Torsten Braun
Georg Carle
Yevgeni Koucheryavy
Vassilis Tsaoussidis (Eds.)

Wired/Wireless Internet Communications

LNCS 3510

Third International Conference, WWIC 2005
Xanthi, Greece, May 2005
Proceedings

Springer

Lecture Notes in Computer Science 3510

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

Editorial Board

Torsten Braun Georg Carle
Yevgeni Koucheryavy Vassilis Tsaoussidis (Eds.)

Wired/Wireless Internet Communications

Third International Conference, WWIC 2005
Xanthi, Greece, May 11-13, 2005
Proceedings

Springer

Volume Editors

Torsten Braun
University of Bern, Institute of Computer Science and Applied Mathematics
3012 Bern, Switzerland
E-mail: braun@iam.unibe.ch

Georg Carle
University of Tübingen, Department of Computer Networks and Internet
Auf der Morgenstelle 10 C, 72076 Tübingen, Germany
E-mail: carle@fokus.gmd.de

Yevgeni Koucheryavy
Tampere University of Technology, Institute of Communication Engineering
Tampere, Finland
E-mail: yk@cs.tut.fi

Vassilis Tsaoussidis
Demokritos University of Thrace
Department of Electrical and Computer Engineering
12 Vas. Sofias Str., 671 00 Xanthi, Greece
E-mail: vtsaousi@ee.duth.gr

Library of Congress Control Number: 2005925605

CR Subject Classification (1998): C.2, H.4, D.2, D.4.4, K.8

ISSN 0302-9743
ISBN-10 3-540-25899-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25899-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11424505 06/3142 5 4 3 2 1 0

Preface

Welcome to the 3rd International Conference on Wired/Wireless Internet Communications (WWIC). After a successful start in Las Vegas and a selective conference in Germany, this year's WWIC demonstrated the event's maturity. The conference was supported by several sponsors, both international and local, and became the official venue for COST Action 290. That said, WWIC has now been established as a top-quality conference to promote research on the convergence of wired and wireless networks.

This year we received 117 submissions, which allowed us to organize an exciting program with excellent research results, but required more effort from the 54 members of the international Program Committee and the 51 additional reviewers. For each of the 117 submitted papers we asked three independent reviewers to provide their evaluation. Based on an online ballot phase and a TPC meeting organized in Colmar (France), we selected 34 high-quality papers for presentation at the conference. Thus, the acceptance rate for this year was 29%.

The selected papers were organized into 9 sessions:

1. Mobility Management
2. Transport Protocols and Congestion Control
3. QoS and Routing
4. Quality of service
5. Wireless Multi-hop Networks and Cellular Networks
6. Traffic Characterization and Modeling
7. Ad Hoc Networks
8. IEEE 802.11 and Other Wireless MAC Protocols
9. Energy Efficiency and Resource Optimization

We would like to thank the authors for choosing to submit their results to WWIC 2005. We would also like to thank all the members of the Technical Program Committee, the members of the Organizing Committee, as well as all the additional reviewers for their effort in providing detailed and constructive reviews. We are grateful to the two keynote speakers, Ian Akyildiz and Michael Smirnoff, for accepting our invitation; and to Springer LNCS for supporting us again this year.

We hope that all participants enjoyed the technical and social conference program, the hospitality of our Greek hosts and the beauty of the conference location. Next year's conference will take place in Bern, Switzerland. We hope to see you there again.

May 2005

Torsten Braun
Georg Carle
Yevgeni Koucheryavy
Vassilis Tsaoussidis

Organization

Steering Committee

Torsten Braun	University of Bern, Switzerland
Ibrahim Matta	Boston University, USA
Nitin Vaidya	University of Illinois at Urbana-Champaign, USA
Vassilis Tsaoussidis	Demokritos University, Greece

Executive Commitee

General Chairs:	Yevgeni Koucheryavy (Tampere University of Technology, Finland)
	Vassilis Tsaoussidis (Demokritos University, Greece)
TPC Co-chairs	Torsten Braun (University of Bern, Switzerland)
	Georg Carle (University of Tubingen, Germany)

Technical Program Commitee

Ozgur B. Akan	Middle East Technical University (Turkey)
Bengt Ahlgren	SICS (Sweden)
Manuel Alvarez-Campana	Universidad Politecnica de Madrid (Spain)
Farooq Anjum	Telcordia Technologies (USA)
Harmen R. van As	TU Wien (Austria)
Hans van den Berg	TNO TELECOM-University of Twente (Netherlands)
Chris Blondia	University of Antwerp (Belgium)
Torsten Braun	University of Bern (Switzerland)
Andrew Campbell	Columbia University (USA)
Xiuzhen Cheng	George Washington University (USA)
Jon Crowcroft	University of Cambridge (UK)
Klaus David	University of Kassel (Germany)
Piet Demeester	Ghent University (Belgium)
Sonia Fahmy	Purdue University (USA)
Giambene Giovanni	Universita degli Studi di Siena (Italy)
Andrei Gurtov	University of Helsinki (Finland)
Peder Johannes	Norwegian University (Norway)
Daniel Kofman	ENST (France)
Yevgeni Koucheryavy	Tampere University (Finland)

