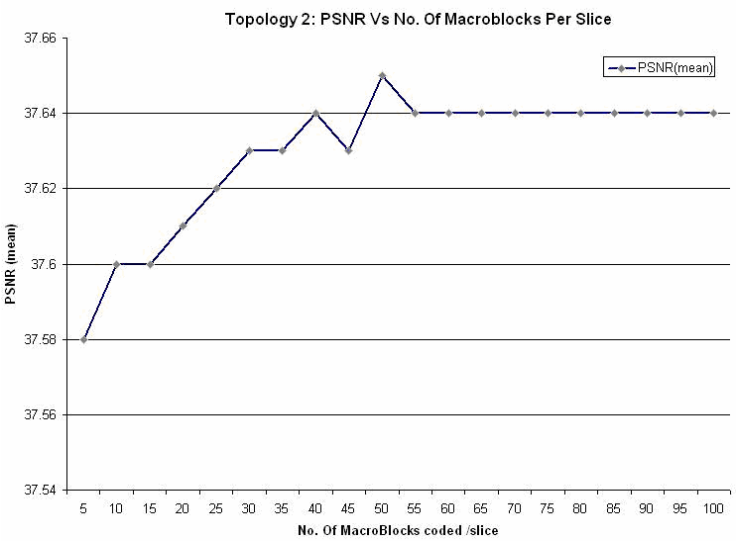


Fig. 9. Mean PSNR vs. Macroblocks per Slice Graph for Topology2

after which there is no further increase in the quality parameter value even when the number of coded macroblocks per slice are increased. The following figure depicts the UDP packets sent and lost during the simulations. Here we see, just like the previous topology, that as the number of macroblocks coded per slice is increased, the number of UDP packets transmitted decreases.

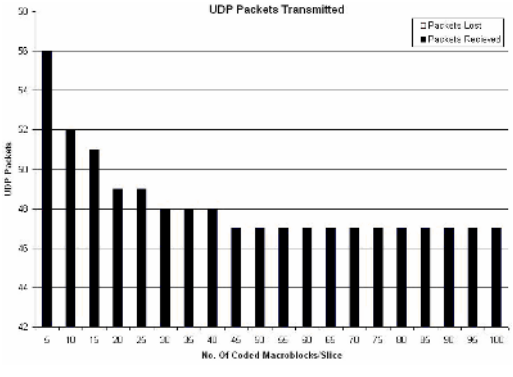


Fig. 10. Packets Transmitted, Received & Lost

5 Conclusions

In this paper, we study the error resilience techniques provided by the emerging H.264 video standard. Using two topologies we analyze the concept of *Slice Groups* in the H.264 video coding standard. From first topology, we can see that as the

number of coded macroblocks per slice is increased, the number of transmitted packets decreases. This means that the increase in the number of macroblocks results in lesser number of coded slices per picture and hence the overall size of the transmitted video decreases, and therefore the number of UDP packets transmitted also decreases. Therefore the number of UDP packets transmitted is inversely proportional to the number of macroblocks coded per slice. Also the number of packets lost decreases, as the number of coded macroblocks per slice is increased, which results in better quality of the received video.

From second topology, as the PSNR increases from 37.57 dB at 5 coded macroblocks per slice to 37.62 dB at 20 macroblocks per slice, and the trend continues till the threshold value of 37.65 dB is reached, after which there is no further increase in the quality parameter value even when the number of coded macroblocks per slice are increased. Here we see, just like the previous topology, that as the number of macroblocks coded per slice is increased, the number of UDP packets transmitted decreases.

References

1. Jinghong Zheng Lap-Pui Chau "Error-concealment algorithm for H.26L using first-order plane estimation", IEEE Transaction on Multimedia , Vol 6, Issues 6, Pg 801-805, Dec 2004.
2. Ralf Schäfer, Thomas Wiegand and Heiko Schwarz "Emerging H.264/AVC standard" Technical Report, Heinrich Hertz Institute, Berlin, Germany
3. Nejat Kamaci, Yucel Altunbasak "performance comparison of the emerging h.264 video coding standard with the existing standards", IEEE Int. Conf. Multimedia and Expo, Baltimore, MD, July 2003.
4. Stephan Wenger and Thomas Stockhammer, "An Overview on the H.26L NAL Concept", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 2nd Meeting: Geneva, CH, January 29 - Feb. 1, 2002.
5. Peter Borgwardt, "Rectangular Slices to Tradeoff Error Resiliency and Coding Efficiency", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 3rd Meeting: Fairfax, Virginia, USA, 6-10 May, 2002.
6. A. Tamhankar and K.R. Rao," Overview of H.264 / MPEG-4 Part10". 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, 2003.
7. Jijun Zhang, Andrew Perkis and Nicolas D.Georganas, "H.264/AVC and Transcoding for Multimedia Adaptation". In Proceedings of the 6th COST 276 WORKSHOP, Thessaloniki, Greece, 6-7 May 2004.

Peer-to-Peer and Overlay Networks

Performance Evaluation of QoS-Aware Routing in Overlay Network

Masato Uchida[1,2], Satoshi Kamei[1], and Ryoichi Kawahara[1]

[1] NTT Service Integration Laboratories
3–9–11, Midori-cho, Musashino-shi, Tokyo 180–8585, Japan
[2] Network Design Research Center, Kyushu Institute of Technology
3–8–1 Asano, Kokurakita-ku, Kitakyushu-shi, Fukuoka 801–0001, Japan
m.uchida@ndrc.kyutech.ac.jp,
{kamei.satoshi, kawahara.ryoichi}@lab.ntt.co.jp

Abstract. A recent trend in routing research is the use of overlay routing to improve end-to-end QoS without changing the network level architecture. The key of this technology is to find alternative routes, which can avoid congested routes, using an overlay network. Evaluating the calculation cost to find such an alternative route to develop the technology is important. Therefore, this paper evaluates how effective the technology can be when the number of alternative route candidates is limited. The evaluation results indicate that the technology is effective even if alternative route candidates are limited to one quarter of the number of routes.

1 Introduction

The Internet has developed to accommodate various applications. Recently, the accommodation of real-time applications, such as VoIP (Voice over IP) and streaming services, has progressed. These applications are sensitive to QoS (Quality of Service), so achieving techniques to avoid congestion and improve end-to-end performance has become important. However, there are several problems in achieving such techniques as follows.

- The Internet has already become a social infrastructure, so implementing new functions that significantly change the existing architecture of the physical network (IP network) is difficult.
- The Internet is composed of multiple ASs (Autonomous Systems) with different management organizations, so implementing new functions on ASs all at once is difficult.

Against this background, controlling traffic using an overlay network, which is a logical network constructed over the physical network, is attracting much attention because it enables us to improve end-to-end QoS without changing the physical network. Examples of previous studies on overlay networks to improve end-to-end QoS are SON (Service Overlay Network) [1], OverQoS [2], and QRON (QoS-aware Routing in Overlay Networks) [3].

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 925–934, 2006.

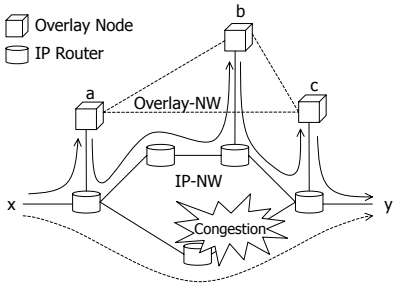


Fig. 1. Concept of routing control using overlay network

Fig. 2. Map of Japan

This paper focuses on routing control using an overlay network to improve end-to-end QoS. Its fundamental concept is illustrated in Fig. 1. In this figure, we assume that the traffic flows on the route from x to y are indicated by the dashed arrow. In addition, we assume that there is a congested router on the route; therefore, the QoS between x and y is degraded. Now, we can say that the congestion can be avoided using the alternative route transiting overlay node b, i.e., the route illustrated as the solid arrow: x→a→b→c→y, where the alternative route is established using the overlay network composed of overlay nodes a, b, and c.

In [4,5], there are reports that there are many alternative routes improving the end-to-end QoS based on the actual traffic data. This means that the distribution of traffic in the IP network is heterogeneous, so existing routing controls in the IP network are not optimal. These reports are important because they demonstrate the potential effectiveness of the routing control using an overlay network.

However, [4] evaluated the improvement of end-to-end QoS only when the optimal alternative route can be selected among all possible alternative route candidates. Therefore, the evaluation of the improvement of end-to-end QoS when the number of alternative route candidates is limited is insufficient. Such an evaluation is important for implementing the routing control using the overlay network because the cost of finding the optimal alternative route among candidates increases as the number of candidates increases.

On the other hand, [5] evaluated the improvement of end-to-end QoS when the number of alternative route candidates has been limited. However, in this evaluation, alternative route candidates were randomly selected and the influence of the selection method on the improvement of QoS was not discussed.

First, we demonstrate that transit nodes offering optimal alternative routes are distributed non uniformly using actual traffic data. That is, a small number of transit nodes can offer optimal alternative routes at a high ratio. Then, we evaluate the improvement of end-to-end QoS when we limit the number of alternative route candidates based on the above-mentioned non uniformity.

Finally, we demonstrate that we can improve end-to-end QoS in the same manner as the case of selecting the optimal alternative route even if we limit the number of alternative route candidates. These evaluations are important for making deployment and selection of transit nodes more efficient.

2 Analysis and Evaluation of Traffic Data Among ISPs

2.1 Traffic Data Among ISPs

The data used in this paper was measured between 18 hosts (overlay nodes), each connected directly to one of 18 geographically separated Japanese ISPs. We measured the delay time between each of two different hosts by executing one ping command per second for three minutes, i.e., 180 packets per hour. This measurement was performed for two weeks. Six nodes were set up in Tokyo, four nodes were set up in Osaka, four nodes were set up in Sapporo, and four nodes were set up in Kumamoto, as shown in Fig. 2. We analyzed the maximum delay time within each three-minute measurement.

We denoted the set of the above 18 nodes as $V = \{v_1, v_2, \ldots, v_{18}\}$. In addition, we denoted the maximum delay time from v_i to v_j $(i \neq j)$ at the n-th hour $(n = 1, 2, \ldots)$ as ping_max(n, v_i, v_j) and the route from v_i to v_j as $v_i \rightarrow v_j$.

2.2 Non Optimality of Default Route

In this section, we evaluate the validity of the default route from v_i to v_j to show the potential effectiveness of the routing control using an overlay network.

First, we evaluate the ratio where there are alternative routes that have better end-to-end QoS than the default route. That is, we evaluate the ratio using traffic data of each day where there is at least one v_k $(k \neq i, j)$ satisfying ping_max$(n, v_i, v_j) >$ ping_max(n, v_i, v_k) + ping_max(n, v_k, v_j) for each combination of i and j $(i \neq j)$. This ratio corresponds to f_0 defined in section 2.3. The ratio of each measurement day is shown in Fig. 3. The ratio is about 0.4 and this result does not depend on the measurement day.

Then, we evaluate how much end-to-end QoS improves using the optimal route. Here, let us define the optimal route from v_i to v_j as follows. If there is at least one v_k $(k \neq i, j)$ satisfying ping_max$(n, v_i, v_j) >$ ping_max(n, v_i, v_k) + ping_max(n, v_k, v_j), then denote v_k minimizing the right hand side of this equation as $v_{n,i,j}$ and define the optimal route as $v_i \rightarrow v_{n,i,j} \rightarrow v_j$. Otherwise, define the optimal route as $v_i \rightarrow v_j$.

The cumulative distributions of maximum delay times of the default route and the optimal route using traffic data of one day for all combinations of i and j $(i \neq j)$ are illustrated in Fig. 4. In addition, the correspondence of maximum delay times between default and optimal routes for all combinations of i and j $(i \neq j)$ is shown in Fig. 5. As shown in these figures, we can see that the maximum delay time of the optimal route is much shorter than that of the default route. Note that this result did not depend on the measurement day.

The result of this section indicates that the routing control in an IP network is not necessarily optimal from the viewpoint of maximum delay time.

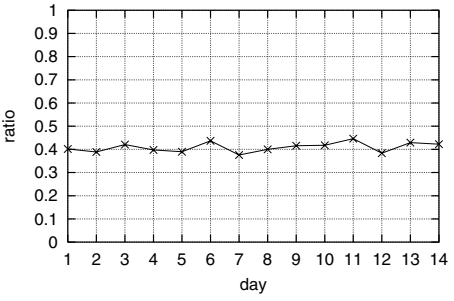


Fig. 3. Ratio where alternative routes have better end-to-end QoS than default route

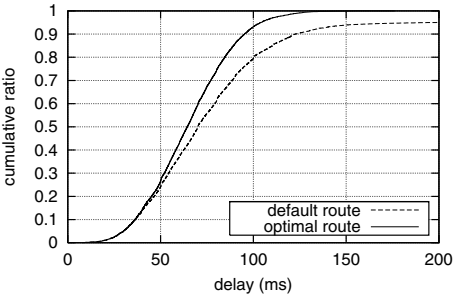


Fig. 4. Cumulative distribution of maximum delay time

2.3 Non Uniformity of Optimal Transit Node

The results of section 2.2 indicate the potential effectiveness of routing control using an overlay network. Here, the evaluation method used in section 2.2 was presumed to select an alternative route that has a better QoS than the default route, where all nodes, excluding the source node and the destination node, are regarded as transit node candidates. Therefore, the evaluation method does not take the cost of selecting the alternative route into consideration. This point becomes an important problem when the number of nodes that constitute the overlay network increases.

Now, we consider limiting the number of transit node candidates to solve the problem. That is, we reduce the cost of selecting the alternative route by regarding only some nodes as transit node candidates. However, there is a possibility that a desirable alternative route, which has a better end-to-end QoS than the default route, cannot be used when we select transit node candidates at random. Therefore, we need to decide which nodes should be transit node candidates.

In this section, we evaluate which node has been selected as the transit node when the optimal route is not the default route. Based on this evaluation, we show that the selection of the transit node, i.e., $v_{n,i,j}$, is strongly biased.

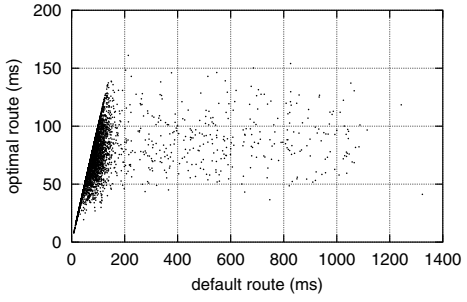


Fig. 5. Correspondence of maximum delay times between default and optimal routes

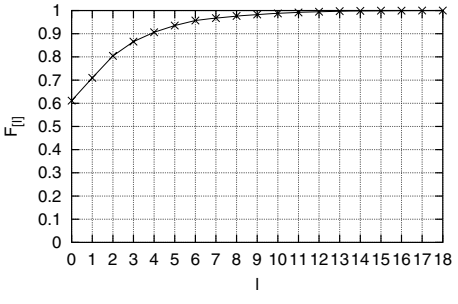


Fig. 6. $F_{[l]}$ vs. l

When v_k $(k \neq i, j)$ does not satisfy ping_max(n, v_i, v_j) > ping_max(n, v_i, v_k) + ping_max(n, v_k, v_j), we set $v_{n,i,j} = v_0$. Let us define the ratio f_k such that v_k $(k = 0, 1, 2, \cdots, 18)$ is selected as the optimal transit node, where

$$f_k = \sum_{\substack{n=1,2,\cdots,N \\ i,j=1,2,\cdots,18,\ i\neq j}} \frac{I(v_{n,i,j} = v_k)}{N \times 18 \times 17}, \quad (k \neq i, j)$$

and

$$I(v_{n,i,j} = v_k) = \begin{cases} 1 & \text{if } v_{n,i,j} = v_k \\ 0 & \text{if } v_{n,i,j} \neq v_k \end{cases}, \quad (k \neq i, j).$$

In addition, let us define the arrangement of f_k in descending order of value by $f_{[l]}$ $(l = 0, 1, 2, \cdots, 18)$. Then, the cumulative value of $f_{[l]}$ is defined by

$$F_{[l]} = \sum_{x=0}^{l} f_{[x]},$$

where $F_{[18]} = 1$ holds.

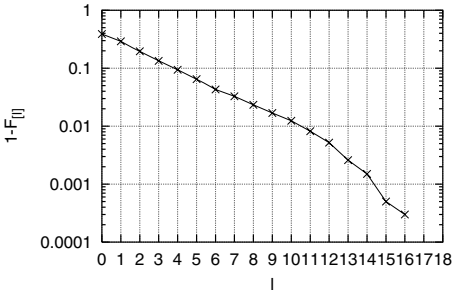


Fig. 7. $1 - F_{[l]}$ vs. l

Table 1. Relationship between $f_{[l]}$ and the geographic position of corresponding node

Position	Tokyo	Osaka	Sapporo	Kumamoto
l	1, 2, 3, 5, 9 and 10	4, 6, 7 and 14	8, 11, 13 and 17	12, 15, 16 and 18

The value of $F_{[l]}$ calculated from the same data as that of Fig. 4, i.e., $N = 24$, is shown in Fig. 6. We find that $F_{[0]}$ is about 0.6. In this case, we find that $F_{[0]} = f_{[0]} = f_0$ holds (see Fig. 3). In addition, we find that $F_{[4]}$ is about 0.9. This indicates that about 90% of the optimal routes can be covered even when the transit node candidates are limited to the top 4 nodes with respect to the value of f_k. This percentage includes the case when the optimal route is the default route. In other words, even when the transit node candidates are limited to the top 1/4 (= 4/16) nodes (note that the number of nodes excluding the source node and the destination node is 16), 3/4 (= $(0.9 - 0.6)/(1.0 - 0.6)$) of the optimal routes, excluding the case where the default route is the optimal route, can be covered. On the other hand, the improvement in the percentage of the optimal routes that we can cover is only 10% if we add the 12 $(16 - 4)$ lower nodes with respect to the value of f_k to the transit node candidates.

Now, we discuss the above result from another viewpoint. The value of $1 - F_{[l]}$ calculated from the same data as that of Fig. 6 is shown in Fig. 7, where the vertical axis is a logarithmic scale. Here, we can say that $F_{[l]}$ is distributed exponentially because the shape of this figure is a straight line. This means that the value of $f_{[l]}$ decreases exponentially as the value of l increases. In other words, a small number of nodes with a high ratio are selected as transit nodes, while the other nodes with a low ratio are selected.

On the other hand, there is an interesting relationship between $f_{[l]}$ and its corresponding geographic position. The relationship is shown in Table 1. The value of f_k increases in order of Kumamoto, Sapporo, Osaka, and Tokyo. Although this result seems to reflect the structure of the Internet in Japan, it also seems to depend on the ISP of the overlay nodes, so detailed observation of the above relationship is for future study.

Note that the characteristics shown in this section do not depend on the measurement date of the analyzed data, though this section only evaluated the data measured within one day.

2.4 Impact of Transit Node Limitation

The discussion of sections 2.2 and 2.3 demonstrates (i) the non optimality of the default route and (ii) the non uniformity of the optimal transit node based on traffic data measured among ISPs. Forty percent of optimal routes are not default routes but alternative routes, and 3/4 of such alternative routes are composed of the top 4 transit nodes with respect to the value of f_k. These results indicate that the end-to-end QoS can be improved sufficiently even when transit node candidates are limited to the above-mentioned four nodes. This section investigates that in detail. For convenience, let us denote the above-mentioned four nodes as v_1, v_2, v_3, and v_4. In the following, we use the same data as that of sections 2.2 and 2.3.

The procedure to select routes when the transit node candidates are limited to v_1, v_2, v_3, and v_4 is given as follows. First, if v_k $(k \neq i, j,\ k = 1, 2, 3, 4)$ satisfies $\mathrm{ping\_max}(n, v_i, v_j) > \mathrm{ping\_max}(n, v_i, v_k) + \mathrm{ping\_max}(n, v_k, v_j)$, let $v'_{n,i,j}$ denote v_k, which minimizes the right hand side of the equation, and $v_i \rightarrow v'_{n,i,j} \rightarrow v_j$ be the route of this case. Then, if there is no such v_k, let the route of this case be $v_i \rightarrow v_j$.

When the selected route is $v_i \rightarrow v_j$, we can categorize it into the following two cases. That is, (i) there is no such v_k satisfying $\mathrm{ping\_max}(n, v_i, v_j) > \mathrm{ping\_max}(n, v_i, v_k) + \mathrm{ping\_max}(n, v_k, v_j)$ for $k = 1, 2, \ldots, 18$ and (ii) there is no such v_k for $k = 1, 2, 3, 4$ but there is for $k = 5, 6, \ldots, 18$. The selected route is $v_i \rightarrow v_j$ for both cases. We set $v'_{n,i,j} = v_0$ for case (i) and $v'_{n,i,j} = v_{19}$ for case (ii). Here, $v'_{n,i,j} = v_0$ means that there are no alternative routes that have better end-to-end QoS than the default route, regardless of the presence of the limitation in the number of transit node candidates. In contrast, $v'_{n,i,j} = v_{19}$ means that there are alternative routes that have better end-to-end QoS than the default route, but the alternative routes cannot be used by limiting the number of transit node candidates, and as a result, the default route is selected. Note that the selected route using the above-mentioned route selection procedure is not necessarily the optimal route. This point is different from the procedure used in sections 2.2 and 2.3 because if $v_{n,i,j} = v_k$ $(k \neq 0, 1, 2, 3, 4)$, then $v_{n,i,j} \neq v'_{n,i,j}$, while if $v_{n,i,j} = v_k$ $(k = 0, 1, 2, 3, 4)$ then $v_{n,i,j} = v'_{n,i,j}$. Therefore, we can say that the selected route when the number of transit node candidates is limited is a sub optimal route.

As shown above, we cannot use the optimal route using $v_5, v_6, \ldots, v_{18}$ as transit nodes when the transit node candidates are limited to v_1, v_2, v_3, and v_4. To evaluate the impact of the limitation, let us define the ratio of selecting v_k as the sub optimal transit node as

$$f'_k = \sum_{\substack{n=1,2,\cdots,N \\ i,j=1,2,\cdots,18,\ i\neq j}} \frac{I(v'_{n,i,j} = v_k)}{N \times 18 \times 17}, \quad (k \neq i, j,\ k = 0, 1, 2, 3, 4, 19).$$

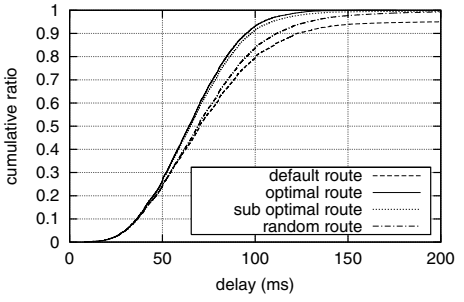


Fig. 8. Cumulative distribution of maximum delay time (superimposed on Fig. 4)

The value of f'_{19} is important for measuring the impact of the limitation in the number of transit nodes. This is because the value of f'_{19} indicates the ratio of alternative routes that have better end-to-end QoS than the default route, but the alternative route cannot be used when the number of transit node candidates is limited. As a result, the default route is selected. Using the same data used in sections 2.2 and 2.3, we see that the value of f'_{19} is about 0.05. This value seems to be sufficiently small, so we expect that the effect of limiting the number of transit node candidates is negligible. In the following, we confirm the conjecture.

The cumulative distributions of the maximum delay time of the default route, optimal route (these two lines are the same as those in Fig. 4), and sub optimal routes that are selected using the above-mentioned procedure are illustrated in Fig. 8. For reference, we also show the results when the transit node candidates are selected at random (random route), where there are four transit node candidates. As shown in this figure, using the sub optimal route, we can obtain almost the same performance as that of the optimal route. In addition, the performance of the random routes is much worse than those of the optimal route and the sub optimal route.

Finally, we look at the above result in detail. The maximum delay time is degraded when the number of transit node candidates is limited if the optimal route from v_i to v_j is an alternative route using $v_5, v_6, \ldots, v_{18}$ as transit nodes as illustrated in Fig. 9. Specifically, for the pair of i and j satisfying $v_{n,i,j} = v_k$ ($k = 5, 6, \cdots, 18,\ k \neq i, j$), the x-coordinate and y-coordinate of each point in this figure is $\mathrm{ping\_max}(n, v_i, v_j) - (\mathrm{ping\_max}(n, v_i, v_{n,i,j}) + \mathrm{ping\_max}(n, v_{n,i,j}, v_j))$ and $\mathrm{ping\_max}(n, v_i, v_j) - (\mathrm{ping\_max}(n, v_i, v'_{n,i,j}) + \mathrm{ping\_max}(n, v'_{n,i,j}, v_j))$, respectively. That is, the x-coordinate and y-coordinate of each point indicate the reduction in maximum delay time without and with limiting the number of transit node candidates, respectively. Here, we set the y-coordinate to 0 when $v'_{n,i,j} = v_{19}$. As shown in this figure, many points are on a straight line with a slope of 1 and a y-intercept of 0. This indicates that a similar performance to that of the optimal route can be achieved by using a sub optimal route. This result also indicates that the number of transit node candidates used in this evaluation is appropriately limited.

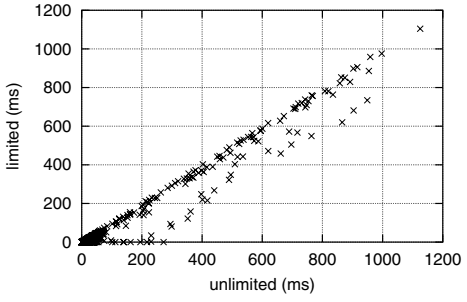


Fig. 9. Comparison of reduction in maximum delay time (horizontal axis: without limitation, vertical axis: with limitation)

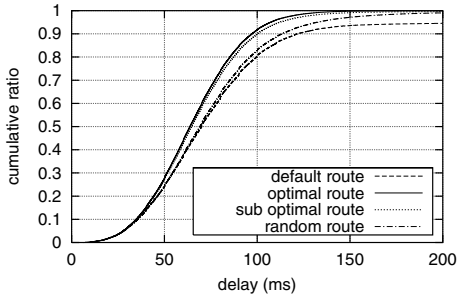


Fig. 10. Cumulative distribution of maximum delay time (case of taking impact of daily variation into consideration)

2.5 Impact of Daily Variation

The analysis of the previous section uses the same data for selecting a transit node and calculating end-to-end QoS using the selected transit node. However, we consider it is natural that appropriate transit node candidates depend on the daily variation. Therefore, this section gives an evaluation taking that into consideration.

The result of a similar evaluation using data taken for 2 weeks, including the day whose data is used in the previous sections, is illustrated in Fig. 10. Here, transit node candidates are selected using the data of the first measurement day. Comparing Figs. 8 and 10, we find that the characteristics of both figures are almost the same. This means that the impact of daily variation of the transit node candidates can be disregarded.

3 Conclusion

This paper gives the following evaluations about the effectiveness of routing control using an overlay network.

- Non optimality of default route:
  We evaluated the end-to-end QoS when the optimal route can be selected. As a result, we found that the end-to-end QoS of the optimal route is much better than that of the default route.
- Non uniformity of optimal transit node:
  We discussed the limitation in the number of transit node candidates. This discussion is important for reducing the cost to select alternative routes that have better end-to-end QoS than that of the default route. As a result, we found that the selection of the optimal transit node is strongly biased.
- Impact of transit node limitation:
  We evaluated the improvement in end-to-end QoS when the number of transit node candidates is limited. As a result, we found that the performance of the sub optimal route is almost the same as that of the optimal route if we select the candidates appropriately.
- Impact of daily variation:
  We evaluated how the daily variation of transit nodes influences the improvement in end-to-end QoS. As a result, we found that the improvement does not depend on the daily variation.

The above results are useful for increasing the efficiency of the deployment of overlay nodes and selection of routes that have better performance than that of the default route. However, further study is needed to justify the evaluation results given in this paper.

References

1. Y. T. Hou, Z. Duan, and Z. Zhang, "Service overlay networks: SLA, QoS and bandwidth provisioning," Proc. IEEE ICNP'02, November 2002.
2. L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: Offering QoS using Overlays," Proc. HotNets-I, October 2002.
3. L. Zhi and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," IEEE J. Select. Areas Commun., vol. 22, pp. 29-40, January 2004.
4. S. Banerjee, T. G. Griffin, and M. Pias, "The Interdomain Connectivity of PlanetLab Nodes", Proc. PAM 2004, April 2004.
5. S. Rewaskar and J. Kaur, "Testing the Scalability of Overlay Routing Infrastructures," Proc. PAM 2004, April 2004.

Path-Aware Multicast for Efficient File Distribution in Peer-to-Peer Overlay Networks

Chun-Hsin Wu, Jia-Wei Li, Yueh-Ju Chen, and Jum-Ping Lin

Department of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan
wuch@nuk.edu.tw, {jwli, yjchen, jplin}@syslab.csie.nuk.edu.tw

Abstract. This paper discusses how to transmit the same file or the same part of a large file from a source peer to a group of destination peers efficiently. Considering the quality of overlay links and the restriction of Internet firewalls, we propose a hybrid approach of amplification and multicast, called amplicast, to improve the efficiency of file distribution. By intelligent peer selection and the support of replicate peers, the proposed amplicast approach constructs a series of multicast sub-trees to alleviate the load of the source peer and allow a requesting peer to decide when and where to join a tree, based on its own benefits. With link status cache and top-set sampling heuristic, the proposed PeerTop can significantly reduce the cost of exhausted network probing. Experiments with real-world Internet traffic data demonstrate that our approach can distribute contents very efficiently.

1 Introduction

Transmitting files from a source peer to a group of destination peers is a very fundamental operation in overlay networks for content distribution. For example, in a peer-to-peer (P2P) file-swapping system, a peer may simultaneously receive multiple requests from other peers for the same file, so it expects to distribute the requested part to all the requesting peers as fast as possible with the least effort. In a content-push application, a source peer also often needs to replicate the same content to a specified group of peers. For example: a user may want to upload a new program to a cluster of machines, a content provider needs to replicate contents to a set of mirror sites, or a company needs to duplicate data to a couple of backup sites. All of these applications require an efficient content distribution system.

When a peer receives requests for the same content from many other peers, the simplest scheme implemented by many existing P2P systems, called root-serve, is to serve all the requesting peers successively by itself. To improve download performance, a large file may be split into small pieces, which can be downloaded concurrently from different supplying peers, and a requesting peer may issue parallel requests to multiple supplying peers simultaneously for the same piece of content.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 935–944, 2006.

Another common approach to content distribution is to incorporate the peers that have just received a copy of the content to help serve other pending peers [8]. This is often called the amplification approach. As a peer receives the requested content, it can become a supplying peer of the content at the next round. When a requesting peer is served and ready to serve others, the original supplying peer notifies some pending requesting peers that a new supplying peer with a replicate of the content is available, or redirects some pending requests to the new supplying peer. Compared to the root-serve approach, the amplification approach would reduce the load of the source peer and the waiting time of the requesting peers significantly. During the amplification process, however, most requesting peers still need to wait several rounds before being served, especially those peers that are scheduled for the last round.

In this paper, we adapt a multicast approach for file distribution. We focus on the problem of how to transmit the same file or the same part of a large file efficiently from a source peer to a group of destination peers. Although multicast mechanisms have been widely proposed to transmit data packets or multimedia streams simultaneously to a group of nodes, little work has been done to optimize multicast mechanisms for file distribution. We aim to reduce the longest waiting time for the source peer to serve the last requesting peer, and the total waiting time to serve all the requesting peers. Our studies show that splitting a single multicast tree into multiple multicast sub-trees for file distribution, sometimes performs much better than without such splitting.

In addition, the proposed multicast approach to file distribution takes the dynamics of overlay networks into concern. Qualities of overlay link paths between pairs of peers are probed and collected by PeerTop. Peer nodes behind firewalls or NAT are identified during multicast tree construction. The proposed amplicast algorithm utilizes the path information among requesting peers to plan and construct highly-efficient file-transmission multicast trees.

This paper is organized as follows. In Section 2, we present the basic ideas of the proposed amplicast approach. Detailed designs are discussed in Section 3. To reduce the cost of network probing, we propose a heuristic mechanism called PeerTop in Section 4. Simulation results to demonstrate the performance of the proposed algorithm are shown in Section 5. In Section 6, we discuss some related works. Section 7 concludes this paper.

2 A Multicast Approach to File Distribution

The multicast approach to file distribution is different from that of live streaming. In a live streaming application, the resulting multicast tree expects that every node of the tree can receive a video stream smoothly. The tree nodes, which have sufficient bandwidth to receive the stream from a parent node, may have different start times to receive the stream, but the stream will last the same duration for all the nodes. Therefore, a live streaming application usually constructs a single multicast tree to connect as many capable nodes as possible to minimize the start time of each node. When packet-loss occurs, it may tolerate the absence of

some stream frames, but at the expense of stream quality. For file distribution, however, there is no explicit bandwidth limitation for a requesting peer to be selected as a tree node, so it is more likely that all the requesting peers can be jointly organized as a multicast tree. Besides, a copy of the requested file received by a requesting peer must be intact. The time for nodes of the multicast tree to receive an integral replicate of content may be different from each other, and it is low-bounded by the slowest connection along the tree path from the node to the tree root.

There are many challenges to organizing a group of autonomous peers into an efficient multicast tree over the Internet. For example, requesting peers may have different levels of computing or networking capacity; a peer may stay behind a firewall, or refuse to forward contents; or some of the paths between two peers may be congested, or very slow. If a connection of the tree between two requesting peers is very slow, all the peers descending below the connection will suffer a delay caused by the slow connection even if they have very fast connections to others. Thus, if a multicast algorithm does not avoid choosing peers with low-bandwidth connections at the upper levels of the tree, it is possible to construct a very slow multicast tree where a requesting peer may wait much longer than in the amplification approach to receive the content.

Amplifiable Multicasting. Since the amplification approach and the multicast approach have their strengths and weaknesses, we propose a hybrid approach called amplicast for content distribution. In amplicast, we may not connect all the requesting peers in a single multicast tree. For example, during the construction of a multicast tree, if a requesting peer finds that joining the tree to receive the content at the current round will result in a later start time than joining another multicast tree at some later round, the peer will not be connected to the multicast tree at the current round. This implies that amplicast may construct more than one multicast tree to distribute requested content from the source peer to a group of requesting peers.

Although a peer may find that waiting to request another peer at the next round is probably better than joining a multicast tree at the current round, it cannot be guaranteed that the expected peer will be available at the next round. Due to limited capacity of peers, the requesting peer may need to wait for one or more rounds; even longer than joining the tree immediately. If the requesting peer is unlucky, it may incur a starvation problem where it always waits for a better supplying peer, but it does not have high priority to be served. Therefore, a requesting peer should carefully consider whether to wait for more rounds.

Path-aware Multicasting. Before constructing the multicast trees, there is the critical issue of how to probe and collect useful network information among the connections of peers. Traditionally, most P2P applications only consider the static links of a peer to the outside and assume that the network status of a peer to the other peers is fixed. The network information for a peer in these P2P systems is usually assigned manually or measured once at boot time. Such information may mislead the peer to make a wrong decision. On the Internet, the network status of end-to-end connections varies from path to path. The

bandwidths and latencies from a peer to other peers may be very different. In this paper, we assume the peers will probe each other to measure real-time network information, such as bandwidth and ping time or delay, as it does in an overlay networking system [1]. Then, the source peer in the amplicast algorithm will collect and analyze as much network information as possible and update to construct the multicast trees for content distribution.

3 Amplicast: Hybrid of Amplification and Multicast

The basic steps of the proposed amplicast approach are explained as follows. First, the source peer who wants to distribute content to a group of requesting peers collects peer-to-peer network information. Then, it analyzes the information and constructs multicast trees. When the source peer picks one candidate peer to join the tree, the peer can decide whether to accept or not. The candidate peer may decide to wait for the next round. This strategy considers both the benefits of the source peer and the requesting peers. If a multicast tree cannot include any more requesting peers, the algorithm will work use the amplification approach. The implementation of the algorithm consists of network probing, group setup, and content transmission phases.

Network Probing. When a source node receives requests from others, it individually acknowledges these requests and asks the requesting nodes with a probe set to measure the network status of their connections to the set. With admission control mechanism [4], the source node can identify free-riding nodes and decide which nodes it will serve and what kind of information it needs. The admitted nodes form the probe set. After receiving the acknowledgement message and the probe set from the source node, a requesting node will start to probe each other node. When the probe is finished or a requesting peer is asked to respond immediately, it may return the complete or partial result to the source node. Note that we can ask a node X to report the download bandwidth it can achieve from another node Y. If node Y is behind a firewall or refuses to transmit packets to node X, node X can tell the source node this situation. If no nodes can measure their download bandwidth from node Y, the source node will detect that node Y is a freeloader or that it is behind a firewall.

Group Setup. After collecting the network probe information from the requesting nodes, the source node conducts the amplicast algorithm to construct multicast trees for transmitting content. The planning results include the tree structure, the timing information, and specifying when and from whom a node should request content. The source node then sends sufficient information to each node, so that they can begin setting up connections for content multicasting.

Content Transmission. When the nodes have set up a multicast tree and prepared for transmission, the source node starts to transmit content through the structures it organizes.

Several design issues arise in the design of amplicast:

Peer Selection. Because the link bandwidth of the node connecting to one node is usually different from connecting to others, if we specify the link band-

width of a peer with only one static parameter like most current P2P systems, it might happen that the node often gets content fast, but does not have enough bandwidth to serve others. Therefore, if we utilize the pair-wise network information of peers, we can estimate a node's real capacity more precisely and take a firewall or NAT into consideration.

On the other hand, a requesting peer has different criteria for source selection. A requesting peer is most likely not to care about how long a source peer will take to distribute, or how long the other peers will take to receive the content. For its own sake, it would like to be served by a capable source peer, so it can get a copy of the content quickly and reliably, even without the need to contribute. Since most of the requesting peers are greedy and act alike, some source peers tend to become popular and "hot", so for a future request it might queue for a long time. This would probably make the system unstable and unfair. If a requesting peer can choose a source peer based on the information about not only the source's capacity, but also its load, we would probably be able to make the load on the source peers more balanced and thus reduce the probability that a request needs to be queued.

Finish Time Prediction. In the selection of the next requesting peer to be included in the multicast sub-tree, the source peer selects the pending peer with the smallest finish time, so it can keep the longest waiting time it needs to serve the last peer as short as possible. Rather than getting the content from the parent peer suggested by the source peer, a candidate peer will evaluate whether it is faster to wait to get the content from another peer that is occupied in this round. Similar to the amplification approach, a peer becomes a new source peer after it receives a replicate. By evaluating the predicted finish times to get content from other replicate peers, a requesting peer selects the best peer to obtain content from, or decide to wait for another peer to become a new available source peer. Thus, before content transmission, the source peer estimates how long it will take to serve all the requesting peers, and a requesting peer estimates how long it will take to get the content and who the best parent peer is to receive the content from.

Incentives. Our model provides incentives for requesting peers to cooperate with others. If a peer can transmit content to other peers, the source peer can get this information from other peers and utilize it in the construction of the multicast trees. If the peer joins the multicast group, it may help itself to receive content faster. However, if a peer refuses to cooperate with others such that no other peers can receive content from it, the source peer will give it lower priority during the construction of multicast trees. The peer will take the risk of being served in the last round. We believe such incentives will enable the source peer to distribute contents to a group of peers fast.

4 PeerTop: Lightweight Network Probing

Network probing is a very active topic of research for the Internet. For example, RON [1] uses an active probing mechanism to collect information about latency,

packet loss rate and throughput for each virtual link. The PDF [5] system also invokes an all-pair TRACEROUTE session to collect topological information and obtain a set of disjoint paths. RON and PDF systems have demonstrated that application-controlled relay routing in P2P overlay networks can improve the efficiency and reliability of conventional IP-layer routing. However, these methods rely on heavyweight network probing, where each peer node probes and collects link information between each other.

Building on top of a relay-routing overlay like RON, we can implement a cooperative content distribution system using amplicast without extra probing cost. For underlying relay routing, each node of the overlay already constantly maintains all pair-wise link information by heavyweight network probing. Therefore, when a peer is requested by the source node to measure the network status of its connections to the peers of the probe set, it can simply retrieve the timely results maintained by the underlying overlay. In the basic amplicast approach, every requesting node is asked by the source node to evaluate its link connections to all the other requesting nodes, and then report the probed data to the source node. The source node needs to collect all the pair-wise link information from all the requesting peers. For a probe set of N nodes, though the N-by-N link probing might be conducted simultaneously and the content transmission time should be much longer than the network probing time, it will still incur substantial network message overhead and delay.

The amplicast algorithm works well even though the probe data is not complete or timely. In a P2P system without link information provided by the underlying overlay, we adapt our idea for low-cost relay routing [2] to develop PeerTop, which aims to reduce network probe overhead without any significant decrease in performance. In PeerTop, each peer will cache all the link information newly probed or collected. When a requesting node receives a probe set, it will order the set first based on their previous probed data recorded in the cache, and then probe the nodes of the set according to the order. In the best case, a requesting node will be able to probe all the nodes in the probe set in time. After a while, the source node will ask a requesting node to return the link information of the top nodes, which are a portion of the probe set with better link connections to the requesting node. In the next section, we will simulate the effects of top set size on the performance of amplicast.

5 Experimental Results

To evaluate the performance of the proposed amplicast algorithm, we implement all the approaches described above and test them on networks of different sizes and topologies. First, we use Brite to generate the random networks and Waxman models to generate topologies of 64, 128, 256, 512, and 1024 nodes. For each size, we generate 100 topologies and average the simulation results. We use heavy-tailed distribution for bandwidth. The length of the content for distribution is 100 Mbytes. A peer will serve at most four other peers at a time. Further, we collected the PlanetLab dataset between May 24 and May 30, 2004 every two

hours using a similar method described in [2]. By screening out the PlanetLab nodes with node failure or link failure, the collected dataset consisted of 212 nodes. A prototype of 64 nodes was also implemented on PlanetLab.

Longest Waiting Time. Given a set of requesting peers, the longest waiting time measures how long the system takes to distribute the content to all the peers. If a simulated network has a link with very slow bandwidth and the algorithm does not avoid it, it tends to affect the performance of the system badly. Fig. 1(a) shows the average longest waiting time of the three approaches for different network sizes. The amplicast approach performs much better than the other approaches. Since the amplification approach considers only the bandwidth of the outbound link for a peer, it is very likely that the real bandwidth of an end-to-end connection for the peer to another peer is much slower. In addition, to amplify the capacity of the system quickly, the amplification approach favors the peers with high bandwidths, so the peers with lower bandwidths are always served in the last round. This will delay the system and extend the time required to finish serving all the peers.