Rolf Kraemer	IHP Microelectronics (Germany)
Adrian Lahanas	University of Cyprus (Cyprus)
Peter Langendoerfer	IHP Microelectronics (Germany)
Ben Liang	University of Toronto (Canada)
Pascal Lorenz	University of Haute Alsace (France)
Petri Maehoenen	University Oulu (Finland)
Christian Maihoefer	Daimler Chrysler (Germany)
Saverio Mascolo	Politecnico di Bari (Italy)
Ibrahim Matta	Boston University (USA)
Ingrid Moerman	Univ. of Gent (Belgium)
Edmundo Monteiro	University of Coimbra (Portugal)
Ioanis Nikolaidis	University of Alberta (Canada)
Guevara Noubir	Northeastern University (USA)
Fotini-Niovi Pavlidou	Aristotle University of Thessaloniki (Greece)
George Pavlou	University of Surrey (UK)
George Polyzos	AUEB (Greece)
Theodoros Salonidis	Rice University (USA)
Jochen Schiller	FU Berlin (Germany)
Andrew Scott	Lancaster University (UK)
Dimitris Serpanos	University of Patras (Greece)
Sherman Shen	University of Waterloo (Canada)
Vasilios A. Siris	University of Crete and ICS-FORTH (Greece)
Roshni Srinivasan	University of Maryland (USA)
Burkhard Stiller	ETH Zurich (Switzerland)
Ivan Stojmenovic	University of Ottawa (Canada)
Phuoc Tran-Gia	University of Wuerzburg (Germany)
Vassilis Tsaoussidis	Demokritos University (Greece)
Thierry Turletti	INRIA, Sophia-Antipolis (France)
Lars Wolf	Braunschweig University (Germany)
Adam Wolisz	Technical University of Berlin (Germany)
Miki Yamamoto	Osaka University (Japan)
Chi Zhang	Florida International University (USA)
Martina Zitterbart	TU Karlsruhe (Germany)
Michele Zorzi	University of Ferrara (Italy)

Organizing Commitee

Sotirios Kontogiannis	Demokritos University (Greece)
Lefteris Mamatas	Demokritos University (Greece)
Ieremias Ntoumpakis	Demokritos University (Greece)
Ruy de Oliveira	University of Bern (Switzerland)
Panagiotis Papadimitriou	Demokritos University (Greece)
Ioannis Psaras	Demokritos University (Greece)
Christos Samaras	Demokritos University (Greece)
Ageliki Tsioliaridou	Demokritos University (Greece)

Additional Reviewers

Evangelos Aggelakis
Marc Bechler
Fernando Boavida
Thomas Bohnert
Laila Daniel
Roman Dunaytsev
Elias Efstathiou
Peder J. Emstad
Stelios Georgoulas
Xiaoyuan Gu
Jarmo Harju
Dan He
Jeroen Hoebeke
Roy Ho
Michael Howarth
Sven Jaap
Riku Jäntti
Yuming Jiang
Verena Kahmann
Stelios Karapantazis
Sotiris Kontogiannis
Teemu Koponen
Øivind Kure
Benoit Latre
Tom Van Leeuwen
Tim Leinmüller
Lefteris Mamatas
Paulo Marques
Michael Methfessel
Dmitri Moltchanov
John Murphy
Philippe De Neve
Geir Øien
P. Papadimitriou
Thanasis Papaioannou
Liesbeth Peters
Ioannis Psaras
Wei Qian
Niky Riga
Oriol Sallent
Sergei Semenov
Siva Sivavakeesar
Dirk Staehle
Apostolos Traganitis
Despoina Triantafyllidou
Orestis Tsigas
Ageliki Tsioliaridou
Christopher Ververidis
Vasilis Vogkas
Ning Wang
Oliver Wellnitz

Table of Contents

Session : QoS and Routing

Session : Quality of Service

Session : ireless Multi-hop Networks and Cellular Networks

Session : Tra c Characterization and Modeling

Session : Ad hoc Networks

Session : IEEE and Other ireless MAC Protocols

Impact of Link State Changes and Inaccurate Link State Information on Mobility Support and Resource Reservations