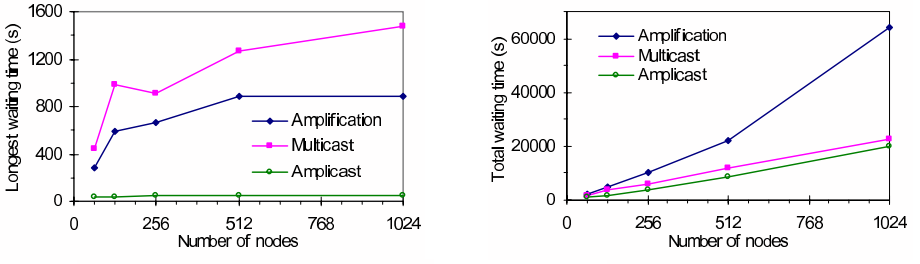


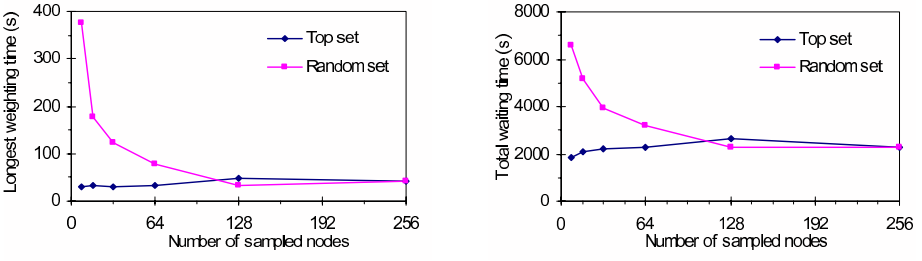
Fig. 1. Simulation of Brite model

Similarly, the multicast approach may also choose to serve a requesting peer through a low-bandwidth link, even as it considers the N-by-N network information. However, it is surprising that the multicast approach performs worse than the amplification approach. This happens when the requesting peer can be served faster by other parent peers, but they are occupied in the current round. In the multicast approach, a requesting peer is always served through an available link, even if the link is very slow. Therefore, the longest waiting time of the multicast approach is determined by the path with the most inefficient link. In the amplicast approach, however, a requesting peer can evaluate whether there is an alternative parent peer that is worth waiting for. In such a case the amplicast algorithm can avoid transmitting content through a very low-bandwidth link. The simulation results illustrate the advantages of the amplicast algorithm.

Total Waiting Time. Total waiting time is the summation of waiting times for all peers. As shown in Fig. 1(b), the amplification approach grows two times more than the multicast and amplicast approaches. In the multicast and

amplicast approaches, a requesting peer can send content to other peers while receiving content, so multiple peers can start to receive content earlier than that in the amplicast approach. This implies that most peers in the multicast and amplicast approaches need not wait for a round. Because the multicast approach is greedy to start transmission for all requesting peers, some of the peers may receive content through a slow link. In the amplicast approach, however, this kind of peer may decide to start receiving content later from another parent peer so that it can obtain the content faster through a better link. Since the starting times constitute a large part of the total waiting time, the improvement of the amplicast algorithm over the multicast approach is not as significant as the improvement over the amplification approach.

Effects of Top-set Size. To evaluate the performance of PeerTop, we examine the cases of 8, 16, 32, 64 and 128 top nodes for the topologies of 256 nodes, and compare it with the random-set approach that selects nodes randomly. As shown in Fig. 2(a) and (b), the total waiting time and the longest waiting time do not change significantly for the top-set approach, but they drop dramatically as the number of sampled nodes increases for the random-set approach. It is clear that the top-set sampling performs better than the random-set sampling, especially when the number of sampled nodes is small.



(a) Longest waiting time (b) Total waiting time

Fig. 2. Evaluation of PeerTop

Because the amplicast approach uses heuristic finished-time prediction for peer selection, increasing the top-set size may harm the performance. As the top-set size increases, a requesting peer may mispredict whether to obtain content at the current round or at rounds through a lower-bandwidth link, which may not be included in a smaller top set. When the size of top-set for PeerTop is very small, every requesting peer in the amplicast algorithm will obtain the requested content through a link with very high bandwidth; otherwise, it will need to wait for later rounds. This implies that the transmission time to retrieve the content from a supplying peer via the high bandwidth link will be small, but it may have to wait longer for the right round. If there are sufficient top links to connect all peers, it will seldom happen that a requesting peer in the amplicast approach needs to wait for next rounds.

PlanetLab Dataset. Using different top-set sizes of 12.5%, 25% and 50% of the total number of PlanetLab nodes, we compare the performance of PeerTop with the N-by-N network complete probing. As shown in Fig. 3, the longest waiting time and the total waiting time of the top-set sampling are decreased as the number of sampled nodes increases, but the finish time distribution for 12.5% top-set sampling is better than that of N-by-N. As explained in the results for the longest time, many more peers in the multicast and the amplicast approaches can start to receive content earlier than in the amplicast approach. A few of the peers in the multicast approach receive content through a slow link, but these nodes in the amplicast approach may start receiving content later from another parent peer through a faster link. In addition, in Fig. 3(d), we find that more nodes in the 12.5% top-set sampling will utilize better links to their parent than that in the N-by-N sampling. The experiment with the PlanetLab dataset also showed that PeerTop can perform as well as the amplicast approach with N-by-N link information, and the top-set sampling performs better than the random-set sampling, especially when the number of sampled nodes is small.

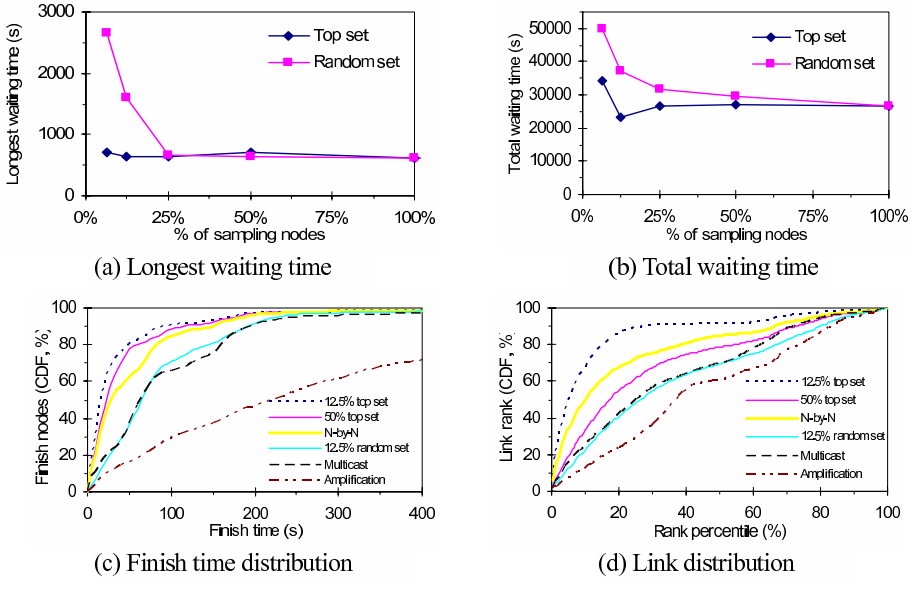


Fig. 3. Evaluation of PlanetLab dataset

6 Related Work

Several approaches have been proposed to encourage peers to cooperate. Xu et al. [8] proposes an algorithm for fast system amplification that encourages all the nodes that have received files to serve other waiting nodes. Zigzag [7] provides a regular way to form a multicast tree and maintain its completeness. CoopNet [6]

cooperates with clients to construct several distribution trees. If one sub-stream fails, other sub-streams still form the major streaming with distortion. Similar to amplicast, Narada [3] makes use of network dynamics, but it optimizes for packet transmission latency rather than for distribution time of the file. Further, it targets video conferencing applications that need live streaming in a single round and can tolerate packet loss. Amplicast, though, supports intact file transmission and may construct multiple multicast sub-trees to utilize high-bandwidth links.

7 Conclusion

In this paper, we propose a hybrid approach that utilizes file amplification and stream multicast to address the problem of file distribution. We also propose PeerTop, which includes link cache and a heuristic of top-set sampling, for lightweight network probing. In the proposed amplicast algorithm, peer selection based on finish time prediction and incentive of peer makes it quite different from conventional multicast algorithms. It can potentially reduce the load of the source peer and reduce the waiting time of the requesting peers. By considering the bandwidth of end-to-end paths and the benefit to requesting peers, the amplicast algorithm can organize peers into a series of multicast trees that can efficiently distribute content to a group of peers. It combines the advantages of amplification and multicast: in multicast, most peers can start to receive content earlier, and in amplification, peers can wait to choose a better server in order to avoid receiving the content from a low bandwidth link. Our simulation results show that the proposed amplicast algorithm significantly outperforms the amplification and multicast approaches.

References

1. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris: Resilient Overlay Networks. ACM Symposium on Operating Systems Principles, 2001.
2. Chen-Mou Cheng, Yu-Sheng Huang, H.T. Kung, and Chun-Hsin Wu: Low-Cost Relay Routing for Achieving High End-to-End Performance. IEEE Globecom, 2004.
3. Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang: A Case for End System Multicast. ACM SIGMETRICS, 2000.
4. H. T. Kung and Chun-Hsin Wu: Differentiated Admission for Peer-to-Peer Systems: Incentivize Peers to Contribute Their Resources. Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.
5. T. Nguyen and A. Zakhor: Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks. IEEE INFOCOM, 2003.
6. Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou Kunwadee Sripanidkulchai: Distributing Streaming Media Content Using Cooperative Networking. ACM/IEEE NOSSDAV, 2002.
7. Duc A. Tran, Kien A. Hua, Tai Do: ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming. IEEE INFOCOM, 2003.
8. Dongyan Xu, Mohamed Hefeeda, Susanne Hambrusch, Bharat Bhargava. On Peer-to-Peer Media Streaming. IEEE ICDCS, 2002.

Dynamic Algorithms to Provide a Robust and Scalable Overlay Routing Service

Bart De Vleeschauwer, Filip De Turck, Bart Dhoedt, and Piet Demeester

Ghent University - IBBT - IMEC, Department of Information Technology
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
bart.devleeschauwer@intec.ugent.be

Abstract. Service providers and companies wishing to connect a number of distributed sites need a QoS enabled and resilient network to provide their services. As network providers can not yet offer multidomain end-to-end QoS and Internet path outages can last several minutes, using Overlay Service Networks to route around congested or failing parts of the network is a hot topic in the research community. Typically a full mesh topology is used to connect the servers of the Overlay Service Network. Because this approach is not scalable, we propose to use a dynamic topology that is only a fraction of the full mesh. A novel algorithm that automatically reconfigures the topology when link outages or congestion occur is introduced. We have also developed an on-demand overlay routing algorithm that decreases the overlay network load. Through simulation it is shown that our algorithms allow to offer a robust routing service in a scalable way.

Keywords: Resilience, Overlay Network, Overlay Topology, QoS, Overlay Routing.

1 Introduction

Service providers need QoS and resilience to offer a good and reliable service to their customers. It is therefore essential that their data does not cross congested or failing Internet links. Currently the Internet still lacks the ability to provide real end-to-end QoS. Routing in the Internet is achieved by the interaction between BGP routing between autonomous systems (ASs) and shortest path routing within an AS. BGP and shortest path routing like OSPF focus on the minimization of the overall network load by minimizing the hop count but do not take QoS metrics into account and do not offer any guarantees regarding the delay/bandwidth on a path. Often better paths, with a lower delay and more bandwidth, are available [1]. When the traffic on a physical link exceeds a certain threshold, this causes link congestion, resulting in massive packet loss and long delays. While research on QoS solutions at the network layer has already resulted in the development of different promising techniques, like IntServ and DiffServ, their deployment is still scarce. In order to provide end-to-end QoS, QoS techniques need to be implemented in all the ASs in the Internet. Another

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 945–954, 2006.

problem in the Internet is the duration of Internet path outages. When Internet links or routers fail, the network needs to find an alternative route, and routing tables need to be updated. In [2] the authors show that Internet path failures may last longer than 5 minutes in 30 % of the cases, and in 4 % of the occurences the path even goes down longer than 30 minutes.

In recent years there has been a lot of research on the provisioning of services by a third party located at the edge of the network. This can either be a group of end users, collaborating to exploit the use of their shared resources and equipment, or a service provider that wants to offer a service the network itself can not yet provision. Examples of the former are the success of peer-to-peer networks to share files across the Internet and to offer a VOIP service[1]. Examples of the latter can be found in [3,4,5,6]. In [3] the authors have developed an experimental overlay network that uses a full mesh topology to provide a resilient communication infrastructure at an overlay layer and in [4] the authors looked at server location algorithms for overlay networks with the overall acceptance rate for multimedia connections as an optimization criterion. [6] compares a number of static topologies to provide a resilient overlay service. The authors conclude that the topology has a great impact on the overall performance of overlay networks.

In this paper we look at overlay service networks (OSNs) to offer a resilient QoS enabled routing service in a scalable way. By providing an overlay layer routing infrastructure, OSNs are able to route around failing Internet links and avoid congested parts of the network. Existing solutions use full mesh topologies [3] or smaller static topologies [5] to interconnect the overlay servers. While using a full mesh topology allows to find optimal routes and is very robust due to its high connectivity, this approach is hardly scalable, and reported to only be sustainable for overlay networks having up to 50 nodes [3]. Using a static topology that is only a subgraph of the full mesh topology is scalable, but can make the overlay network disconnected in case of node/link failure and is often unable to find good routes between every pair of overlay servers. We therefore propose to use a dynamic topology that at any time only contains a fraction of the links present in the full mesh topology and solves the connectivity issues of using smaller static topologies. When an Internet fault occurs or some Internet links get congested, the corresponding overlay links are removed from the topology and new overlay links are created, thus enabling the overlay provider to provide a fully connected overlay network. We have developed a distributed topology reconfiguration algorithm (DTR) that is able to significantly improve the connectivity of the overlay topology. In addition we describe an on demand overlay routing algorithm that minimizes the number of overlay hops (LSODR). By combining the topology reconfiguration algorithm and our routing algorithm we are able to offer an overlay routing service that is not only able to provide the ability to route around Internet path faults and provide QoS, but is also scalable, thereby minimizing the consumption of both network and overlay resources. We

[1] www.skype.com

use simulation to prove the effectiveness of both the topology reconfiguration and routing algorithms.

This paper is structured as follow: Section 2 describes the OSN concept and the key service it provides. Section 3 houses the relevant algorithms to build an overlay topology and section 4 discusses how to maintain the dynamic overlay topology. Our novel routing algorithm is described in 5, the simulation results are located in 6 and conclusions are drawn in 7.

2 OSN Concept

An OSN consists of a number of overlay servers placed by a service provider at sites that need to be interconnected with each other in a reliable and robust way. The core functionality of these servers is providing an overlay layer routing infrastructure that is able to find alternative routes when the direct Internet connection between two sites should fail or degrade severely. When this happens, the clients that use this connection send their traffic to the overlay server, instead of using the direct Internet path. The overlay network is then responsible for finding an alternative route for the connection between the end points by routing at the overlay level. In order to have an overlay path available between the different sites at any time, the overlay network maintains a logical topology that defines the connectivity of the OSN. By probing the overlay links regularly, values for their end-to-end delay, loss and bandwidth can be inferred. The overlay servers exchange link state messages and are thus able to have a consistent view of the overlay topology and the link states of the different overlay links at any time.

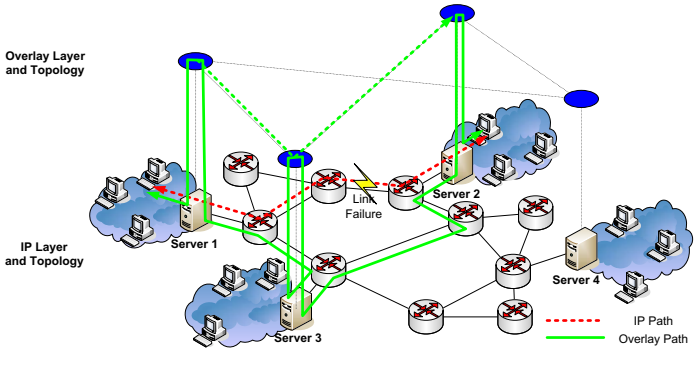


Fig. 1. Using an overlay network to circumvent IP failure

In fig. 1 the core functionality of an OSN is depicted. The direct Internet path between sites 1 and 2 is compromised by the failing of an IP link, the OSN is able to circumvent the link failure by routing via server 3 in the overlay network, thus enabling sites 1 and 2 to communicate. Both the overlay layer, with the overlay

topology and the network layer are shown in the figure. In subsequent figures we will no longer show the IP layer.

The way an OSN provides its connectivity service is determined by its solution to three subproblems. First, an OSN needs to construct a topology that connects the overlay servers. Second, the overlay network needs to maintain this topology and needs to react to the dynamics of the underlying Internet. A third subproblem is the way the OSN routes its traffic in the logical topology. A discussion of these subproblems and their solutions is provided in sections 3, 4 and 5 respectively.

3 OSN Topology Determination

In order for the overlay network to function properly, it needs to construct a topology, consisting of overlay links, that connect the overlay servers. An overlay link connecting two servers corresponds with the Internet path connecting these servers. The overlay network sees the underlying IP network as a black box and can only infer end-to-end information for the construction of the overlay topology. This means that we have information on end-to-end delay and bandwidth, but do not know the Internet topology on which the overlay network is built and the IP routers that are used when traffic is sent from one overlay server to another. While there are several ways to construct the topology, it determines several important characteristics of the overlay network. An ideal topology is able to provide backup paths for the connections between every pair of servers at a low cost and does not require a huge amount of probing overhead. The overhead to maintain an overlay topology is a direct consequence of the number of edges in the overlay network, because every edge needs to be probed at regular intervals by the connected overlay servers and its state needs to be forwarded to the other overlay servers. As it is possible to connect any pair of overlay servers with an overlay link, due to their connectivity in the underlying network, a careful selection must be made as to what overlay links to include in the topology to realize a scalable OSN that is also able to offer a robust routing service. Hereafter we describe 4 topology construction algorithms:

Full Mesh(FM). A full mesh topology has an overlay link for every pair of overlay nodes in the network. While it offers a very high connectivity, it is not scalable as the number of overlay edges scales quadratically with the number of overlay servers.

***k*-Minimal Spanning Trees (*k*-MST).** This topology was proposed in [6] and builds k minimal spanning trees that share as few overlay links as possible.

Mesh Tree (MT). This topology is constructed by building a minimal spanning tree and then connecting the nodes in this tree that have a grandparent-grandchild relationship or that are siblings.

(k, l)-Bounded Spanning Tree ((k, l)-BST). We propose this topology construction algorithm in order to guarantee that the maximal delay experienced

by a connection never exceeds a certain threshold. First the k-MST algorithm is applied to construct an initial topology consisting of k MSTs. In the next step, the minimal delay of all the possible connections in that topology is checked, if the delay of a connection is higher than a threshold, new links are added to the topology. The l shortest paths between the end points of that connections in the FM topology, that do not exceed the threshold, are computed and the links of the path requiring the least number of new links are added to the topology.

4 OSN Topology Reconfiguration

When an IP link fails or is congested, the overlay links using this IP link are also affected. As the overlay links are probed at a regular interval, the overlay servers at the end points detect the overlay link failure and must act appropriately. When using a static topology, the overlay servers react by eliminating this link temporarily from the overlay topology and no longer sending traffic through it. When using a full mesh topology, the redundancy in available overlay links makes sure that a backup path can be provided. However, topologies with fewer links might get disconnected due to the failure of some links. In this paper we argue that the flexibility of the overlay network must be used to adapt the overlay topology to the changed Internet environment. So instead of having a static topology, we propose to dynamically alter the overlay topology to continue to offer a fully connected overlay network, when an overlay link fails. For this end we have developed the Dynamic Topology Reconfiguration (DTR) algorithm.

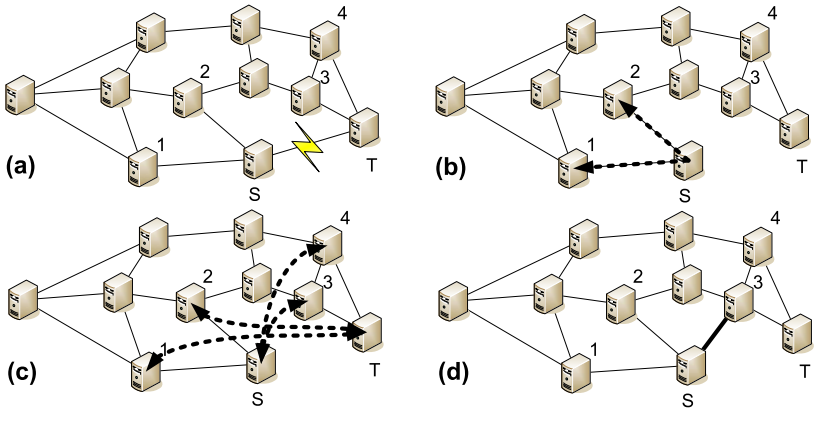


Fig. 2. Reconfiguring the overlay network: in (a) an overlay link fails due to a failure in the underlying network. When an overlay server S detects the link outage, it asks all its neighbors (servers 1 and 2) to probe the target node of the link (b). Servers 1 and 2 then probe T and S probes all the servers (servers 3 and 4) with a link to the target node itself (c). The link that is best suited to replace the failing link (S-3) is then added to the overlay topology (d).

Dynamic Topology Reconfiguration. The core idea behind the DTR algorithm is that when an overlay link fails, this link must be replaced with a new overlay link that is able to provide a new, low-delay, path to replace the failing edge. The DTR algorithm works in the following way: when a failing overlay link $s - t$ is detected, one of the endpoints s of the link tries to find a new overlay edge. To do this it probes its delay to the neighbors of t that are not neighbors of itself and also asks its neighbors that do not belong to the neighbor set of t to probe their delay to t and report it back to s (s can find the relevant servers, that engage in the DTR algorithm, in the overlay topology). The s server then determines the new intermediate node n that creates the $s-n-t$ path with the lowest delay value. It signals the creation of a new overlay link to the involved servers and the other overlay servers are also informed of the creation of the new overlay edge. An example of this algorithm is depicted in figure 2. A key feature of this algorithm is that it functions in a distributed way and only relies on the communication between a limited set of overlay servers.

5 Load Sensitive On Demand Routing (LSODR)

Standard overlay routing algorithms minimize a metric like delay or loss. To route the data of a connection, every overlay server keeps a forwarding table that contains the next overlay hop towards a target node for the active overlay connections. This forwarding table is periodically updated in order to be able to reflect the changes in the logical topology. For such an update, the overlay server determines the new next hop for all the active connections and removes the entries for the connections that are no longer active. In order to minimize the load experienced by an overlay network we developed the Load Sensitive On Demand Routing algorithm (LSODR) that is designed specifically for overlay networks. The idea behind this algorithm is that when an overlay connection passes an overlay server on its way to its destination, this server probes the direct Internet connection between itself and the target node of that connection. If this direct connection is also acceptable, it will forward the data directly to the target node instead of forwarding it to an intermediate overlay server. In doing this, the overlay server will use an "Internet shortcut" to the destination of the connection and is able to reduce the load of the OSN, since the number of involved overlay servers for the connection decreases. Fig. 3 depicts how the LSODR algorithm works in an example network, the forwarding tables are shown for the relevant servers.

6 Evaluation

To evaluate the algorithms, we have built a multilayer network simulator, that simulates both the overlay layer and the network layer. We performed a number of simulations on the European network depicted in fig. 4. This network has 37 nodes and 57 edges and connects a number of major cities in Europe. The results that we present are averaged over 200 iterations in which overlay servers were

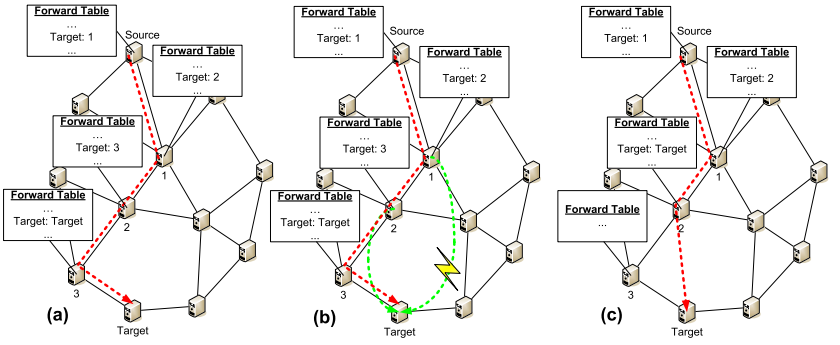


Fig. 3. LSODR algorithm illustrated: at first the connection uses the best overlay path that is found in the overlay topology between Source and Target (a). The intermediate servers periodically probe their direct connectivity to Target, and detect that there is also a direct 2-Target link (b), as a result, server 2 decides to forward the traffic directly to the target node(c), thereby reducing the load on server 3.

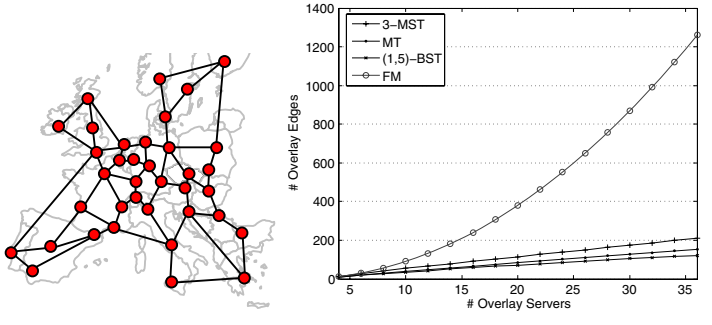


Fig. 4. The network we used to test the different algorithms and the number of overlay edges for topologies constructed with the topology construction algorithms for different sizes

randomly placed in the network. To simulate the failure of an IP edge due to congestion or a network fault, links were chosen at random from the network. We assumed that the OSN has the time to adapt its topology to the changes caused by every single link failure, before a new failure takes place. Routing in the IP network was supposed to minimize the hop count.

6.1 Overhead Comparison

To measure the overhead caused by the different topology construction algorithms, we generated networks of different sizes and constructed the corresponding overlay topologies with the algorithms. We used the number of overlay edges as the criterion for the probing overhead. Figure 4 shows the number of overlay edges for an increasing number of overlay nodes.

Obviously, the number of overlay edges in a full mesh topology scales very badly. The number of overlay edges is quadratic in the number of overlay servers for this topology. All the other investigated topologies show a linear relationship between the number of overlay servers and the number of overlay links.

6.2 Recovery Ratio

To evaluate the way an overlay network reacts to IP link failure, we varied the number of failing IP links and the size of the overlay network. In a first set of graphs, we show the recovery ratio, defined as the ratio of the number of recoverable connections to the number of failing connections. This expresses the number of connections that can be recovered by routing at the overlay layer when their direct Internet connection fails. A connection in our discussion represents the ability to send data between two servers. Graph 5 shows the behavior of the different topologies for increasing numbers of failing IP links in an overlay network of 20 servers and the evolution of the recovery ratio for an increasing number of overlay servers with 5 failing IP links.

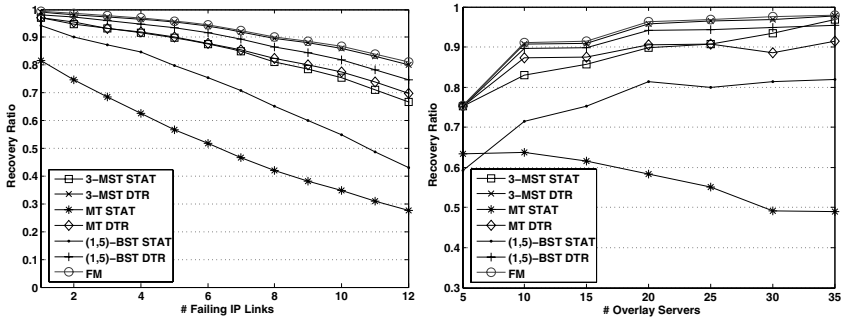


Fig. 5. Evolution of the recovery ratio with an increasing number of failing IP edges and an increasing overlay network size, STAT and DTR are used to respectively denote the plain removal of the failing overlay edge from the topology and the usage of the DTR algorithm to reconfigure the topology

The results clearly show that by dynamically changing the overlay topology, the overlay network is able to significantly improve the overall recovery ratio and offer a service similar to that offered with a full mesh topology. There is also a dependency on the total number of overlay edges in the topology. The 3-MST topology, which has the most links, is able to almost reach the recovery ratio of the full mesh topology with the DTR algorithm. The performance of the DTR algorithm depends on the number of overlay edges as more overlay edges increase the number of neighbours of a server and consequently the probability of finding good links to replace the failing overlay edges. The MT topology, which has a very bad performance in the static case, can even be rendered more robust than the static 3-MST topology by using the DTR algorithm. When varying the

number of overlay servers, we see that the DTR algorithm is able to improve all topologies for different overlay sizes. We also observe that for the majority of the topologies, an increasing number of overlay servers increases the overall recovery ratio. The reason for this is that when an IP link fails, a bigger overlay network will cover a bigger part of the network and is thus able to route around the failure more efficiently. The MT topology seems to be an exception to this general rule. After a thorough analysis of the data, we saw that the reason for this is that when the overlay network is deployed at many sites in the network, the MT topology will often have multiple links using the same IP edges, as a result it will be less able to route around the congestion of such a link and performs worse. Overall, the DTR algorithm performs well for overlay networks of different sizes and is able to enhance all the topologies we tested, even with very bad network conditions (12 failing IP links). As the number of edges in the BST, MST and MT topologies scales linearly with the number of overlay nodes, we are able to provide a resilient OSN routing service in a scalable way by using the DTR algorithm.

6.3 Routing Evaluation

To evaluate the advantage of using LSODR routing, we look at the LSODR overlay load reduction, defined as the ratio of the difference in number of overlay hops between the LSODR path and the lowest delay path to the number of hops in the lowest delay path, for the connections that are routed via the overlay network when their direct connection fails. The LSODR algorithm was configured in such a way that an intermediate server decided to route directly to the target node of the connection if the direct link resulted in a path with at most 50 % more delay than the standard overlay path found with shortest delay routing.

The graphs show that the reduction of the OSN load by LSODR routing depends on the specific topology and the network conditions. The reason that the performance depends on the specific topology is that topologies with many

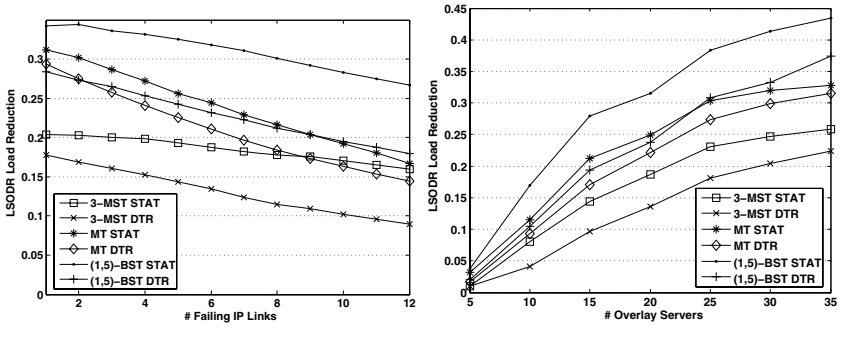


Fig. 6. The evolution of the LSODR overlay load reduction for a 20 server overlay network with increasing numbers of failing IP links and for overlay networks of different sizes with 5 failing IP links

links are already able to offer a short overlay hop path. However, when less links are available, as is the case with the (1,5)-BST algorithm topology, the overlay path with shortest path routing will be longer and the LSODR algorithm will perform better. The topologies that are maintained using the DTR algorithm have also got a lower load reduction, because they automatically create new low delay paths to replace failing overlay links. When more IP edges fail, the gain of using the LSODR routing is less. The reason for this is that the underlying Internet itself is worse due to more failing IP edges. As a result, the chance that the direct Internet connection from an intermediate server to the target node is valid is smaller. The second graph clearly shows that the advantage of using LSODR routing increases with the size of the OSN. This is due to the longer overlay path length for the larger overlay networks, which creates the "Internet Shortcuts" more frequently. This effect is especially visible for the topologies with few edges.

7 Conclusion

In this paper, we describe two algorithms to enhance overlay service networks. The Dynamic Topology Reconfiguration algorithm dynamically changes the overlay topology to adapt to changing Internet conditions. We have shown that it is able to greatly increase the recovery ratio for topologies constructed with a variety of topology construction algorithms, that have a number of edges that scales linearly in the number of overlay servers. We have also developed the Load Sensitive On Demand Routing algorithm that minimizes the load experienced by the overlay service network. Through simulation we show that this routing algorithm succeeds in leveraging the total load experienced by the overlay servers significantly, especially for larger overlay networks. These two algorithms provide an efficient way to provide a robust overlay routing service, that is scalable and only causes a minimal overhead.

References

1. S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, "Detour: a case for informed internet routing and transport," Tech. Rep. TR-98-10-05, 1998.
2. N. Feamster, D. Andersen, H. Balakrishnan, and F. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," in *Proc. ACM SIGMETRICS*, june 2003.
3. D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Symposium on Operating Systems Principles*, pp. 131–145, 2001.
4. B. De Vleeschauwer, F. De Turck, B. Dhoedt, and P. Demeester, "On the construction of qos enabled overlay networks," in *Quality of Future Internet Services (QofIS)*, vol. 3266 of *Lecture Notes in Computer Science*, pp. 164–173, september 2004.
5. L. Lao, J.-H. Cui, and M. Gerla, "Multicast service overlay design," in *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05)*, 2005.
6. Z. Li and P. Mohapatra, "Impact of topology on overlay routing service," in *INFOCOM*, p. 418, March 2004.

A Decentralized Scheme for Network-Aware Reliable Overlay Construction

Shinichi Ikeda, Tatsuhiro Tsuchiya, and Tohru Kikuno

Graduate School of Information Science and Technology, Osaka University,
Osaka 565-0851, Japan
{t-tutiya, kikuno}@ist.osaka-u.ac.jp

Abstract. Peer-to-peer (P2P) systems are often constructed in overlay networks at the application layer without taking the physical network topologies into consideration. The mismatch between physical topologies and logical overlays can cause a large volume of redundant traffic and considerable performance degradation of the P2P systems. In order to alleviate this mismatching problem, we propose an algorithm that iteratively reshapes the topology of an overlay. The algorithm is fully decentralized and only relies on local information available at each node. Also, the algorithm preserves the total number of links during the process of iterative modifications, thus maintaining the resiliency to failures.

1 Introduction

In the past few years, we have seen the rapid growth of peer-to-peer (P2P) applications and the emergence of overlay infrastructures for the Internet. A challenging research problem in this area is to develop overlay architectures that can support these P2P applications without overloading network resources.

We discuss the issue of topology mismatching in this paper. Overlay networks are typically constructed at the application layer without taking the physical network topologies into consideration. The mismatch between physical topologies and logical overlays can cause a large volume of redundant traffic and considerable performance degradation of the P2P systems [1].

Recent approaches to building efficient overlay routing networks employ the abstraction of a distributed hash table (DHT) [2,3,4]. These DHT schemes use a global naming scheme based on hashing to assign keys to data items and organize the nodes into a graph that maps each key to a responsible node. The graph is hierarchically structured to enable efficient routing with the DHT. These protocols assume obedience to the protocols and ignore participants' incentives. In reality, however, nodes may behave selfishly, seeking to maximize their own benefit.

In this paper, we consider unstructured overlay networks consisting of nodes that behave selfishly; we propose a solution to the topology mismatching problem for such networks. Unstructured overlays are widely used in Internet-wide deployed systems. These overlays do not rely on global naming or any additional hierarchical structure. Instead, they use flooding, epidemic protocols, or random walks to disseminate messages or to query content stored by overlay nodes.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 955–964, 2006.

Schemes for building unstructured overlays are typically oblivious to the underlying network topology. Hence, communication between nodes on the overlay tends to impose a high load on the network and to result in unnecessary performance degradation.

The proposed algorithm optimizes unstructured overlays according to a proximity criterion in order to reduce network load. The algorithm iteratively reshapes the topology of an overlay in a way that reflects geographic locality so as to reduce network load. The algorithm is fully decentralized and only relies on local information available at each node. Also, the algorithm evenly balances the node degree while preserving the total number of links during the process of iterative modifications. Because of this property, the overlay can maintain and even improve the resiliency to failures.

Our work is the most closely related to the work by Massoulié et al [5]. In [5], they proposed an algorithm called LOCALISER, which shares many characteristics with our algorithm. Apart from small design details, the most significant difference between LOCALISER and our algorithm is that in ours each modification to the topology is performed only when the nodes involved in this action will benefit from this modification, which means that our algorithm allows each node to behave selfishly. In contrast, in the LOCALISER algorithm, the node that initiates a modification action must sacrifice its benefits by disconnecting a link to a neighbor node. Hence their algorithm can work well only when the overlay nodes are obedience to this algorithm and act in a self-sacrificing manner.

Other related work differs from ours, for example, in that dedicated servers are required [6], or in that resiliency is not taken into consideration [7,8].

The remaining part of this paper is organized as follows. In Section 2 we outline our design requirements for overlay networks. In Section 3 we present a basic idea behind the proposed algorithm and give the description of our algorithm. Results of an experiment are presented in Section 4. Section 5 concludes the paper.

2 Background

2.1 Unstructured Peer-to-Peer Overlay Networks

We can view an unstructured P2P overlay network as an undirected graph, where the vertices correspond to nodes in the network, and the edges correspond to open connections maintained between the nodes. A node i is said to be a *neighbor* of another node j iff they maintain a connection between themselves. The *node degree* of a node i is the number of i's neighbors. Messages may be transferred in either direction along the edges. For a message to travel from node i to node j, it must travel along a path between these node in the graph.

There are many schemes for building unstructured-overlay networks. In the most traditional scheme, a node that wishes to join the overlay network connects to highly publicized nodes or nodes known by a bootstrapping node. This preferential attachment feature and the incremental growth nature of P2P systems account for scale-free network-type topologies [1,9]. In these networks the

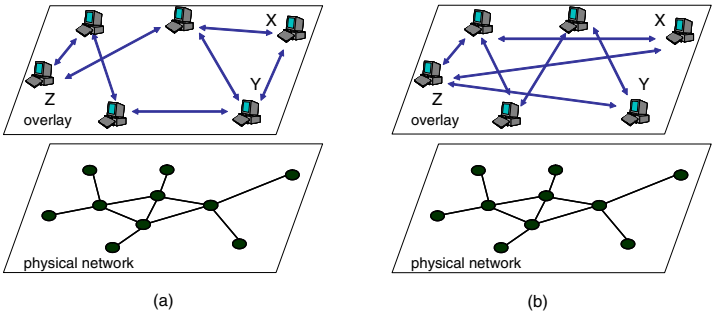


Fig. 1. Two different mappings of the overlay to the underlying network

node degree distribution follows a power-law; most nodes have few links and a tiny number of hubs have a large number of links. Due to this property, these networks are highly robust when facing random node failures, but vulnerable to well-planned attacks.

More sophisticated protocols can be used to obtain more "balanced" structures of overlay networks. For example, SCAMP [10], a decentralized protocol for building unstructured overlay networks, yields an overlay network where the degree distribution is a normal distribution-like curve. SCAMP can do this without maintaining any global information about the overlay.

These protocols are typically oblivious to the underlying network topology; all connections are treated as if they are equivalent, and so the resulting overlay does not reflect the underlying network topology.

2.2 The Topology Mismatching Problem

As stated before, overlay networks are often constructed without taking the physical network topologies into consideration. The mismatch between physical topologies and logical overlays is a major cause of performance degradation. Given the considerable traffic volume generated by P2P applications, it is crucial also from the perspective of their impact on the network infrastructure that they efficiently utilize available networking resources. The larger the mismatch between the underlying physical network topology and the P2P application's overlay topology, the bigger the stress on the underlying network.

This problem of topology mismatching is illustrated in Fig. 1. Fig. 1(a) and Fig. 1(b) depict overlay networks where six nodes are participating. As shown in these figures, these two overlays are built on the same physical network, which is schematically illustrated as the undirected graph where a vertex represents a router or an end host. In Fig. 1(a), the overlay closely matches the underlying network topology. The transmission of a message generated by node X to Y involves only two physical links.

In the right picture, on the other hand, the overlay topology does not match the infrastructure that well. In this case, the shortest path from X to Y in the

overlay is X-Z-Y which is two-hop long. If the message from X to Y is routed along this path, at least eight physical links are involved in this communication.

3 The Proposed Algorithm

To alleviate the topology mismatching problem, we propose an algorithm that reshapes the overlay topology by making each node modify the connections between their neighbors. Each node following this algorithm relies only on local information in determining how to modify its connections autonomously. The self-organizing behavior of an overlay network emerges from a collection of such autonomously behaving nodes.

To evaluate the global cost of an overlay topology G, we introduce the following cost function, as in [5]:

$$C(G) \equiv \sum_{(i,j)\in E} c(i,j) + w \sum_{i\in V} d_i^2 \tag{1}$$

where

- E is the set of edges.
- $c(i,j)$ is the cost of communication between nodes i and j. $c(i,j)$ is a proximity metric in the underlying physical network and could be the round trip time measured with ping or a more complex measure incorporating bandwidth availability on the path between i and j.
- w is a non-negative real constant.
- V is the set of nodes.
- d_i is the degree of node i.

This cost function is used in our algorithm to determine whether each possible topology transformation is effective or not; that is, if $\Delta C \equiv C(G') - C(G)$ is less than zero, then the transformation from G to G' is effective in the sense that the cost function value will be decreased.

As will be stated later, our algorithm maintains the total number of links. Hence the parameter w specifies the emphasis placed on degree balancing relative to locality. If $w > 0$, then the right term in (1) is minimized when d_i is the same for all i. On the other hand, if $w = 0$, the graph is optimized only to take into account network proximity.

3.1 Algorithm Steps

The proposed algorithm is fully decentralized. Each node i executes the following steps locally and periodically.