Liesbeth Peters*, Ingrid Moerman, Bart Dhoedt, and Piet Demeester

Department of Information Technology (INTEC),
Ghent University - IMEC, Sint-Pietersnieuwstraat 41,
B-9000 Gent, Belgium
Tel.: +32 9 33 14900, Fax: +32 9 33 14899
{Liesbeth.Peters, Ingrid.Moerman, Bart.Dhoedt, Piet.Demeester}
@intec.UGent.be

Abstract. The increasing use of wireless networks and the popularity of multimedia applications, leads to the need of QoS (Quality of Service) support in a mobile IP-based environment. This paper presents the framework, needed to support both micromobility and resource reservations. We present an admission control mechanism in which a mobile host can trigger reservations without performing handoff, taking advantage of link state changes caused by the handoff of other mobile hosts. We also investigate the impact of inaccurate link state information and the occurrence of simultaneous handoffs on the performance of the handoff and reservation mechanism. This impact is higher when only a small part of the mobile hosts can receive QoS service at the same time. For the simulations, we use Q-MEHROM [10]. Herein, QOSPF [11] gathers the link state information and calculates the QoS tables. However, the ideas and results presented in this paper are not restricted to these protocols.

1 Introduction

Today, wireless networks evolve towards IP-based infrastructures to allow a seamless integration between wired and wireless technologies. Most routing protocols that support IP mobility, assume that the network consists of an IP-based core network and several IP domains (access networks), each connected to the core network via a domain gateway. Mobile IP [1, 2], which is standardized by the IETF, is the best known routing protocol that supports host mobility. Mobile IP is used to support macromobility, while, examples of micromobility protocols are per-host forwarding schemes like Cellular IP [3], Hawaii [4], and tunnel-based schemes like MIPv4-RR [5]. These protocols try to solve the weaknesses of Mobile IP by aiming to reduce the handoff latency, the handoff packet loss and the load of control messages in the core network.

* Liesbeth Peters is a Research Assistant of the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium).

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 1–11, 2005.

Most research in the area of micromobility assumes that the access network has a tree or hierarchical structure. However, for reasons of robustness against link failures and load balancing, a much more meshed topology is required. In our previous work, we developed MEHROM (Micromobility support with Efficient Handoff and Route Optimization Mechanisms). It shows a good performance, irrespective of the topology, for frequent handoffs within an IP domain. For a detailed description of MEHROM and a comparison with Cellular IP, Hawaii and MIPv4-RR, we refer to [6].

In a mobile IP-based environment, users want to receive real-time applications with the same QoS (Quality of Service) as in a fixed environment. Several extensions to RSVP (Resource Reservation Protocol) under macro- and micromobility are proposed in [7]. However, the rerouting of the RSVP branch path at the cross-over node under micromobility again assumes a tree topology and introduces some delay. Current work within the IETF NSIS (Next Steps in Signalling) working group includes the analysis of some existing QoS signalling protocols for an IP network [8] and the listing of Mobile IP specific requirements of a QoS solution [9]. In [10], we presented Q-MEHROM, which is the close coupling of MEHROM and resource reservations. By defining the resource reservation mechanism as an extension of the micromobility protocol, resources can be re-allocated at the same time that the routing tables are updated.

In this paper, we investigate how the admission control of a mobile host can take advantage of link state changes due to the handoff of other mobile hosts. We also study the impact of inaccurate link state information and simultaneous handoffs on the handoff and reservation mechanism. The rest of this paper is structured as follows. Section 2 presents the framework used. Section 3 describes a way to enhance the admission control mechanism. In Sect. 4, the impact of inaccurate link state information and simultaneous handoffs on the handoff and reservation mechanism is explained. Simulation results are presented in Sect. 5. The final Sect. 6 contains our concluding remarks.

2 Framework

Figure 1 presents the framework, used to support micromobility routing and resource reservations. A micromobility protocol updates the routing tables in the access network to support data traffic towards mobile hosts (MHs). In this paper, we consider the resource reservations for data flows towards MHs.

Micromobility Support and Resource Reservations. The central block of Fig. 1 is responsible for the propagation of mobility information and the reservation of requested resources through the access network.

For the simulations, Q-MEHROM [10], based upon the micromobility protocol MEHROM [6], is used. MEHROM is a per-host forwarding scheme. At the time of handoff, the necessary signalling to update the routing tables is kept locally as much as possible. New entries are added and obsolete entries are explicitly deleted, resulting in a single installed path for each MH. These characteristics

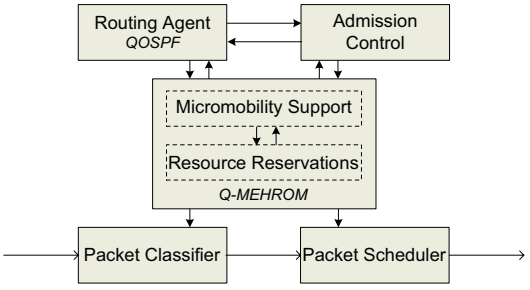


Fig. 1. Framework for micromobility and QoS support in an IP-based access network

make the MEHROM handoff scheme very suitable to be closely coupled with a resource reservation mechanism for traffic towards the MHs. During handoff, Q-MEHROM propagates the necessary QoS information as fast as possible to limit the degradation of the delivered QoS. Hereby, the signalling is restricted to the part of the new path that does not overlap with the old path and reserved resources along the unused part of the old path are explicitly released.

Topology and Link State Information. A micromobility protocol needs information about the access network topology, e.g. to find the next hop to the domain gateway (GW). A resource reservation mechanism requests information about the link states, e.g. to find a path with enough available bandwidth. The routing agent, presented by the upper left block of Fig. 1, gathers this information and computes paths in the access network that satisfy given QoS requirements. The micromobility and resource reservation mechanism, in its turn, informs the routing agent every time resources are reserved or released on a link.

For the simulations, we use QOSPF, as described in [11]. QOSPF advertises link metrics, like available bandwidth and delay, across the access network by link state advertisements. As a result, all routers have an updated link-state database. To provide useful information to the micromobility and resource reservation mechanism, QOSPF calculates, in each router, several QoS routing tables from the database. Q-MEHROM uses information from the following QoS tables:

- The Delay table of a router has this router as source and an entry for every access router (AR) as destination. Therefore, the router calculates the path with smallest delay to a specific AR. The next hop to that AR is then put in the Delay table.
- A Bandwidth table is used to reserve resources for a new traffic flow or to switch from best-effort to QoS service. Here, the available bandwidth on the links of the access network as well as the hop count are taken into account. As we consider traffic towards the MHs, a router calculates a Bandwidth table with the GW as source and itself as destination. For a certain value of the hop count, the path with at most this amount of hops and with maximum bandwidth is calculated. The Bandwidth table gives the last node on the path before reaching the router.

– A Reuse-Bandwidth table is used for the handoff mechanism of flows with QoS service. After handoff, it is possible that the new path partly overlaps with the old path. Resources along this common part should be reused and not allocated twice. Therefore, QOSPF must consider the resources, reserved for the MH before handoff, also as available resources. Using this additional information, the Reuse-Bandwidth table is calculated.

Admission Control. The upper right block of Fig. 1 represents the admission control policy. We have chosen for a simple admission control priority mechanism: priority is given to handoff requests above new requests. When a new request for resources is made by a MH, its AR decides whether the access network has enough resources to deliver the required QoS. If not, the request is rejected. If, at the time of handoff, the required resources for an existing connection can not be delivered via the new AR, the handoff request is not rejected, but the delivered service is reduced to best-effort. At the time of a next handoff, the availability of resources is checked again and reservations can be made.

When the micromobility and resource reservation mechanism of an AR must decide how to treat a new request or handoff request, the admission control mechanism gives the routing agent information about the resources that were reserved by the MH before handoff. The routing agent in its turn provides information about paths with sufficient resources in the access network. The admission control then informs the micromobility and resource reservation mechanism about the availability of resources.

3 Impact of Link State Changes on Admission Control

The simple admission control mechanism, explained in Sect. 2 and used in [10], has an important drawback: as long as a MH, receiving best-effort service, does not perform handoff, its service remains best-effort. Even if enough resources became available due to the handoff of other MHs.

In order to overcome this important drawback, we propose to extend the admission control policy. In this extended admission control mechanism, the AR must check whether enough resources became available for one of the MHs in its routing cache that still receive best-effort service. This check can be triggered periodically or after the receipt of a link state advertisement, indicating that the available resources on a link are increased. However, both of these trigger mechanisms can not avoid that the AR starts the resource reservation mechanism while the MH performs handoff to another AR, possibly leading to router inconsistency in the access network. Therefore, we propose a solution in which the MH itself triggers the link state check by the AR, by sending a new Mobile IP Registration Request. The MH sends this trigger when it receives a new beacon, i.e. a Mobile IP Agent Advertisement, from its current AR, if it receives best-effort service and is not performing handoff soon. To make an estimation about the next time of handoff, it can be very useful to use link layer (L2) information, e.g. the signal strength of the beacons. If the AR detects that enough

resources became available, the AR starts the resource reservation mechanism, and the delivered service is switched back from best-effort to QoS. We will call the reservation of resources without performing handoff, the switch mechanism.

4 Use of Inaccurate Link State Information

As the state of the access network changes constantly, e.g. as a result of handoffs, the QoS tables need to be recalculated as time passes by. Several approaches for these recalculations can be used. A QoS table can be recalculated either:

- Periodically at given time intervals P, irrespective of the number of link state changes;
- After an amount of N_{ads} received link state advertisements or interface changes, proportional to the number of link state changes;
- On demand, depending on the number of times information is requested from the QoS table.