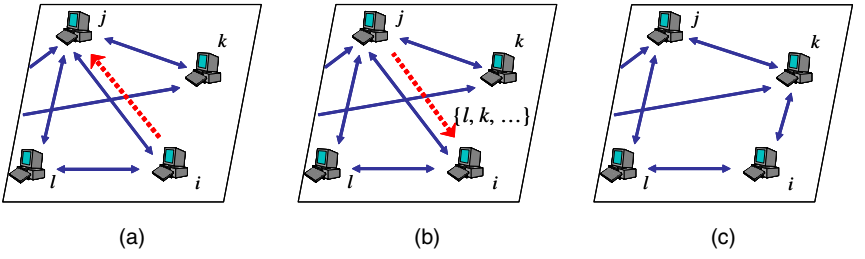


Fig. 2. Transformation steps

1. Choose one of its neighbors, node j, uniformly randomly. Send a message to node j (Fig.2(a)).
2. Node j sends back to i the set L of the addresses of j's neighbors (Fig.2(b)).
3. Choose $\min\{f, |L|\}$ nodes $\{k_1, k_2, \ldots, k_{\min\{f,|L|\}}\}$ in L that are not neighbors of i, uniformly randomly. If no such node exists in L, then stop. Otherwise, measure the cost from i to each chosen node $k' \in \{k_1, k_2, \ldots, k_{\min\{f,|L|\}}\}$ and evaluate the cost resulted from replacing link (i, j) with link (i, k'). The cost is the difference of global costs between these topologies, which is
$$\Delta C \equiv c(i, k') - c(i, j) + 2w(d_{k'} - d_j + 1).$$
4. Choose the node, say node k, that leads to the minimum ΔC. If the minimum $\Delta C \geq 0$, then reject this transformation; otherwise, perform it as follows: Send a message to k asking it to establish a connection with i. If k accepts this request, then establish the link between i and k, and remove the link between i and j (Fig.2(c)).

Note that the node i, which is the initiator of this transformation, does not lessen its benefits in this transformation, since it simply replaces a high-cost link with a new link which is lower cost. Also, node k can reject to establish the link between i and k if it expects that the new connection would incur additional cost. Thus our algorithm does not impose on the nodes the burden of behaving in a self-sacrificing fashion. As stated in the first section, this is in contrast to LOCALISER, in which the node that initiates a modification action must bear cost of disconnecting a link to a neighbor node [5].

It should also be noted that the algorithm makes no changes that would increase the global cost. This is subtle but important difference from LOCALISER. LOCALISER is a Metropolis algorithm, which permits the system to transit a worse state with a certain probability to avoid to get stuck at local minima of the cost function.

Since our algorithm does not allow such transitions, one might think that our algorithm would cause the system to converge to undesirable local optima. From our preliminary experiments, however, we found that this is not the case.

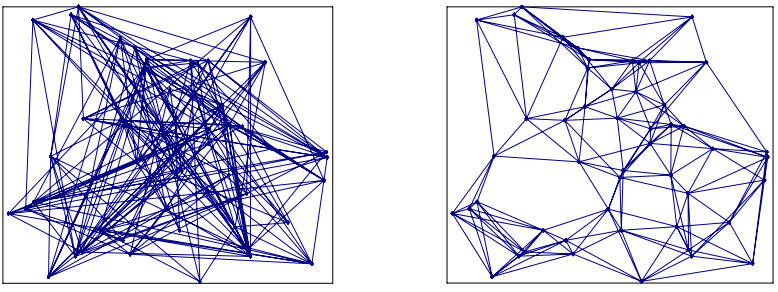


Fig. 3. Topologies of overlay networks: (a) the initial network; (b) the network yielded by the proposed algorithm

In the preliminary experiments, we repeated many runs of the algorithm while varying the initial networks and the probability of accepting a bad transition; but we found no clear case where accepting bad transitions caused a better topology or faster convergence. Our conjectured reason for this finding is the distributed nature of the algorithm; in the algorithm every node can have a chance to initiate a topology change, which means that there are always many candidate transformations that are checked in Step 4, so are there acceptable transformations.

3.2 An Illustrative Example

Here we describe an illustrative example of 50 nodes placed uniformly at random on a two-dimensional space. In this example, the communication cost between two nodes is defined as the Euclidean distance between them on the two-dimensional space. Fig. 3(a) depicts an example of an unstructured overlay which is not optimized. The proposed algorithm refines this initial overlay by make each node modify its connections so as to favor near nodes as its neighbors. Fig. 3(b) shows the topology that was obtained by repeating such a local modification sufficiently many times. Note that the number of edges is the same in Figs. 3(a) and 3(b). Hence the apparent sparsity of the latter network is due to the edges being much shorter on average.

4 Experimental Results

In this section, we show the results of our experiments. We first constructed the initial networks of 100 nodes and 200 nodes. We adopted the Barabasi-Albert scale-free network model [9]; we started with a small number m_0 nodes, at every

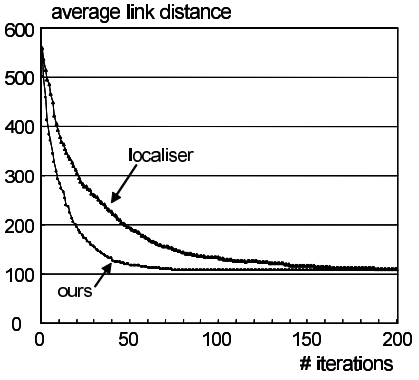


Fig. 4. Average distance of the links in the overlay as a function of number of iterations

time step we added a new node with $m(\leq m_0)$ links that connect the new node to m different nodes already present in the network. The probability Π that a new node will be connected to node i depends on the connectivity d_i of that node, so that $\Pi(d_i) = d_i / \sum_i d_i$. We set $m_0 = 3$ and $m = 3$.

The communication cost $c(i, j)$ between two neighboring nodes i, j was set to the Euclidean distance between them on a 1000×1000 square where all nodes are uniformly randomly placed.

We simulated the behaviors of our proposed algorithm ($f = 5$) and the LOCALISER algorithm [1] when they were applied to these two initial topologies. We set w in the cost function $C(G)$ to 20. In doing so, we assumed that all nodes are loosely synchronized and execute each iteration of these algorithms in rounds. We also assumed that a node does not reject the request for establishing a link in Step 4 in our algorithm.

4.1 Link Distance

To evaluate the performance of our algorithm in reflecting the underlying physical network topology, we measured the average communication cost between two end nodes of a link in the overlay.

Fig. 4 presents how the average communication cost varied as the time elapsed in the network of size 100. One can clearly see that the average cost gradually decreases as the number of iterations executed increases and that our algorithm is faster to converge than LOCALISER. Almost the same curves were obtained for the network of size 200.

Table 1 summarizes the average link distance when the algorithms were iterated 50, 100, and 200 times, together with the initial values. These results demonstrate that our algorithm is more effective than the LOCALISER algorithm in capturing the underlying network topology.

[1] The design parameter T was set to 1.

Table 1. Average distance of the links in the overlay

network	100 nodes				200 nodes			
iteration	0	50	100	200	0	50	100	200
localiser	558	193	132	110	541	187	110	78
ours	558	118	108	107	541	97	76	73

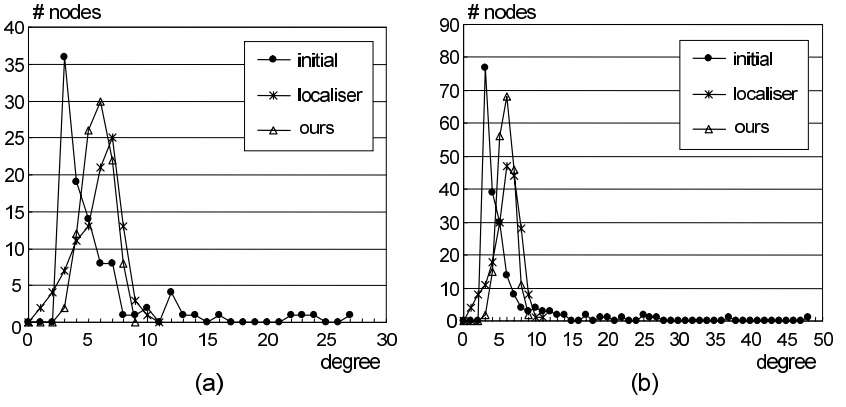


Fig. 5. Distribution of node degrees

4.2 Node Degree

Here we show how the proposed algorithm affects the node degree. Figs. 5(a) and 5(b) depict the distributions of the degree for the networks obtained just after 50 iterations. The results shown in Figs. 5(a) and 5(b) are those obtained for the networks of size 100 and 200, respectively. The horizontal axes represent the node degree, while the vertical axes denote how many nodes have that degree. From these graphs, one can see that the distribution becomes sharply concentrated around the average value when our algorithm or LOCALISER is used. In the initial network, a few nodes have very high connectivity, and the other most nodes have only few links. Intuitively speaking, this helps decrease the sensitivity to failures by balancing the reliance evenly on all the nodes. The resiliency to failures is more directly evaluated in the following subsection.

4.3 Resilience

To address the resilience of the networks, we study the probability that the network is fragmented into isolated subnetworks when some fraction of the nodes have failed. Here we consider the six topologies: the two initial topologies of sizes 100 and 200; the two topologies yielded by our algorithm; the two topologies yielded by LOCALISER. We performed 50 iterations for each algorithm.

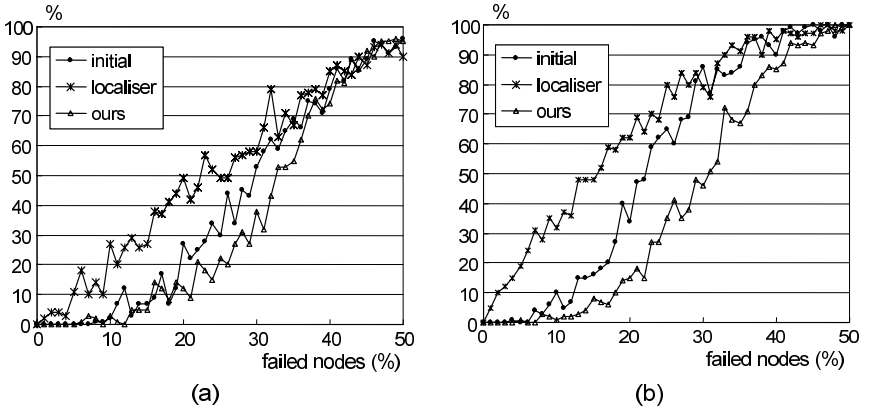


Fig. 6. Network fragmentation probability in the face of node failures

We selected 100 random patterns of failed nodes for each different percentage of failed nodes, and measured how many times that network fragmentation occurred for each of these topologies.

The results of these experiments are depicted in Fig. 6. Figs. 6(a) and 6(b) display the results for the networks of size 100 and 200, respectively. In these graphs, the horizontal axes represent the percentage of failed nodes, while the vertical axes denote the mean percentage that correct nodes are isolated into two or more partitioned network fragments.

As clearly seen in these graphs, in a large range of percentage of failed nodes, the robustness to node failures was significantly improved, by applying our algorithm. For example, if failed nodes were less than 10%, almost no fragmentation occurred in the topologies optimized by our algorithm. On the other hand, network fragmentation was observed for the initial networks and those yielded by LOCALISER with a greater likelihood.

5 Conclusions

In this paper, we have proposed an algorithm which optimizes unstructured large-scale overlay networks by reshaping the overlay via iterative refinements. The proposed algorithm helps the overlay reflect the underlying network topology while maintaining the resiliency to failures. The algorithm is fully decentralized and only relies on local information available at each node. We present the results of simulation which demonstrate the effectiveness of the algorithm. Future work includes the study of the algorithm in more practical settings. Specifically, we consider using the GT transit-stub model [11], as a model of underlying networks.

Acknowledgments

This research was supported in part by "Priority Assistance for the Formation of Worldwide Renowned Centers of Research — The 21st Century Center of Excellence Program" of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. Ripeanu, M., Foster, I., Iamnitchi, A.: Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Computing Journal **6** (2002)
2. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA (2001) 161–172
3. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: Proceedings of the 2001 ACM SIGCOMM Conference. (2001) 149–160
4. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany (2001)
5. Massoulié, L., Kermarrec, A.M., Ganesh, A.J.: Network awareness and failure resilience in self-organising overlay networks. In: Proceedings of the 22nd IEEE Symposium on Reliable Distributed Systems (SRDS '03). (2003) 47–55
6. Xu, Z., Tang, C., Zhang, Z.: Building topology-aware overlays using global soft-state. In: ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems, Washington, DC, USA (2003) 500–508
7. Liu, Y., Zhuang, Z., Xiao, L., Ni, L.M.: A distributed approach to solving overlay mismatching problem. In: 24th International Conference on Distributed Computing Systems (ICDCS 2004), Tokyo, Japan, IEEE (2004) 132–139
8. Ren, S., Guo, L., Jiang, S., Zhang, X.: Sat-match: A self-adaptive topology matching method to achieve low lookup latency in structured p2p overlay networks. In: 18th International Parallel and Distributed Processing Symposium (IPDPS 2004). (2004)
9. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
10. Ganesh, A.J., Kermarrec, A.M., Massoulié, L.: Peer-to-peer membership management for gossip-based protocols. IEEE Transactions on Computers **52** (2003) 139–149
11. Zegura, E.W., Calvert, K.L., Bhattacharjee, S.: How to model an internetwork. In: IEEE Infocom. Volume 2., San Francisco, CA, IEEE (1996) 594–602

BACS: Split Channel Based Overlay Multicast for Multimedia Streaming*

Joongsoo Lee, Xuan Tung Hoang, and Younghee Lee

Computer Networks Lab.
Information and Communications University
119 Munjiro, Yuseong-gu, Daejeon, Korea
{jslee, tung_hx, yhlee}@icu.ac.kr

Abstract. Multipath multicast has focused on how to deal with bandwidth instability and unfairness in forwarding overhead. Creating multiple disjoint trees is good for applied applications requiring aggregated throughput, such as content distribution. However, the path heterogeneity of different trees may cause data asynchrony in the receiver's view, making it difficult to use in real time applications. In this paper, we propose a new delivery structure named *cluster tree* that utilizes bandwidth efficiently and lessens asynchrous sub-stream arrival. Cluster is composed of the interconnection of nodes within a latency boundary to each other, and the parent-child relationship between clusters forms a tree. Members of a cluster exchange disjoint sub-streams with peers in the same cluster and adapt to network dynamics cooperatively. This rate control mechanism can adapt to bandwidth fluctuation. The simulation result shows that cluster tree increases effective packets and reduces average source-to-leaf latency.

1 Introduction

There has been much work in recent years on the topic of P2P systems [1,2] and overlay multicast [3,4,5,6,7,8]. The effective building of application level networking frameworks leverages efficient group communication such as contents delivery network, multimedia streaming, and contents sharing. The instability of bandwidth becomes a challenging issue for real time applications such as multimedia streaming and video conferencing. Multipath [7, 9, 10] and path diversity [11] are two approaches aiming to solve path stabilization. SplitStream [9] and distributed k-MST [10] propose algorithms make multiple disjoint trees (MDTs).

The performance metric of multimedia streaming is not only bandwidth but also effective packets. We define effective packets as packets that arrive within a timing constraint. In the viewpoint of effective packets, packets arriving after the time limit consume bandwidth for nothing. MDT schemes make sub-streams

* This work is supported in part by Grant No. A1100-0502-0077 from MIC(Ministry of Information and Communication) of Korea and Grant No. R01-2003-000-10562-0 from Korea Science and Engineering Foundation.

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 965–974, 2006.

travel heterogeneous paths and asynchronous arrival is not focused on much. Disjoint trees are constructed without having a relationship with each other and so do not accommodate asynchronous data delivery. We argue that it is possible to reduce the effects of asynchrony by making latency based grouping without sacrificing bandwidth utilization.

We propose a new delivery structure called *cluster tree*, which means the parent-child relationship is formed not between nodes but between clusters. A cluster receives a single degree of stream, where divided sub-streams are transported to each member and the whole stream is restored after exchanging disjoint sub-streams among cluster members. Sub-streams in cluster tree have a stronger relation than those in MDTs. Members of a cluster have the responsibility to adapt network dynamics in a cooperative way. Unlike traditional trees, the members of the parent cluster can send packets to any of members of the child cluster and the sending rate is controlled by *collective rate control*. The collective rate control scheme maintains the bandwidth usage from a parent cluster to a child cluster as one degree of stream, and controls the rate from each member of a parent cluster to each member of a child cluster dynamically. Asynchronous data arrival can be reduced using careful clustering. Nodes that have small latency to each other form a cluster and have a high possibility of being close in a physical network. To build a cluster, only nodes that pass a latency bound test can become members of a cluster. Therefore, the members of a cluster have relatively homogeneous latency distribution.

The remainder of this paper is organized as follows. Section 2 summarizes related work and Section 3 presents an introduction about the notion of effective packets. A detailed description of proposed scheme is presented in Section 4. Methods used to evaluate the performance are described in Section 5, and our conclusions are drawn in Section 6.

2 Related Work

Many previous works mention a stress, which is the degree of same packets traveling across a link, as an important performance metric. Since overlay multicast nodes duplicate incoming packets for data delivery, a link between the nodes suffers as many times the load as the number of children. Several works have been proposed to enhance bandwidth utilization and share the carrier nodes' overhead fairly. CoopNet [12] provides a load share mechanism between a server and clients. It uses a centralized algorithm running at the server to build the trees. Bullet [7] and SplitStream [9] are somewhat related with this paper. Bullet targets high bandwidth data dissemination on the mesh. It proposes a mechanism to discover disjoint data. SplitStream aims at fair share of forwarding loads among all the members of a multicast session. To deliver a stream, it uses multiple disjoint trees. Splitting a stream into multiple sub-streams can leverage bandwidth utilization effectively. Both Bullet and SplitStream, however, do not impose the effects of asynchrony caused by their split channel strategy when they construct delivery structures. Anthony Young et al. [10] designed and

implemented a system that constructs k minimum spanning tree in a distributed manner. It forms a good quality mesh to support high bandwidth applications. The advantage of the k-MST algorithm over SplitStream is that it helps to route in multipath more efficiently. Still, the asynchrony issue is not addressed.

PDF [11] uses a redundant path and forward error correction coding (FEC) to reduce packet loss rate. Traceroute is used to find a redundant path which shares minimum links with default path. This path diversity scheme is very useful for streaming from servers to clients on overlay networks, but it is not applicable to multicast because it only focuses on the path between sender and receiver.

3 Bandwidth Utilization and Effective Packets

A packet delivery path between two ends consists of hops of routers and the end-to-end available bandwidth is characterized by a bottleneck link. When $u_i^\tau(t_0)$ is the average utilization of link i, with $0 \leq u_i^\tau(t_0) \leq 1$, Manish et al. [13] defines the available end-to-end bandwidth $A^\tau(t_0)$ during a time interval $(t_0, t_0 + \tau)$ as follows:

$$A^\tau(t_0) = \min_{i=1\ldots H} \{C_i \left[1 - u_i^\tau(t_0)\right]\} \tag{1}$$

As depicted in Fig. 1, suppose that packets start with the rate R_0 from the source and the packet forwarding rate of the intermediate router i is R_i. Packets arrived at the buffer are interposed by cross traffic, so that the amount of overhead of router i decides the outgoing rate at each hop. Bandwidth occupation of the cross traffic B_i^c affects R_i by sharing the capacity of router i. We assume bandwidth is fairly shared according to the amount of packets injected when the capacity is smaller than required. Within this context, the outgoing rate of stream from node i is shown in (2).

$$R_i = \min\left(R_{i-1}, \frac{C_i R_{i-1}}{B_i^c + R_{i-1}}\right) \tag{2}$$

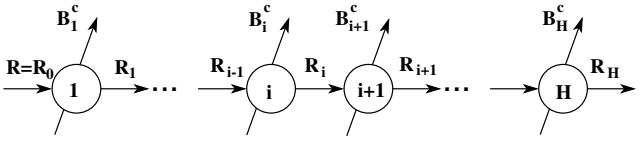


Fig. 1. End-to-end bandwidth is characterized by the bottleneck link along the path. B_i^c is the cross traffic at node i and makes the bandwidth less available.

Higher quality stream may fluctuate more when cross traffic is injected. Now we formulate the effects of stream size and cross traffic to quality of video using the ratio of successfully delivered packets, called *effective packets*. Multimedia streaming is dependent on timing constraints and therefore only packets arrived

within the requirements – such as the buffering time limit – contribute to the presentation. The timing constraint is denoted as τ'. Available bandwidth is somewhat dynamic and the varying cross traffic interferes with packet arrival at the receiver. If the bandwidth share of the stream at time t_0 is R_i and additional cross traffic $\triangle B_i^c$ travels router i, then *effective packets* $E^{\tau'}(t_0)$ arrived during τ' and *ineffective packets* $R_{i-1}\tau' - E^{\tau'}(t_0)$ are shown in (3).

$$E^{\tau'}(t_0) = \frac{C_i R_{i-1}}{B_i^c + \Delta B_i^c + R_{i-1}} \times \tau'$$

$$R_{i-1}\tau' - E^{\tau'}(t_0) = \frac{R_{i-1}\tau'(B_i^c + \Delta B_i^c + R_{i-1} - C_i)}{B_i^c + \Delta B_i^c + R_{i-1}} \tag{3}$$

Equation (4) is the ratio of effective packets to the original packets sent. Because we assumed that end-to-end bandwidth share is characterized by R_i, R_{i-1} is not that much different from R_0. Equation (4) shows that effective packets decreases when sending rate increases, therefore the higher bit rate stream is more sensitive to cross traffic than the lower one. We can reduce the number of ineffective packets by forcing sub-streams of the original stream to travel a different path.