For an increasing value of P and N_{ads}, the number of calculations in the routers of the access network decreases at the cost of possible inaccurate information in the QoS tables. If the information is calculated on demand, the most accurate link state information, available by the router, is used to calculate the QoS tables.

While the routers in the access network use the QoS tables to find the next hop towards the old AR or towards the GW, the ARs use the information also during admission control. Even though an AR decided that enough resources, e.g. bandwidth, are available to make reservations for a MH, still the following can occur during the handoff or switch mechanism:

- When a router requests information from a QoS table, no longer a next hop on a path with enough available bandwidth may be found;
- When a router wants to make reservations on a link, the available bandwidth on the link may no longer be sufficient.

When such an error is detected, the delivered service is switched to best-effort. These errors can occur when the QoS tables of the AR are not up to date. This is the case when the AR has not the most recent link state information, due to the fact that the propagation of link state information through the network requires some time. Even when the correct link state information is already available, the AR may not have recalculated the QoS tables yet, due to the fact that these tables are calculated periodically or after a number of link state changes.

Moreover, even when the QoS tables are up to date at the moment of handoff, errors can occur as several MHs can perform handoff more or less at the same time. Especially in the case of mobility, the latter reason causes the major part of the errors. If ARs broadcast beacons to trigger handoff at the MHs, the sending of beacons influences the occurrence of errors:

- MHs newly arriving in the same cell, send a handoff request when a beacon from that AR is received, resulting in simultaneous handoffs;
- MHs in different cells, performing handoff more or less at the same time, can cause errors on the links of the access network.

5 Evaluation

The statements in Sects. 3 and 4 are supported by the evaluation results in this section. The network simulator ns-2 [12] is used, with Q-MEHROM as micromobility and reservation mechanism and QOSPF as routing agent [13]. However, the conclusions are not restricted to these protocols. In what follows, the results are average values of a set of 200 independent simulations of each 1800 s, i.e. there is no correlation between the sending of beacons by the ARs, the movements of the MHs and the arrival of data packets in the ARs.

The following parameter values are chosen: *1)* The wired links of the access network have a delay of 2 ms and a capacity of 2.5 Mb/s. For the wireless link, IEEE 802.11 is used with a physical bitrate of 11 Mb/s. *2)* Every AR broadcasts beacons at fixed time intervals of 1.0 s. These beacons are Mobile IP Agent Advertisements. The distance between two adjacent ARs is 200 m, with a cell overlap d_o of 30 m. All ARs are placed on a straight line. Eight MHs, labeled 1 to 8 in Fig. 3, move at a speed v_{MH} and travel from one AR to another, maximizing the overlap time to d_o/v_{MH}. *3)* CBR (constant bit rate) data traffic patterns are used, with a bitrate of 0.5 Mb/s and a packet size of 1500 bytes. For each MH, one UDP connection is set up between the sender (a fixed host in the core network) and the receiver (the MH). The requested resource is thus an amount of bandwidth of 0.5 Mb/s. *4)* Tree, mesh and random topologies are investigated. The simulated topologies are given in Fig. 2.

Using these values, the access network is highly loaded and the accuracy of the information in the QoS tables becomes important. As the wired links have a capacity of 2.5 Mb/s and a small part of this bandwidth is reserved for control traffic, only 4 reservations of 0.5 Mb/s can be made on a link. In the tree topology, the link capacities closest to the GW form the bottleneck for the number of MHs that can make a reservation. In the meshed and random topology, Q-MEHROM is able to take advantage of the extra links to offer QoS

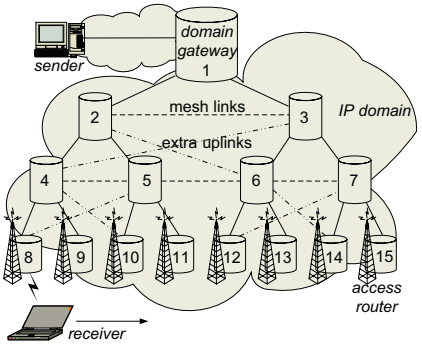


Fig. 2. Simulated access network topologies. The mesh topology consists of the tree structure (full lines) with the indicated additional mesh links (dashed lines). The random topology is formed by adding extra uplinks (dotted lines) to the mesh topology

service to more MHs. The capacity of the links closest to the ARs becomes then more important.

5.1 Extended Admission Control Mechanism

To investigate the performance of the extended admission control mechanism, proposed in Sect. 3, the fraction of time that a MH receives QoS service is measured. At the end of each handoff or switch mechanism, the current AR sends a Mobile IP Registration Reply to the MH. This message is extended with a flag r, indicating whether the requested resources were successfully reserved ($r = 0$ means best-effort service, $r = 1$ means QoS service).

We define $T_{r=1}$ as the fraction of the total simulation time that a MH receives QoS service. $T_{r=1}$ is then given by formula (1). At moment t_i, the MH receives the flag r_i of the i^{th} handoff, with H the total number of handoffs. At t_0 the MH performs power up and at t_{end} the simulation ends. Although the effects during the handoff and switch mechanisms are neglected, this metric gives a good idea of the fraction of time that QoS service is obtained.

$$T_{r=1}(\%) = \frac{(t_1 - t_0)r_0 + \sum_{i=1}^{H-1}(t_{i+1} - t_i)r_i + (t_{end} - t_H)r_H}{(t_{end} - t_0)} \quad . \tag{1}$$

The use of only the handoff mechanism (left figure of Fig. 3) is compared with the use of the extended admission control mechanism, i.e. both the handoff and switch mechanism (right figure). The extended admission control mechanism clearly results in a better performance. The improvement is most significant for the tree topology, as only a few MHs can receive QoS service at the same time. Therefore, there is an important chance that at the time of handoff, not enough resources are available and the MHs highly benefit from the switch mechanism to make reservations when other MHs perform handoff and release their resources. For the mesh and random topologies, more MHs can receive QoS service at the same time. Even then, the MHs benefit from the use of the switch mechanism.

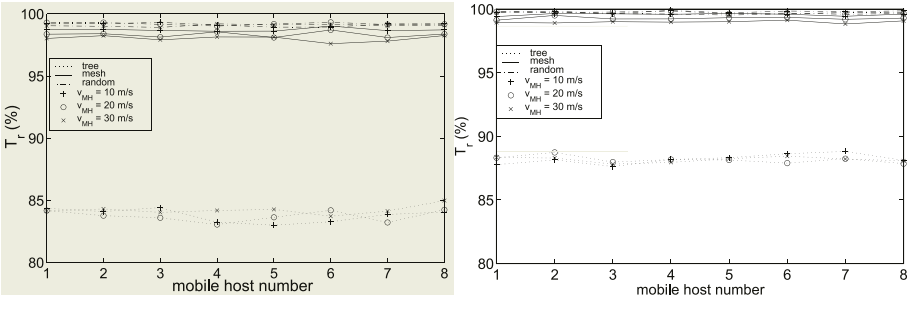


Fig. 3. Fraction of time that resources are reserved for each MH. In the left figure, only the handoff mechanism is used. In the right figure, the extended admission control is used

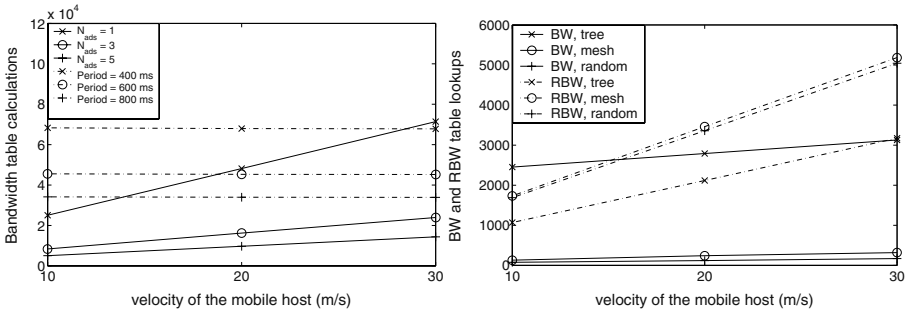


Fig. 4. Average number of Bandwidth table calculations (left figure) for the tree topology. The results for the mesh and random topology are very similar. Average number of lookups for the Bandwidth (BW) and Reuse-Bandwidth (RBW) table (right figure)

5.2 Use of Inaccurate QOSPF Information

Calculation of the QoS Tables. The information in the QoS tables should be as accurate as possible. The Delay table needs no frequent recalculation, as the link delays are not expected to change frequently. The available bandwidth on the links changes frequently, due to the bandwidth reservations and releases by Q-MEHROM. As QOSPF needs additional input information about the old path of a specific MH, the Reuse-Bandwidth table can only be calculated on demand at the time of handoff. However, the different recalculation mechanisms, explained in Sect. 4, can be applied to the Bandwidth table.

The left figure of Fig. 4 gives the average number of Bandwidth table calculations during one simulation. When N_{ads} or P increases, the number of calculations decreases, as expected. For a given value of N_{ads}, the amount of calculations depends upon the MH's velocity: for higher velocities, the number of handoffs increases which results in more bandwidth reservations and releases, thus more link state changes. The right figure gives the average number of Bandwidth and Reuse-Bandwidth table lookups, which equals the number of on demand calculations. The Bandwidth table is used when no previously reserved resources must be taken into account. This includes the path setup at power up and the switching from best-effort services to QoS or vice versa. Only for the tree topology, the Bandwidth table is frequently used, as only a few MHs can make reservations at the same time and thus a MH's service switches more often from best-effort to QoS and back.