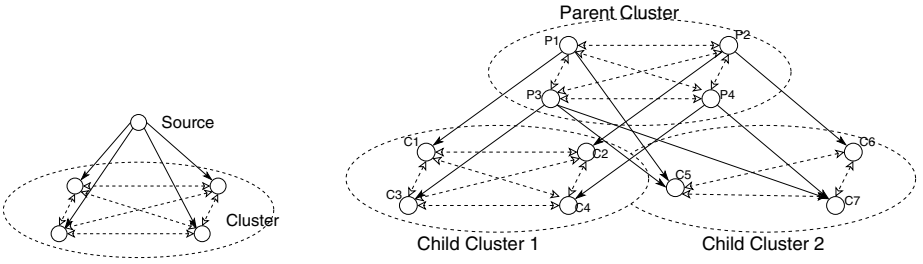
$$\frac{E^{\tau'}(t_0)}{R_0\tau'} = \frac{C_i}{B_i^c + \Delta B_i^c + R_{i-1}} \quad (when \quad R_{i-1} \approx R_0) \tag{4}$$

Multipath multicast using sub-streams has initially smaller R_0 and results in an increase of effective packets ratio. Sub-streams flowing through different paths may meet the bandwidth requirement but the arrival time of each sub-stream varies according to path quality, latency and hops. The performance of multiple disjoint paths is not good enough in the view of effective packets even though the aggregated throughput may look good. To address this problem we use a latency based grouping called clusters. When constructing a forwarding tree each node becomes a member of a cluster after testing whether it has relatively homogeneous latency to other members. The details of topology construction are presented in the following section.

4 Cluster Based Delivery Structure

Figure 2 shows the proposed delivery structure. Clusters connected with multiple links make parent and child relationships, and each link delivers a different sub-stream. All the members of a child cluster can make the original stream by exchanging their own sub-stream. For example, a child node c_1 receives three sub-streams s_2, s_3, and s_4 through c_2, c_3, and c_4 respectively, and then places packets into correct positions so that the original stream is restored when $\sum_{i=1,\ldots,4} s_i = S$ is satisfied.

A node forwards only packets delivered from its parent nodes. Forwarding packets from siblings to lower levels may propagate more delay as the depth of



(a) From the source to first level cluster.

(b) Cluster tree structure from second level cluster to lower levels.

Fig. 2. Cluster tree. The source sends packets to its child cluster and the child clusters also forward packets received from the source. If the source node has enough bandwidth it can adopt multiple child clusters. Each node in a cluster receives different streams and it merges whole stream by exchanging an other sibling's stream.

tree increases and child clusters suffer more data asynchrony as a result. A child node can receive from any node in the parent cluster and the sending rate is controlled by the report of child nodes.

Nodes of the parent cluster that do not send packets directly are called parents in law in contrast to real parents. In Fig. 2-(b) , P1 is a parent in law to C2, C3, and C4. Parent nodes in law also affect children in law by the rate control scheme, which is described in Sect. 4.2. In Fig. 2-(b) each member of a parent cluster sends sub-streams to each of its child nodes as an assigned sending rate (depicted as solid lines). The dashed lines among cluster members show the sub-stream exchange.

4.1 Tree Construction

Leader selection. A cluster leader is a representative node to measure latency before accepting a new node. In the selection process, the largest delay to cluster (LDC) is used as a metric. The LDC of a node stands for the maximum value among delays between the current node to other members in the cluster. The node that has the smallest LDC is logically at the center of the cluster. Measuring the latency between the new node and a cluster leader can reduce the number of latency measurements. Otherwise, the new node should measure latencies to all the nodes in a tree. When a node joins a cluster, the LDC of the new one is compared to that of the leader and the smaller one is marked as a leader. When a leader is changed it is reported to both the parent and the child cluster.

Join process. At first, a new-comer sends a join request to the root, then is informed of a list of random k leaders. The cluster leaders measure the latency to the new node and report it whether the delay is acceptable or not. For admission, the leader compares the latency with the latency boundary (LB) asserted by the session or by the application. LB defines the jitter boundary in consideration of the buffer size and/or application characteristics.

A cluster leader gives acceptance for test (AT) only if a) the cluster size is smaller than cluster size limit k; and b) latency between the new-comer and the leader is less than the LB. On receiving AT, a node adds the leader onto the test list. For clusters of which leaders are on the test list, the node performs an LB test for all members of the current cluster. If the node has latencies less than the LB, it receives an acceptance notification (AN). If a node receives multiple ATs, it joins the cluster with the LDC.

If no clusters inform AT or AN to a node, then it makes a cluster by itself and becomes a child of the bottom-most cluster. If nobody joins a single node's cluster, the node performs a join trial after waiting random amount of time.

4.2 Adaptation for Network Status

Many possible scenarios such as congestion, link failure, and multicast node failure, can change network status. Optimization of a tree is a hard problem and even worse in the proposed cluster tree scheme. If nodes move up and down to find the optimal position as in the case of Overcast [6], these moving nodes may affect many configurations of parent and child clusters. To avoid this complex operation, collective rate control is the first step of adaptation to network dynamics instead of optimization.

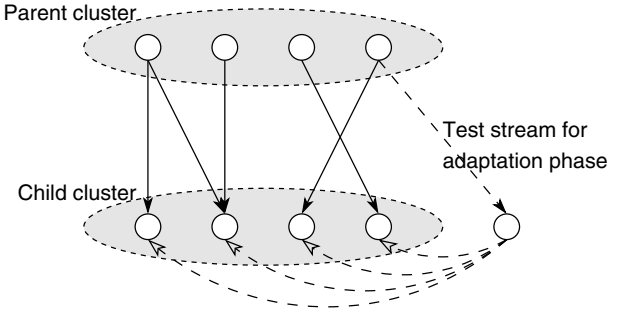


Fig. 3. After passing an adaptation phase, a node can join the cluster and operate in a delivery phase

Any nodes in a parent cluster can send a sub-stream to any member of a child cluster. If a node joins a cluster, the new node operates in the adaptation phase. A parent node who has available out-degree sends test packets for adaptation phase. The sending rate to the new child is resolved after several rounds of tests by linearly increasing the rate. Child nodes report the quality of path via the new node. If the path quality is not satisfactory, the two rounds before the failed round is selected for the test stream. This is the maximum bandwidth the tester parent node can send but the overall rate is controlled cooperatively. For example, one of the controlling policies is a ratio of sending rate to maximum bandwidth. By controlling this ratio fairly, each path can have less fluctuation caused by cross traffic.

Even in the delivery phase, the path quality may enter an unsatisfactory state. Child nodes periodically report the receipt of packets arrived and the parent nodes use them as a reference in the decision of adaptation. If a child node receives fewer packets than sent, the parent node regards bandwidth as insufficient. In this case, a collective rate control tries to compensate for the low quality path. A path from a parent node to a child node may come to have a lower bandwidth than it had in adaptation phase. In this case, the other nodes need to send more packets than it was originally assigned. A child node who has the largest maximum bandwidth from parent nodes is selected as a candidate. This node is tested in the adaptation phase only for the additional rate. For example, node p_1 is assigned to send 100Kbps to the child node c_1 and node p_2 is assigned to send 80Kbps to the child node c_2, where p_1 and p_2 are members of the parent cluster, and c_1 and c_2 are members of the child cluster. If available bandwidth of the path from p_2 to c_2 becomes 60Kbps, then node p_1 needs to send an additional 20Kbps stream to c_1. The adaptaion phase tests the path from p_1 to c_1 for the additional 20Kbps stream.

In some cases, rate control cannot enhance path quality. If a node's LDC goes higher than LB, the members of the cluster may experience jitter problem. The cluster excludes the jittering node if the exponential moving average of LDC goes higher than LB.

5 Performance Evaluation

In order to evaluate the differences between cluster tree, single tree and multipath multicast, we performed a simulation using NS-2, and transit-Stub topology is generated using GT-ITM [14]. We generated 1,000 node networks and picked up multicast nodes randomly up to 300.

We constructed the tree for the simulation as follows.

- Single tree: A minimum spanning tree is built using latency metric in advance.
- Multipath tree: Three disjoint trees are used for this measurement. One tree is a minimum spanning tree but the other two are constructed one by one using several parameters. To maintain the fairness of tree depth, the average number of children is decided by the minimum spanning tree. By constructing a non-minimum spanning tree, a node picks up randomly from the closet 20 nodes.
- Cluster tree: Nodes join the multicast according to the sequence randomly scheduled. The joined node is positioned by the implementation of the proposed scheme. After the joining process finishes, the monitoring starts. In this simulation, the maximum number of cluster members is set to 4.

Figure 4 shows the effective packets ratio. A buffer size determines whether a packet is effective or not. The buffer size is calculated by delay bandwidth product. The average delay of links are used for delay bandwidth product. The detouring packets through disjoint tree made the multipath multicast the poorest. As the number of nodes increase, the single tree's performances decrease.

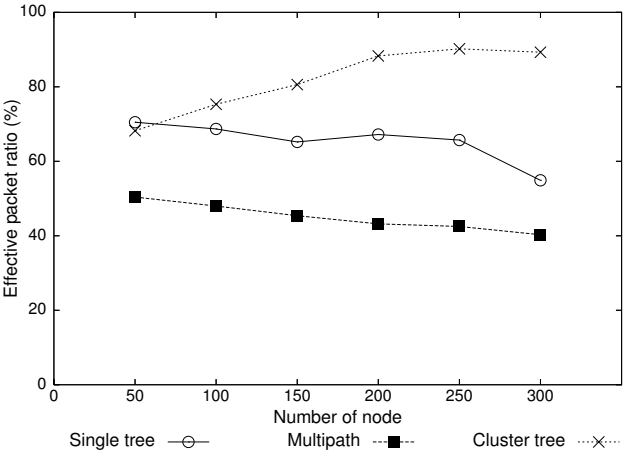


Fig. 4. Effective packets ratio of single tree, multipath, and cluster tree multicast

Here the performance bottleneck is the multicast traffic itself. Increasing numbers of multicast nodes may incur more traffic and eventually degrade the performance. The cluster tree, however, becomes better as the membership enhances because more members are given more opportunities to make good clusters. Good clustering may induce packet localization and it is also one of the reasons why cluster tree performs well.

Figure 5 shows the average latency monitored at leaf nodes. Here the latency means the delay from a source and the leaf receivers. The overall trend increases as the number of nodes increase. The cluster tree outperforms because its average

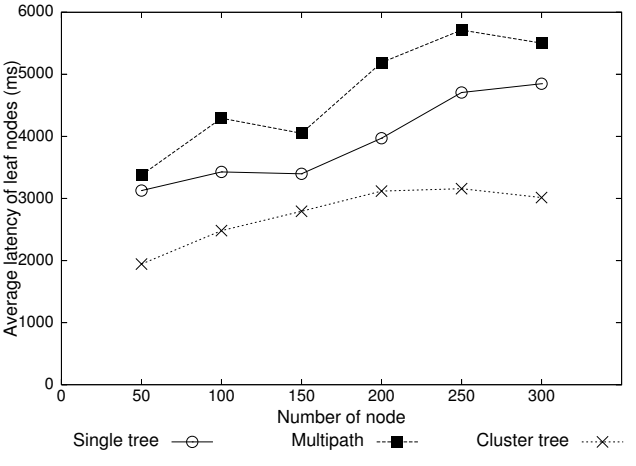


Fig. 5. Average latency at leaf nodes

depth is much smaller than the other two. Clustering allows more nodes to be positioned at the same depth. The multipath scheme suffers larger delay because of detouring paths.

6 Conclusion

We proposed a novel delivery structure for multimedia streaming that considers both bandwidth management and the asynchrony caused by splitting a trunk stream into several sub-streams. The cluster tree adapts to dynamic network status using a collective rate control scheme. Our results show that the cluster tree outperformed in terms of effective packets and average source-to-leaf latency. Even though the density increases, packet localization through clustering may save global resources. We tried to make reasonable assumptions while performing our simulation. But in spite of our efforts, there exists the limitation of simulation environments, such as cross traffic. We are preparing a performance evaluation on PlnaetLab [15] to analyze how the cluster tree perfoms in real environments.

References

1. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2001) 161–172
2. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2001) 149–160
3. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. In: SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2002) 205–217
4. Castro, M., Druschel, P., Kermarrec, A., Rowstron, A.: SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications (JSAC) **20**(8) (2002) 1489–1499
5. Chu, Y.H., Rao, S.G., Zhang, H.: A case for end system multicast. In: Proceedings of SIGMETRICS. (2000) 1–12
6. Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, M.F., Jr, J.O.: Overcast: Reliable multicasting with an overlay network. In: OSDI. (2000) 197–212
7. Kostić, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: high bandwidth data dissemination using an overlay mesh. In: Proceedings of the nineteenth ACM symposium on Operating systems principles. Volume 37, 5 of Operating Systems Review., New York, ACM Press (2003) 282–297
8. Rowstron, A.I.T., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Middleware. (2001) 329–350
9. Castro, M., Druschel, P., Kermarrec, A.M., Nandi, A., Rowstron, A.I.T., Singh, A.: Splitstream: High-bandwidth content distribution in cooperative environments. In: IPTPS. (2003) 292–303

10. Young, A., Chen, J., Ma, Z., Krishnamurthy, A., Peterson, L.L., Wang, R.: Overlay mesh construction using interleaved spanning trees. In: Proceedings of IEEE INFOCOM. Volume 1. (2004) 396–407
11. Nguyen, T., Zakhor, A.: Path diversity with forward error correction (pdf) system for packet switched networks. In: Proceedings of IEEE INFOCOM. Volume 1. (2003) 663–672
12. Padmanabhan, V.N., Wang, H.J., Chou, P.A., Sripanidkulchai, K.: Distributing streaming media content using cooperative networking. In: Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV-02), New York, ACM Press (2002) 177–186
13. Jain, M., Dovrolis, C.: End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. In: SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2002) 295–308
14. Zegura, E.W., Calvert, K.L., Bhattacharjee, S.: How to model an internetwork. In: Proceedings of IEEE INFOCOM. (1996) 594–602
15. PlanetLab: (http://www.planet-lab.org)

Heterogeneity Aware P2P Algorithm by Using Mobile nodeID

Kyungbaek Kim and Daeyeon Park

Department of Electrical Engineering & Computer Science,
Division of Electrical Engineering,
Korea Advanced Institute of Science and Technology (KAIST),
373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
kbkim@sslab.kaist.ac.kr, daeyeon@ee.kaist.ac.kr

Abstract. The peer-to-peer systems have become an extremely popular platform for large-scale content sharing. A lot of research papers discussed the Distributed Hash Table (DHT) based p2p algorithms to promise that idle resources may be efficiently harvested. However, p2p systems are composed of components with extremely heterogeneous availabilities and for nodes which join/leave the system frequently, the system will generate a lot of information maintenance traffic such as routing information update traffic and data copy traffic to keep the efficiency of the DHT based p2p algorithms.

In this paper, we suggest the mobile nodeID based p2p algorithm to reduce the overhead by exploiting the heterogeneity of participant nodes efficiently. Unlike the DHT based p2p algorithms, the nodeID of a node changes according to its characteristic to support the p2p system efficiency and each nodes takes the different responsibility in accordance with its nodeID. We classify nodes into the two types according to the characteristics of nodes : the reliable nodes and the leaf nodes. The reliable node which is the more stable and more reliable node acts as the more important role of the routing and the replication. The leaf node which joins/leaves very frequently acts as the simple role to minimize the information maintenance traffic. The reliable node has the load-balanced ID to balance the loads and the leaf node has the load-free ID to reduce the responsibility.

We examine the efficiency of our p2p algorithm via a event driven simulation and show that the information maintenance traffic reduces and the routing process is more efficient.

Keywords: peer-to-peer, algorithm, mobile nodeID, heterogeneity.

1 Introduction

In these days, peer-to-peer systems have become an extremely popular platform for large-scale content sharing. Unlike client/server model based storage systems, which centralized the management of data in a few highly reliable servers, peer-to-peer storage systems distribute the burden of data storage and communications among tens of thousands of clients. The wide-spread attraction of this model arises from the promise that idle resources may be efficiently harvested to provide scalable storage services. A lot of research papers discussed the Distributed Hash Table (DHT) based p2p routing algorithms (Chord, Pastry, Tapestry and CAN) [2][3][4][5].

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 975–984, 2006.

In contrast to traditional systems, peer-to-peer systems are composed of components with extremely heterogeneous availabilities - individually administered host PCs may be turned on and off, join and leave the system and have intermittent connectivity, and are constructed from low-cost low reliability components. For example, one recent study[6] of a popular peer-to-peer file sharing system found that the majority of peers had application-level availability rates of under 20 percent and only 20 percent nodes have server-like profiles. In such an environment, failure is no longer an exceptional event, but is a pervasive condition. At any point in time the majority of hosts in the system are unavailable and those hosts that are available may soon stop servicing requests.

A big issue in current DHT based p2p algorithms is the high overhead of maintaining DHT routing data structure and the stored data. When a node joins/leaves the system, the affected routing data structure on some existing nodes must be updated accordingly to reflect the change. Moreover, most p2p systems employ some form of the data redundancy to cope with failure and when the membership of nodes changes, these systems generate huge overhead of compulsory copies for the data availability. Especially, for nodes which join/leave the systems frequently, the p2p system will generate a lot of routing information update traffic and data copy traffic. It does not only increase the consumption of the network bandwidth, but also affects the efficiency of DHT based routing algorithms. Until now, DHT algorithms are not widely used in commercial systems yet, most p2p file sharing systems are still using non structured p2p mechanisms.

In this paper, we suggest the mobile nodeID based p2p algorithm to reduce the information maintenance overhead by exploiting the heterogeneity of participant nodes efficiently. Unlike the DHT based p2p algorithms, the nodeID of a node changes according to its characteristic to support the p2p system efficiency and each node takes the different responsibility in accordance with its nodeID. We classify nodes into the two types according to the characteristics of nodes : the reliable nodes and the leaf nodes. The reliable node which is the more stable and more reliable node acts as the more important role of the routing and the replication. The leaf node which joins/leaves very frequently acts as the simple role to minimize the information maintenance traffic. The reliable node has the load-balanced ID to balance the loads and the leaf nodes has the load-free ID to reduce the responsibility.

The reliable nodes are more stable and more reliable nodes and these nodes act as more important roles such as routing and replication. The reliable nodes have Load Balanced ID (LBID) which is evenly distributed and balances the workload of each reliable node. This LBID is dynamically assigned and the LBID routing table which helps for routing to any reliable nodes is also organized when the LBID is assigned. The leaf nodes join/leave very frequently on the system and the majority of the participant nodes are these leaf nodes. These nodes act as simple roles such as servicing the request and helping the reliable nodes. The leaf nodes have Load Free ID (LFID) which makes the ID region of a leaf node as small as possible and reduces the effect of the dynamic membership change which increases the information maintenance overhead. According to these basic behaviors, because the frequently joining/leaving of nodes occurs as the leaf nodes which make little overhead, we can reduce the overhead of the whole p2p system and achieve more efficient routing without the frequent updates.

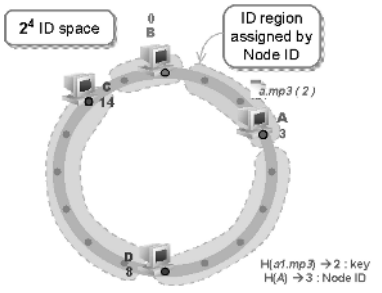


Fig. 1. Overview for the general DHT based P2P algorithm

This paper is organized as follow. In section 2, we describe the DHT based p2p algorithm and its problem. Section 3 introduces the detail of the mobile nodeID based p2p algorithm. The simulation environment and performance evaluation are given in section 4. Finally conclude in section 5.

2 Background

There are many DHT based p2p algorithms such as Chord, Pastry, Tapestry and CAN [2][3][4][5]. Each node has a DHT which is a small routing table and any node can be reached in about $O(logN)$ routing hops where the N is the total number of nodes in the system. To achieve this efficient and bounded routing, there are some rules for the organizing the participant nodes. First of all, each node has a unique nodeID which is taken by hashing any identifier of a node, and according to its nodeID it maps on the ID space where the nodes and the objects are co-located with the nodeIDs or the keys which are the hashed values of the nodes or the objects. In the figure 1, the node id of node A is 3 and it maps on the position for 3. After the mapping of the node id, each node knows its ID region from the next position of its previous nodeID to its nodeID and each node should store and service the objects for its ID region. In the figure 1, node A takes its ID region from 1 to 3 because its node id is 3 and its previous node B locates on 0 and when a node wants to get a.mp3 whose key is 2, node A gets the request for it.

Though these well-organized rules make the routing of the p2p system efficient and bounded, a big issue in current DHT based p2p algorithms is the high information maintenance overhead of maintaining DHT routing data structure and the stored data. Because its node id is already given by the hashing function and its position on ID space is already fixed, when a node joins/leaves the p2p system, the ID region of its neighbor nodes changes and the stored data should be copies for the new ID region to service the right and reliable object, and the update of the routing table is also needed. In figure 1, if node A leaves, the ID region of node D changes and the object from 1 to 3 should be copied from node A to node D. Moreover, the affected routing table which has the entry with node A must be updated. In this case, one recent research[6] of a popular p2p file sharing system found that 80 percent of total nodes of a p2p system join/leave very

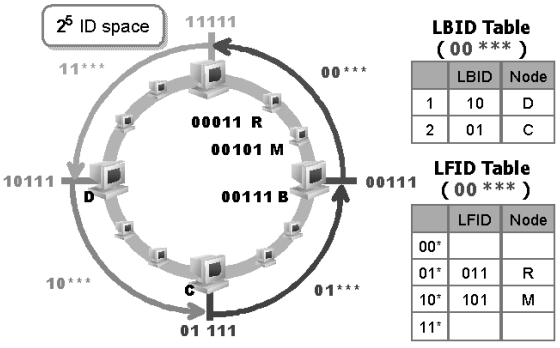


Fig. 2. Overview of Mobile nodeID based P2P Algorithm

frequently and the majority of nodes have the application-level availability rate of under 20 percent. In such an environment, the information maintenance overhead is getting worse and this overhead discourages that the DHT based p2p systems are deployed to the real world.