Inaccurate Information. Figure 5 shows the average number of errors during one simulation for the different recalculation mechanisms of the Bandwidth table. As explained in Sect. 4, an error occurs when a router fails to find a next hop or fails to make a reservation. The cause of such an error stems from the fact that the QOSPF information, used by the AR to perform admission control, was not up to date or became incorrect shortly afterwards due to the handoff

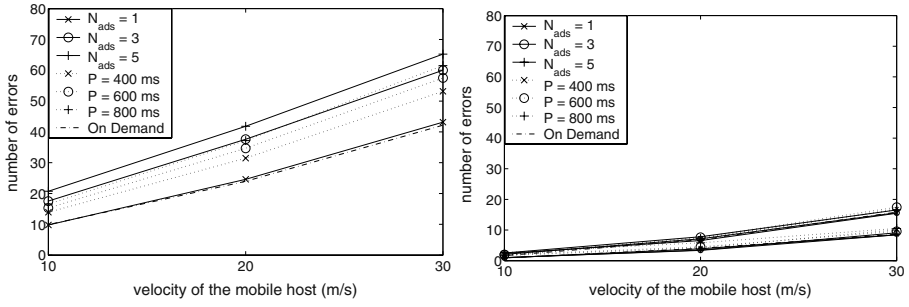


Fig. 5. The left figure gives the average number of errors for the tree topology. The right figure gives the results for the mesh and random topology, the upper bundle of curves applies to the mesh topology, the lower bundle to the random topology

of other MHs. When an error is detected, Q-MEHROM switches to best-effort service.

For the tree topology (left figure of Fig. 5), the parameters N_{ads} and P have an important impact. For higher values of N_{ads} and P, the probability that an AR uses inaccurate information and that an error occurs during the path setup, increases. The number of errors also increases for higher velocities as this results in more frequent admission control decisions by the ARs. Although Fig. 4 showed that for a velocity of 20 m/s, the number of calculations for $N_{\text{ads}} = 1$ equals the number for $P = 600$ ms, the number of errors is significantly higher in the latter case. The reason is that not only the amount of table recalculations but also the moment of recalculation influences the accuracy of the information at the time of admission control. In the case of a mesh and random topology (right figure), the calculation mechanism has a insignificant influence on the occurrence of errors, as Q-MEHROM does not use the Bandwidth table often. In addition, the results show much lower values, as more MHs can make a reservation at the same time, which makes the admission control decision less critical.

Simultaneous Handoffs. Even if the most accurate information ($N_{\text{ads}} = 1$ or on demand) is used by the ARs, errors can occur when multiple MHs perform handoff at the same time. Therefore, the moments that the handoff mechanisms are triggered, i.e. when the MHs receive a beacon and decide to perform handoff, influence the occurrence of errors. To study the impact of simultaneous handoffs, two situations are considered. The left figure of Fig. 6 presents the results for the situation in which all ARs send beacons at the same times. As a result, all handoffs, also handoffs in different cells, start more or less simultaneously. For the right figure, simultaneous handoffs are avoided by choosing different start times for each AR. In both cases, the Bandwidth table is calculated on demand. The figures show that ARs should not be synchronized, as this

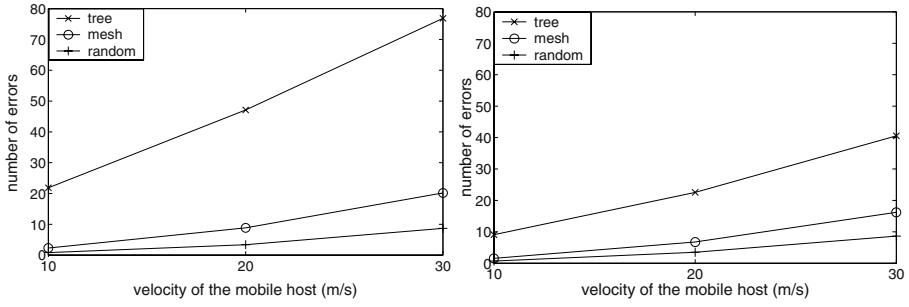


Fig. 6. The average number of errors per simulation. For the left figure, all access routers start to send beacons at 0.5 s. For the right figure, the start times for access router 1;. . . ; 8 are given by (0.125 s; 0.75 s; 0.375 s; 0.0 s; 0.5 s; 0.875 s; 0.25 s; 0.625 s)

implies that more admission control decisions are made at the same time, and the chance that an error occurs during the handoff or switch mechanism is much higher.