3 Mobile nodeID Based P2P Algorithm

3.1 Overview

Previous DHT based p2p algorithms lack the explicit methods for exploiting the heterogeneous characteristics of participant nodes. The main reason of this lack is the static nodeID which makes the location of a node fixed on the ID space, and the system with the static nodeID is not flexible. We address this problem with the mobile nodeID which changes according to the characteristics of a node. We classify the participant nodes into two types : reliable nodes and leaf nodes. The reliable nodes are more reliable and more stable nodes and the leaf nodes join/leave very frequently. The nodeID of a reliable node is well distributed on the ID space and makes that the each reliable node gets fair ID region and balanced loads. The leaf node gets the node id which makes its ID region as small as possible to minimize the information maintenance overhead for joining/leaving of it.

Figure 2 shows the overview of the p2p system that uses the mobile nodeID based p2p algorithm. The participant nodes are on the 2^5 ID space and the number of bits for a nodeID is 5. The nodeID is consist of the *Load-Balanced ID (LBID)* and the *Load-Free ID (LFID)* and ,in this example, the first 2 bits of a nodeID mean the LBID and the other 3 bits are for the LFID. In this figure, the large sized computer means the reliable node and the small sized computer means the leaf node. To distribute the participant nodes efficiently, we divide the ID space into many *sub-regions* which are the balanced ID regions. Each sub-region has one reliable node which represents this region and many leaf nodes which assist the reliable node. That is, the reliable node is mainly responsible for the objects for the sub-region and the leaf nodes service the objects for the small ID region which is assigned by both of their node id and the LFID

```
If No Reliable Node
   LBIDnew = set all bits of LBID to 1
   Level = 1
Else
   If( Join ){
      If( Level < Levelthres ){
         // Accept Join
         LBIDnew = set exclusive bit of Levelth bit of LBIDtarget
         LBID RT entry Update
         Levelth RT entry = LBIDnew
         Level++
      }
   }
   Else{
      If( Temporal Routing Entry )
         Forward Join request to this entry
      Else If( There is no temporal routing entry )
         If( sending node != last RT entry )
            Forward Join request to next RT entry of sending node
         Else
            // finalize
            Sending Notification of Finalization to all of RT entries
   }
```

Fig. 3. Basic algorithm of the LBID assignment

table on behalf of the reliable node. For example, when a node wants to get an object whose key is 00001, the reliable node B takes the request, however when a node tries to get an object with 00010, the leaf node R takes the request to assist the reliable node, because the ID region of node R is from 00010, the start of the LFID slot to 00011, its nodeID. All nodes on the same sub-region have the same LBID and they are identified by the LFID. The all bits of LFID for the reliable node are set to 1 and the LFIDs of other leaf nodes change according to the behaviors of them.

The LBID table and the LFID table are used to lookup the location of a node or an object. When a node joins, we get its static nodeID by hashing its identifier. The first thing is to route to the right sub-region according to the LBID table. In this case, a node which gets a join request forwards it to the next node which is the node of the most prefix matched entry of the LBID table. After finding the sub-region, the LFID table assigns the right LFID to the new node. This join process is only for the leaf nodes and in the next section, we show the detail of the whole join process. Moreover, when a node tries to lookup an object, it sends a lookup request with the object key which is its hashed value to any other participant node. Like the case of the join process, it forwards to the right sub-region by the LBID tables and find the node whose ID region is responsible for the key by the LFID tables. Basically, we use the hashed values of both of nodes and objects, but they are used only for finding the location of them. When the nodes join to the p2p system, their ids are newly assigned by the p2p system and change according to its characteristics for the nodes to be locate on fit places.

3.2 Mobile nodeID

Load Balanced ID is the identifier for the reliable node. Each reliable node is assigned this LBID and it is responsible for storing and servicing the request for the fair and balance ID region. For this, the LBID is well distributed and evenly divided. Moreover, because there is no routing information, we need LBID routing table which helps routing to any reliable nodes.

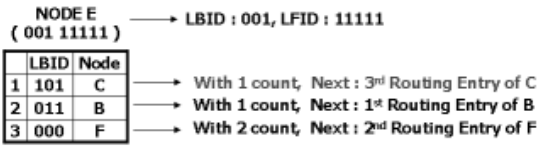


Fig. 4. LBID Routing Table of node E and finalization rule

This LBID is assigned after a node finds the any reliable nodes. If there are not enough reliable nodes, a new node is assigned the LBID and acts as the reliable node without any relation of its real characteristics. A new reliable node is assigned the right LBID and create the LBID routing table according to its LBID. Each reliable node has the state information such as Join, Level, Full and Leaf. When the Join bit sets to 1, this node can process the join request and create new LBID for the new node. The Level bit is the depth value which means how many join requests is processed in this node, that is, how many routing entries are filled. The Full bit sets to 1 after the enough reliable nodes join the p2p system and they are ready to get the leaf nodes. The Leaf bit means the number of leaf nodes which is connected to a reliable node. According to these state information, LBID is assigned automatically and correctly.

The basic algorithm for the LBID assignment is in figure 3. When a node joins to the system and there is no reliable node, the new node has the new LBID whose all bits set to 1. Otherwise, when any reliable node gets a join request, it creates new LBID based on the two information which are its LBID and the Level bit. That is, the $level_{th}$ bit of LBID sets to the exclusive bit and this is the new LBID. This simple rule makes the difference of LBID of any two closest reliable nodes even and each reliable node gets the balanced and fair ID region. The LBID routing table which is used for routing to any reliable nodes is also organized when the new LBID is created. The basic rule is the N_{th} entry of the routing table has the node information whose N_{th} bit of LBID is exclusive to the owner's LBID. In figure 4, the node whose LBID is 001 has the LBID routing table whose 1st entry has the information for the node C whose LBID is 101 and 2nd entry has the information for the node B whose LBID is 011. These bit-wise exclusive entries make the LBID routing table and any node can reach any other nodes. This LBID routing table has $logN$ of routing entries and the maximum routing hops are limited to $logN$, where N is the number of LBID bits. When there is not proper node information for a routing entry, we set the temporal routing entry. This temporal routing entry has the node information which does not matched but closest to the right node information. When the node which has the temporal routing entry gets a join request, it forwards the join request to the temporal node which is ready to process join request. After this forwarding process, the new node replaces the temporal routing entry with right routing entry and the LBID routing table is composed completely.

After the enough number of reliable nodes join, every reliable nodes should set the Full bit to 1 and be ready to get leaf nodes. When a join request routes to the reliable node by the LBID routing table and every node can not process the join request, the last requested node knows that the reliable nodes are assigned fully and the finalize mechanism should start. In this case, each node does not know the whole of the reliable

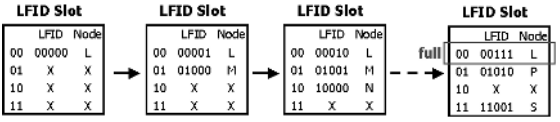


Fig. 5. LFID slot and its change according to the lifetime

nodes, but only knows the $logN$ routing entries. According to this, one node can not notify to all node and needs the efficient and systematic notification. To achieve this notification, a node sends the notification with TTL count value to every nodes which is on the routing table. Like figure 4, the notification to the N_{th} routing entry has $N-1$ TTL count value, except the 1st entry whose notification has 1 TTL value. The target node which gets the notification reduces the TTL value by 1 and sends the notification to the $M-1_{th}$ routing entry, where M is the position of the target node on the routing entry of the sending node. However, if the target node is 1st routing entry of the sending node, it sends the last notification message to its last routing entry.

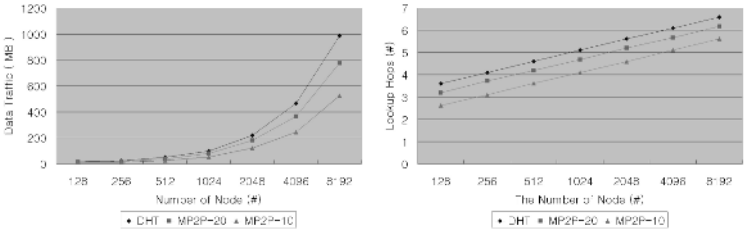
Load Free ID is the identifier of the leaf nodes. Because each leaf node is an unreliable node which joins/leaves very frequently, we should minimize the management cost for the effect of the dynamic membership change by assigning little load to leaf nodes. At first, LFID is close to 0 and when the access time of the leaf node increases, LFID also increases for the node to get more load and help the p2p system.

A new node routes to a reliable node by the LBID of the unique node key. When the Full bit of the reliable node is 1, this node processes the join request and increases the number of the Leaf bit by 1. To help assigning LFID, every reliable node has the LFID slot which divides the sub-region and each slot manages the leaf node information. The figure 5 shows the LFID slot and its change according to time. When node L joins, the first slot which is 00 slot assigns the LFID 00000 to node L. After time pass, the LFID of node L changes to 00001 and new node M gets 01000 for the second slot which is 01 slot, and so on. Each node is responsible for storing and requesting the data for the id space from the LFID 000 to the current LFID for each slot. When the LFID changes the data copy occurs, but this traffic is smaller than the management traffic of previous DHT, because these leaf nodes are free for data availability. Though each LFID increases according to the time, it can not increase more than the slot size.

4 Performance Evaluation

4.1 Simulation Setup

We make our p2p simulator which emulates the node behavior on the application layer. We implement the previous DHT based p2p algorithms such as pastry and chord and our mobile nodeID based p2p algorithm. We use 160 bit ID space to identify nodes and the number of LBID bits changes according to the number of the representative nodes which are assigned by the total number of the participant nodes. To make this dynamic characteristic, we use poison distribution whose average is 4, and to assign join/leave duration of a node, we use exponential distribution. According to this poison



(a) Bandwidth Usage (b) Lookup Hops

Fig. 6. Comparison of the bandwidth usage and the lookup hops

distribution, the lifetime of 80% of total nodes is below 60% of total simulation time, that is, only 20% of total nodes have the reliable server-like profile. Recent researches [6] measure the life distribution of the p2p nodes and show the similar distribution, and we can tell that this distribution is similar to the real world. This characteristics of nodes are assigned when the nodes are created and by using the exponential distribution with this characteristics, we can generate the on-time for which the nodes are on the p2p system and the off-time for which the nodes are off.

In the next results, DHT means the DHT based p2p algorithms and MP2P-N means the mobile nodeID based p2p algorithm in which the N percent nodes of total nodes act as representative nodes. That is, MP2P-20 can have the sub-region twice as many as MP2P-10. According to the number of the sub-region, the system makes up the bits of LBID and the bits of LFID.

4.2 Bandwidth Usage

The main problem of the current DHT p2p is the high management cost. In the figure 6(a), we show how the mobile nodeID based p2p algorithm reduces the management cost. To evaluate this cost, we assume that each node obtains same number of objects, that is, if the total number of nodes is 100 and the total number of objects is 10000, and if the total number of nodes is 200, the number of objects is 200000. In this case, the our p2p algorithm reduces the data management cost extremely. The main reason of this improvement is the behavior of leaf nodes. In DHT p2p, the frequent join/leave of leaf nodes cause the compulsory copies and update cost for routing information. However, in our p2p algorithm, the dynamic behavior of leaf nodes does not affect the data availability and the routing efficiency. According to these, the MP2P can reduce more management traffic. Moreover MP2P-10 reduces more traffic than MP2P-20. On the same node characteristics, the MP2P-20 needs more reliable nodes than MP2P-10, and the average availability of the reliable nodes of the MP2P-20 is less than the MP2P-10. In the MP2P-20, the transitions for the reliable nodes occurs more than the MP2P-10 and MP2P-20 exhausts more network bandwidth than MP2P-10. To prevent this side effect, we need the adaptive method which manages the number of reliable nodes according to the state of the nodes and this work is our ongoing job.

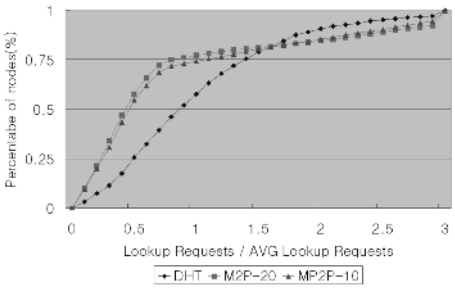


Fig. 7. Load distributions of whole participant nodes

4.3 Lookup Hops

In the p2p system, the lookup cost is also important parameter for the scalability because there are too many participants. Figure 6(b) shows the comparison of the lookup hops. For all algorithms, the lookup hops are proportion to the Log N, where N is the total number of nodes. The mobile nodeID based p2p algorithm performs more efficient lookup than normal DHT. The reason is that the our p2p algorithm uses the reliable nodes to route the lookup request and the number of these nodes are much less than the total nodes. These reliable nodes are more stable and more powerful than other nodes and they are durable nodes for the many routing requests. Additionally, the leaf nodes assist the reliable nodes to take the request for the ID region and the load of the reliable node are reasonable.

4.4 Load Balance

The figure 7 shows the load distribution for the total nodes. In this figure, we define the load of a node as the number of lookup requests of it divided by the average number of lookup requests of whole nodes. As the nature of the previous DHT based p2p algorithm, the load is distributed to the whole of nodes by the shape of the normal distribution and the average load of nodes is nearly 1. This behavior causes the heavy information maintenance overhead because the nodes which join/leave very frequently can be responsible for the big ID region. On the other hand, in our mobile nodeID based p2p algorithm, the load distribution can be classified into the representative nodes and the leaf nodes. About 75 percent of nodes have less load than other nodes because these nodes act as leaf nodes which join/leave frequently and they takes the responsible for small ID region which is assigned by the LFID. The average load of the leaf nodes are about 0.4 and these nodes are distributed uniformly. Otherwise, the representative nodes take much more load because they are alive for a long time and represent for the sub-region. The average load of these nodes are about 2. This feature which classifies the load according to nodes is very useful for the p2p system on the heterogeneous network which is consist of the various nodes such as servers, workstations and PCs. Some p2p approaches need the server-like components to increase the efficiency, and our algorithm can exploit these components easily and efficiently because the server-like nodes locates for the reliable nodes automatically.

5 Conclusions

In this paper, we suggest the mobile nodeID based p2p algorithm to reduce the information maintenance overhead by exploiting the heterogeneity of participant nodes efficiently. Unlike the DHT based p2p algorithms, the nodeID of a node changes according to its characteristic to support the p2p system efficiency and each node takes the different responsibility in accordance with its nodeID. The reliable node which is the more stable and more reliable node acts as the more important role of the routing and the replication. The leaf node which joins/leaves very frequently acts as the simple role to reduce the information maintenance traffic. The reliable node has the load-balanced ID to balance the loads and the leaf nodes has the load-free ID to reduce the responsibility. This algorithm is very good for the p2p system on the heterogeneous environment which is consist of the various kinds of nodes such as servers, workstations and PCs, because it locates the server-like nodes at the positions for the reliable nodes and can exploit these nodes efficiently. However, our algorithm may over-provision for the reliable nodes and this may decreases the performance of our algorithm. The adaptive method for the whole state of nodes to keep the proper number of reliable nodes is our ongoing job.

References

1. K.Kim and D.Park. Efficient and Scalable Client Clustering For Web Proxy Cache. *IEICE Transaction on Information and Systems*, E86-D(9), September 2003.
2. I.Stoica, R.Morris, D.Karger, M.F.Kaashoek, and H.Balakrishnan. Chord: a scalable peer-to-peer lookup service for internet applications. *In Proceedings of ACM SIGCOMM 2001*, August 2001.
3. A.Rowstron and P.Druschel. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems. *In Proceedings of the International Conference on Distributed Systems Platforms(Middleware)*, November 2001.
4. B.Y.Zhao, J.Kubiatowicz, and A.Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. *UCB Technical Report UCB/CSD-01-114*, 2001.
5. S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker. A scalable content-addressable network. *In Proceedings of ACM SIGCOMM 2001*, 2001.
6. S. Saroiu et al. A measurement study of peer-to-peer file sharing systems. *In Proceedings of MMCN 2002*, 2002.
7. R. Bhagwan, K. Tati, Y. Cheng, S. Savage and G. M. Voelker. Total Recall: System Support for Automated Availability Management. *In Proceedings of NSDI 2004*, 2004.

A Reciprocal Capacity Based Adaptive Topology Protocol for P2P Networks*

Huirong Tian, Shihong Zou, Wendong Wang, and Shiduan Cheng

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, P.R. China
tianhr@bupt.edu.cn

Abstract. A Reciprocal Capacity based Adaptive Topology Protocol (RC-ATP) for P2P networks is proposed in this paper. It is based on the rational belief that a peer is only willing to maintain connections with those which will benefit it in future. Reciprocal capacity is defined based on peers' capacity of providing services and of recommending service providers. As a result, reciprocal peers connect each other adequately. Therefore the resulting topologies are more efficient and resilient compared with Adaptive Peer-to-Peer Topologies (APT).

1 Introduction

Peer-to-Peer (P2P) networks have many benefits over traditional client-server approaches to cooperative working, data sharing and large scale parallel computing. Unfortunately, the P2P service availability is affected by the heterogeneity of peers' capacity and the topology of P2P networks. In P2P networks, it is mainly the subjective configurations of peers that cause the heterogeneity of their capacity, but not the underlying physical infrastructures[1]. Currently, there still exist a large amount of P2P services with unreliable qualities and malicious actions[2]. Therefore, in order to improve the P2P service availability, the peers' voluntary operations must be taken into account as well as the incentive mechanism.

The P2P overlay topology is very important to the P2P network performance such as scalability, efficiency and resilience. The most popular unstructured P2P systems (e.g. Gnutella) are not highly scalable or efficient because of peers' random inter-connections[3]. Alternative structured P2P schemes [4][5] are efficient for locating files but don't support semantic queries and are lack of the adaptation to highly dynamic P2P environments. In order to design scalable, efficient, and robust overlay topologies, the Adaptive Peer-to-Peer Topologies Protocol (APTP)[3] takes the issues of malicious peers or freeriders as inherent parts of the topology design. In [3], a peer directly connects to those from which it is most likely to download satisfactory content. It adds or removes neighbors based on its local trust and connection trust of them which are decided by its transaction history. However, APTP doesn't consider the difference between local trust and

* This paper was supported by the National 973 project of China(No.2003CB314806, No2006CB701306), the NSFC (NO.90204003, NO.60472067).

I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, pp. 985–994, 2006.

connection trust. Once a peer unwittingly serves as a conduit through which a malicious peer disseminates inauthentic files and is disconnected by its neighbor, it will have no chance to connect this neighbor again even if it has provided many authentic files to this neighbor. This hinders good peers with similar interests getting connected adequately, which motivates us to do this work.

A Reciprocal Capacity based Adaptive Topology Protocol (RC-ATP) for P2P networks is proposed in this paper. In RC-ATP, it is assumed that the peer is rational. Hence, a peer is only willing to maintain connections with those which will benefit it in future. In order to keep connections with such peers, it should serve them in return. Reciprocal capacity is defined based on peers' capacity of providing services and of recommending service providers. As a result, reciprocal peers connect each other adequately. In addition, a response selection mechanism is proposed to reduce the probability of trying to download files from malicious peers. Therefore the resulting topologies are more efficient and resilient compared with Adaptive Peer-to-Peer Topologies (APT).

The rest of this paper is organized as follows: section 2 presents related work. Reciprocal Capacity is defined in section 3. In section 4 we propose the reciprocal capacity based adaptive topology protocol for P2P networks. The simulation and analysis of the resulting P2P topology is followed. In the final section the conclusion is stated.

2 Related Work

Other related studies have been proposed in [6-9]. In [6], a P2P topology evolving algorithm is provided based on peers' global trust. However in large scale P2P networks, the feasibility and the necessity of establishing global trust for every peer are still doubtful. Cooper B.F. et al[7] presents a scheme to solve peers' overload problems by allowing them to self-organize into a relatively efficient network. In [7] a peer only disconnects peers when it becomes overloaded, where the connections between peers can be search links or index links. A similar scheme is proposed in [8] and [9], where one type of link (search link) is considered. A peer rates its neighbors' capacity of processing queries and independently computes a level of satisfaction according to its own capacity and its neighbors' capacity. Then it gathers more neighbors with high capacity to improve the satisfaction level until it decides that its current set of neighbors is sufficient to satisfy its capacity. The purpose of [7-9] is to adjust the load between peers and make it match to peers' capacity. Neither of these schemes addresses the issues of malicious peers or freeriders in P2P networks.

3 Reciprocal Capacity

In order to clarify the idea easily, we take the P2P network as a file-sharing network. We define *reciprocal capacity* as a peer's subjective belief that other peers will benefit it in future. In the P2P network, what one peer can do for others is either providing the file directly or forwarding the query message to

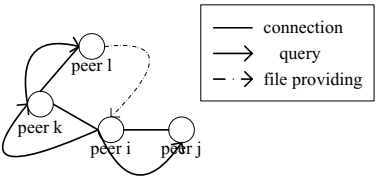


Fig. 1. Description model of capacities

enable it to be responded by other peers. As shown in Fig.1 , the query issued by peer i is propagated to peer l through its neighbor peer k. After getting the response from l , peer i tries to download the file from peer l . Although peer k can not provide the file, but it makes peer i be responded by peer l. So we believe peer k recommends peer l and call it the recommender of peer l(the file provider). Thus the capacity of peers in P2P networks is classified into two kinds: the capacity of providing services and the capacity of recommending service providers, denoted by PSC and $RSPC$ respectively. And the reciprocal capacity (noted as RC) is a weighted sum of these two kinds of capacities which are decided based on the peers' behavior history.

Before explaining how to get the value of PSC and $RSPC$, we define a property set $C^i = (\tau^i_{min}, \tau^i_{max}, w^i_{file}, w^i_{rec}, f^i_{pnlt}, threshold^i_{cap}, win^i)$ for every peer i with constraints:$\tau^i_{min} \leq \tau^i_{max}$ and $w^i_{file} + w^i_{rec} = 1$.

- τ^i_{min} is the minimal connection number that peer i should maintain.
- τ^i_{max} is the maximal connection number that peer i can serve.
- $w^i_{file}(w^i_{rec})$ is the weight that $PSC(RSPC)$ of peer contributes to RC of j believed by peer i.
- f^i_{pnlt} is the penalty factor for peers' malicious actions in the current observing window. It is defined to prevent peers which have a good behavior history from concentrative malicious actions.
- $threshold^i_{cap}$ is the lowest reciprocal capacity that peer i can sustain with its neighbors. In other words, peer i will disconnect the neighbor whose reciprocal capacity is lower than $threshold^i_{cap}$.
- win^i is the observing window size. It is defined to distinguish the peer's recent behavior from the past long-term behavior.

If peer i has interacted with peer j for M times. $Sat^p_{ij}(UnSat^p_{ij})$ is the number of satisfactory(unsatisfactory) transactions that peer i has had with peer j . We define PSC_{ij} as the probability that the $(M+1)th$ transaction with j will satisfy peer i. According to the deduction in [10], PSC_{ij} can be calculated by the following equation:

$$PSC_{ij} = \frac{Sat^p_{ij} + \alpha^p}{Sat^p_{ij} + UnSat^p_{ij} + \alpha^p + \beta^p}, \alpha^p, \beta^p > 0, \tag{1}$$

where α^p and β^p are referred as hyper parameters, which represent the prior satisfactory and unsatisfactory transactions respectively.

We introduce the observing window win^i and the penalty factor f^i_{pnlt}, and define the PSC as follows:

$$PSC_{ij} = \frac{Sat^p_{ij} + \alpha^p}{Sat^p_{ij} + UnSat^p_{ij} + \alpha^p + \beta^p + \sum_{k=1}^{m^p_{ij}} f^i_{pnlt}}, m^p_{ij} > 0, \quad (2)$$

where $UnSat^p_{ij}$ is redefined as the number of unsatisfactory transactions that peer i has had with peer j until the last observing window. m^p_{ij} represents the number of unsatisfactory transactions peer i has had with peer j in the current observing window. f^i_{pnlt} is determined according to the unsatisfactory level.

Similarly,

$$RSPC_{ij} = \frac{Sat^r_{ij} + \alpha^r}{Sat^r_{ij} + UnSat^r_{ij} + \alpha^r + \beta^r + \sum_{k=1}^{m^r_{ij}} f^i_{pnlt}}, m^r_{ij} > 0, \alpha^r, \beta^r > 0, \quad (3)$$

where Sat^r_{ij} is the number of satisfactory service providers that peer j has recommended to peer i (In the scenario of Fig.1, if peer i is satisfied with the transaction with peer l, we say peer k has recommended a satisfactory service provider to peer i. Otherwise, peer k has recommended an unsatisfactory service provider to peer i), and $UnSat^r_{ij}$ is the number of unsatisfactory service providers that peer j has recommended to peer i until the last observing window. α^r and β^r are referred as hyper parameters, which represent the prior number of satisfactory and unsatisfactory service providers which have been recommended respectively. m^r_{ij} is the number of unsatisfactory service providers that peer j has recommended to peer i in the current observing window.

Therefore the reciprocal capacity of peer j that peer i believes is defined as

$$RC_{ij} = w^j_{file} \times PSC_{ij} + w^i_{rec} \times RSPC_{ij} \quad (4)$$

When the system starts up, $Sat^p_{ij} = UnSat^p_{ij} = m^p_{ij} = 0$ and $Sat^r_{ij} = UnSat^r_{ij} = m^r_{ij} = 0$. So $PSC_{ij} = \alpha^p/(\alpha^p + \beta^p)$(noted as PSC_{init}), which is the probability that a strange peer would provide a satisfactory service, and $RSPC_{ij} = \alpha^r/(\alpha^r + \beta^r)$(noted as $RSPC_{init}$), which is the probability that a strange peer would recommend a satisfactory provider. Thus $RC_{ij} = w^j_{file} \times PSC_{init} + w^i_{rec} \times RSPC_{init}$(noted as $Init\_Capacity$), which is the probability that a peer would get benefit from a strange peer.

4 RC-ATP

We assume that peers can not easily change identities. When a peer starts up, it uses the bootstrapping mechanism as that in Gnutella to find other nodes. Then, it directly sends a connection request to a random node until it has τ_{min} neighbors. To locate services in P2P networks, peers initiate queries which are flooded as that in Gnutella. To calculate $RSPC$, queries include an original neighbor field which is used to keep tracking of original neighbors to which peers

initially send queries. The response appended with the original neighbor field of the query is directly sent to the query initiator. In this section, the response selection mechanism is presented at first. The reciprocal capacity based topology adaptation mechanism is followed.

4.1 Response Selection Mechanism

Peer i, upon receiving responses, will set a value for every response, noted as p_{gr}. This value reflects the probability that this response would bring a satisfactory transaction. If the responder j has had transactions with peer i, then $p_{gr} = PSC_{ij}$. Otherwise, $p_{gr} = RSPC_{ik}$, where k is the recommender of j.

Then, the responses, each corresponding to a responder which has had transactions with peer i, are ordered by p_{gr} descendingly if their p_{gr} is larger than PSC_{init}. This response list is called a good response list. The responses from strange responders, whose p_{gr} is larger than $RSPC_{init}$, are also ordered by p_{gr} descendingly and appended to the good response list.

After obtaining the good response list, peer i will sequentially try to connect and download a file from the responders in the good response list. If it can not download a satisfactory file from the responders in the good response list, it will try the rest responses randomly until it get a good file or try every response.

How to choose a download source is very important as it determines which peer to have a transaction with. If one peer maintains the connection with a neighbor with low $RSPC$ and high PSC, there is a likelihood of getting malicious responses. This response selection mechanism can reduce the probability of trying to download files from malicious peers.

4.2 Topology Adaptation Mechanism

The cooperation of peers is vital important to the viability of P2P networks[11]. Intuitively, the more cooperative peers are, the more efficient the network is. The reciprocal capacity can reflect the possibility that peers will cooperate in future. So, we adjust the topology by making every peer connect peers with larger RC.

Topology adaptation happens when an observing window is finished. In order to get the profit, a peer will try to maintain τ_{min} neighbors with reciprocal capacity larger than or equal to $Init\_Capacity$ and try to connect peers with larger reciprocal capacity. Before introducing the topology adaptation mechanism, some notations are firstly defined as follows:

$Nb(i)$: the set of peer i's neighbors.

$Fm(i)$: the set of the familiar peers of peer i, which are neighbors of peer i, have ever been neighbors of peer i or have transactions with peer i.

$Fv(i) = \{j | RC_{ij} > Init\_Capacity, j \in Fm(i), j \notin Nb(i)\}$: the set of peers which peer i prefers to connect but hasn't connected yet.

$Fellow(i) = \{j | RC_{ij} \geq Init\_Capacity, j \in Nb(i)\}$: the set of peer i's neighbors with reciprocal capacity larger than or equal to $Init\_Capacity$.

$Fv(i)_{max} = \{j | j \in Fv(i), \forall k \in Fv(i)\ and\ k \neq j, RC_{ij} > RC_{ik}\}$: the peer belongs to $Fv(i)$ whose reciprocal capacity is the largest.

$Nb(i)_{min} = \{j | j \in Nb(i), \forall k \in Nb(i)\ and\ k \neq j, RC_{ij} < RC_{ik}\}$: the neighbor of peer i whose reciprocal capacity is the lowest.

1. Sending a connection request

When it's time to adapt the topology, peer i will do the following steps:

Firstly, If $|Fellow(i)| < \tau^i_{min}$ and $Fv(i) \neq \emptyset$, peer i will send a connection request to $Fv(i)_{max}$, and remove it from $Fv(i)$; If $|Fellow(i)| < \tau^i_{min}$ and $Fv(i) = \emptyset$ it will just send a connection request to a strange peer in the network randomly. If the connection request is accepted and $|Nb(i)| > \tau^i_{max}$, $Nb(i)_{min}$ will be disconnected.

Peer i will try to get connected with τ^i_{min} neighbors whose reciprocal capacity is not lower than $Init\_Capacity$. But if it tries several times and no peer accepts its connection requests, it has no choice but to give up.

Secondly, peer i examines neighbors' reciprocal capacity and disconnects the neighbor whose reciprocal capacity is lower than $threshold^i_{cap}$.

Thirdly, if $Fv(i) \neq \emptyset$ and $|Nb(i)| < \tau^i_{max}$, peer i will send a connection request to $Fv(i)_{max}$ and remove it from $Fv(i)$; If $Fv(i) \neq \emptyset$, $|Nb(i) = \tau^i_{max}|$ and the reciprocal capacity of $Fv(i)_{max}$ is larger than that of $Nb(i)_{min}$, peer i will send a connection request to $Fv(i)_{max}$. If the request is accepted, $Nb(i)_{min}$ will be disconnected.

2. Receiving a connection request

Peer j will only accept the connection request from peer i if RC_{ji} is larger than $Init\_Capacity$ and one of the following conditions is true: 1)$|Nb(j)| < \tau^j_{max}$; 2) RC_{ji} is larger than the reciprocal capacity of $Nb(j)_{min}$. In the second case, peer j will disconnect $Nb(j)_{min}$.

There are two issues not addressed by the above topology adaptation algorithm.

No-reciprocal peers. The issue of freeriders arising in P2P routing is that freeriders choose not to forward queries for others to conserve local bandwidth. Peer i may find that its neighbors neither provide files to it nor recommend file providers to it. This is the case that peer i enters the network from the wrong place or its neighbors are freeriders which only download files, and neither forward query messages nor share their own files. We define the notion of a *no-reciprocal peer* to be a peer whose reciprocal capacity is lower than or equal to $Init\_Capacity$(the no-reciprocal concept can be redefined according to the requirements of different P2P applications or according to different peers' opinions). Peer i will disconnect neighbors which are still no-reciprocal peers after several observing windows.

No responses. Although peer i disconnects its no-reciprocal neighbors, there is still an issue that it may get no responses for a period. The reason is that i's neighbors may only forward query messages for i or provide i a few authentic files at the initial stage to increase their chances of maintaining connections with i . So when no responses are received for several observing windows, peer i will replace neighbors with a random strange peer for τ^i_{min} times.

In RC-ATP, there is two-level topology adaptation. Adapting neighbors based on reciprocal capacity after a observing window can be treated as a short-term adaptation. It is a long-term adaptation that dropping no-reciprocal peers or replacing neighbors for no responses after several observing windows. Peers adjust neighbors in a short term for their interests in a quick profit while the long-term adaptation guarantees them get profit from neighbors and only serve reciprocal peers.

5 Simulation and Analysis

RC-ATP is implemented based on Query Cycle Simulator[12] and is compared with APTP. The experiment data of APTP is obtained by running the demo of adaptive topologies in [12]. There are 800 query cycles in one experiment and the results are averaged over 5 runs.

5.1 Simulation Environment

The network is initialized as the random graph where peers have τ_{min} neighbors. The query message is flooded with TTL=4. In the experiment, there are 500 normal peers and 50 malicious peers (providing inauthentic files in order to undermine the network performance). 25% of the normal peers are freeriders (only downloading files, and neither sharing their own files nor forwarding queries) while the others are good peers(normally downloading and uploading files). Let $\alpha^p, \beta^p, \alpha^r, \beta^r = 1$ so that $Init\_Capacity = 0.5$. To simplify the scenario, all peers have the same property set $C = (3, 20, 0.8, 0.2, 2, 0.4, 1)$. The connection request from peers whose local trust scores are -1 or connection trust scores are -5 will not be accepted in APTP. Normal peers are in the uptime with the uniform random distribution over [0%, 100%] and issue queries in the uptime with the uniform random distribution over [0%, 50%] while malicious peers are always up and issue queries. In addition, different types of peers also vary in their behavior of responding queries and providing files. For good peers, the probability of providing inauthentic files is 5%, while malicious peers will respond to all queries they have received and return inauthentic files for all download requests.

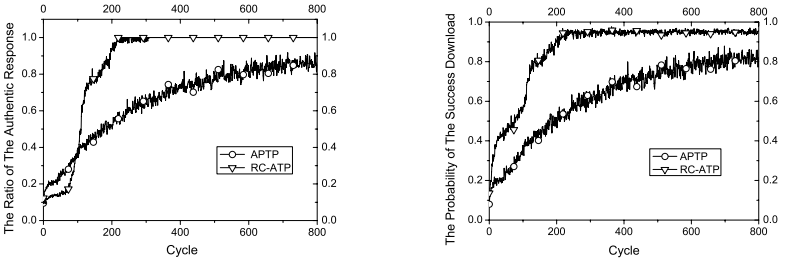
The content distribution model is the same as that in [13]. Each file is characterized by the content category c and the popularity rank r within this category. c and r both follow the Zipf distribution. Files are distributed probabilistically to peers based on their popularities and the content categories that peers are interested in. Distributions used in the model are based on the measurement of real-world P2P networks[11]. In our simulation, 20 content categories are hold in the network and each peer is at least interested in 4 categories.

5.2 Efficiency

The efficiency of the network describes how good peers can efficiently get reliable files. They are as follows:

– The Ratio of the Authentic Response (RAR): reflects the average ratio of responses which are given by good peers to the total responses.
– The Probability of the Success Download (PSD): reflects the average probability that the first download is an authentic file.

If good peer i has issued a query and received $r_i > 0$ responses among which r_i^a are given by good peers. Then $RAR_i = r_i^a / r_i$. If i downloads an authentic file after trying λ_i times, $PSD_i = 1/\lambda_i$. Otherwise $PSD_i = 0$. We define the RAR and PSD of the network separately as the average of RAR_i and PSD_i over all such good peers in the network.



(a) The ratio of the authentic response

(b) The probability of the success download

Fig. 2. Efficiency

Fig.2(a) plots the RAR of RC-ATP and APTP. The RAR of APTP increases rapidly at the initial stage because of the adoption of the connection trust. However, in RC-ATP with the query cycle increasing, malicious peers are quickly known by other peers and eliminated from the network. As a result, the RAR of RC-ATP will be greater than that of APTP at cycle 102 and be stable at 1 after 311 cycles. The PSD of RC-ATP and APTP is shown in Fig.2(b). As the query cycle increasing, the PSD of RC-ATP is approximately 0.95, which is the ratio that good peers provide authentic files for download requests. At the initial stage PSD of RC-ATP is higher than that of APTP although RAR of RC-ATP is lower than that of APTP. This is because the response selection mechanism reduces the probability that good peers try to download files from malicious peers. The higher the PSD is, the lower the likelihood of downloading inauthentic files is.

So Good peers get reliable files more efficiently in RC-ATP than in APTP.

5.3 Resilience

The network resilience studies the effect of removing nodes from the network. Such attacks are classified into two categories: random failures (removing nodes randomly) and intentional attacks (removing nodes intentionally)[14]. In this paper, we evaluate the resilience of the resulting topologies of RC-ATP and APTP

under degree-based attacks (DAttack), betweenness-based attacks (BAttack) as well as random failures (Failure) of nodes. Three metrics are analyzed: the relative size of the largest cluster S (the ratio between the size of the largest cluster and the size of the original network), the average shortest path length between good peers pl (If there is no path between two good peers, the shortest path length between them is defined as $pl_{max} = 15$) and the network diameter d.

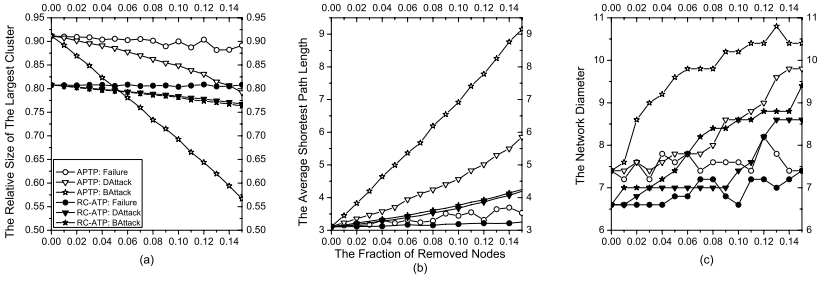


Fig. 3. Results of Failure, DAttack and BAttack of nodes measured by S, pl, and d

Fig.3 summarizes the results of Failure, DAttack and BAttack of nodes measured by S, pl, and d at cycle 800 as functions of the fraction of removed nodes f in [0, 0.15]. In Fig.3(a) the initial S of RC-ATP and APTP is not equal to 1 and the former is lower than the later. This is because in RC-ATP not only all malicious peers but also some freeriders lose connections with the good peers cluster while almost all malicious peers are eliminated from the network in APTP at cycle 800. S decreases, pl increases and d increases in RC-ATP more slowly than in APTP under DAttack and BAttack of nodes. So the network connectivity and the reachability between good peers are better in RC-ATP than in APTP under intentional attacks. The network of RC-ATP is resilient not only to random failures but also to intentional attacks while the network of APTP is only resilient to random failures. The reason is that RC-ATP distinguishes the different capacities of peers and reciprocal peers are connected adequately. So RC-ATP is more resilient than APTP.

6 Conclusion

In this paper, a reciprocal capacity based adaptive topology protocol for P2P networks is proposed. This protocol is based on the rational belief that a peer is only willing to maintain connections with those which will benefit it in future. Reciprocal capacity is defined based on the capacity of providing services and the capacity of recommending service providers, which are calculated according to the peers' behavior history. In addition, a response selection mechanism is proposed to reduce the probability of trying to download files from malicious peers. Compared with APTP, the resulting topology of RC-ATP is more efficient because in RC-ATP good peers can download authentic files with higher PSD.

Due to the adequate connections between reciprocal peers, the network connectivity and the reachability between good peers are better in RC-ATP than in APTP under DAttack and BAttack of nodes. So RC-ATP is more resilient than APTP.

References

1. Buyya R., Stockinger‡ H., Giddy J., Abramson D.: Economic Models for Management of Resources in Peer-to-Peer and Grid Computing. SPIE International Conference on Commercial Applications for High-Performance Computing, Denver, USA, Computational Economics Press, 2001.8, pp.1-12
2. Adar E., Huberman B.A.: Free Riding on Gnutella. Technical Report, SSL-00-63, Internet Ecologies Area Xerox Palo Alto Research Center, Palo Alto, Canada, October 2002
3. Condie T., Kamvar S.D., Garcia-Molina H.: Adaptive Peer-to-Peer Topologies. In the 4th International Conference on Peer-to-Peer Computing, Zurich, Switzerland, August 2004
4. Stoica I., Morris R., Karger D., Kaashoek M.F., Balakrishnan H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In Proceeding of ACM SIGCOMM'01, San Diego ,California, USA, August 2001
5. Ratnasamy S., Francis P., Handley M., Karp R.: A Scalable Content-addressable Network. In Proceeding of ACM SIGCOMM'01, San Diego ,California, USA, August 2001.
6. Wen D.: The Research on Trust-aware P2P Topologies and Constructing Technologies. Ph.D Thesis, National University of Defense Technology, P.R. China, 2003
7. Cooper B.F., Garcia-Molina H.: Ad Hoc, Self-supervising Peer-to-Peer Search Networks. Technical Report, Stanford University, 2003. http://dbpubs.stanford.edu/pub/2003-4
8. Lv Q., Ratsnasamy S., Shenker S.: Can Heterogeneity Make Gnutella Scalable? In Proceeding of the First International Workshop on P2P Systems, USA, March 2002.
9. Chawathe Y., Ratnasamy S., Breslau L., Lanham N., Shenker S.: Making Gnutella-like P2P Systems Scalable. In Proceeding of ACM SIGCOMM'03, Germany, August 2003.
10. Heckerman D.: A Tutorial on Learning With Bayesian Networks. Technical Report, MSR-TR-95-06, 1995. ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf
11. Saroiu S., Gummadi P.K., Gribble S.D.: A Measurement Study of Peer-to-Peer File Sharing Systems. In Proceedings of Multimedia Conferencing and Networking, San Jose, January 2002.
12. http://p2p.stanford.edu/www/demos.htm
13. Schlosser M., Condie T., Kamvar S.: Simulating a File-Sharing P2P Network. In First Workshop on Semantics in P2P and Grid Computing, December, 2002
14. Albert R., Jeong H., Barabasi A.L.: Error and Attack Tolerance of Complex Networks. Nature, Vol.406, pp.378-382, 2000

Author Index