6 Conclusions

In this paper we presented an extended admission control mechanism, in which the MH can trigger the reservation mechanism without performing handoff, taking advantage of resources, released by the handoff of other MHs. We also studied the impact inaccurate link state information and simultaneous handoffs on the occurrence of errors during the handoff or switch mechanism.

Simulation results showed that the use of the extended admission control mechanism increases the fraction of time that the MHs receive QoS service. Mechanisms to calculate the Bandwidth table periodically or after a number of link state changes, reduce the accuracy of the information and increase the occurrence of errors. However, even if the Bandwidth table is calculated on demand, errors can be caused by simultaneous handoffs. Therefore, if the handoff is triggered after the reception of a beacon from a new AR, the ARs should not be synchronized. The impact of the extended admission control mechanism and the use of inaccurate information is higher for situations where only a small part of the MHs can receive QoS service at the same time.

Acknowledgments

Part of this research is funded by the Belgian Science Policy Office (BelSPO, Belgium) through the IAP (phase V) Contract No. IAPV/11, and by the Institute for the promotion of Innovation by Science and Technology in Flanders (IWT, Flanders) through the GBOU Contract 20152 “End-to-End QoS in an IP Based Mobile Network”.

References

1. Perkins, C. (ed.): IP mobility support for IPv4. IETF RFC 3344, August 2002
2. Johnson, D., Perkins, C., Arkko, J.: Mobility support in IPv6. IETF RFC 3775, June 2004
3. Valkó, A.: Cellular IP: a new approach to internet host mobility. ACM Computer Communication Review, January 1999
4. Ramjee, R., La Porta, T., Salgarelli, L., Thuel, S., Varadhan, K.: IP-based access network infrastructure for next-generation wireless data networks. IEEE Personal Communications, August 2000, pp. 34-41
5. Gustafsson, E., Jonsson, A., Perkins, C.: Mobile IPv4 Regional Registration. draft-ietf-mip4-reg-tunnel-00.txt, November 2004 (work in progress)
6. Peters, L., Moerman, I., Dhoedt, B., Demeester, P.: MEHROM: Micromobility support with efficient handoff and route optimization mechanisms. 16^{th} ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems (ITCSS16 2004), pp. 269-278
7. Moon, B., Aghvami, A.H.: Quality-of-Service mechanisms in all-IP wireless access networks. IEEE Journal on Selected Areas in Communications, June 2004, Vol. 22, No. 5, pp. 873-887
8. Manner, J., Fu, X.: Analysis of existing quality of service signaling protocols. draft-ietf-nsis-signalling-analysis-05.txt, December 2004 (work in progress)
9. Chaskar, H. (ed.): Requirements of a quality of service (QoS) solution for Mobile IP. IETF RFC 3583, September 2003
10. Peters, L., Moerman, I., Dhoedt, B., Demeester, P.: Q-MEHROM: Mobility support and resource reservations for mobile hosts in IP access networks. 3^{rd} International Workshop on QoS in Multiservice IP networks (QoS-IP 2005), February 2005, pp. 495-508, LNCS3375
11. Apostolopoulos, G., Williams, D., Kamat, S., Guerin, R., Orda, A., Przygienda, T.: QoS routing mechanisms and OSPF extensions. IETF RFC 2676, August 1999
12. NS-2 Home Page, www.isi.edu/nsnam/ns
13. QoSR in ns-2, www.netlab.hut.fi/tutkimus/ironet/ns2/ns2.html

Comparison of Signaling and Packet Forwarding Overhead for HMIP and MIFA

Ali Diab, Andreas Mitschele-Thiel,
and René Böringer

Ilmenau University of Technology,
Chair for Integrated HW/SW-Systems,
Ilmenau, Germany
{ali.diab, mitsch, rene.boeringer}@tu-ilmenau.de

Abstract. Handoff latency affects the service quality of real-time applications. In this paper we develop an analytical model to analyze the Mobile IP Fast Authentication protocol (MIFA) and compare it to Hierarchical Mobile IP (HMIP). The study compares the signaling costs of the protocols as well as the overall load for packet forwarding. Our study shows that MIFA minimizes the packet delivery cost compared to HMIP. Additionally, MIFA is more efficient when the arrival rate of the packets increases. Thus MIFA outperforms HMIP with respect to signaling cost. From the performance point of view MIFA performs similar to HMIP when the domain consists of two hierarchy levels only, and outperform HMIP otherwise. However, MIFA does not require a hierarchical network architecture as HMIP does.

1 Introduction

The real time applications are highly affected by the disruption in the communication during the movement from one cell to another. As the user mobility of IP-based mobiles increases and the cell size of the system decreases, handoffs will cause frequent service interruptions. Therefore, the development of fast mobility management solutions is a big challenge in future IP-based mobile networks.

When the **M**obile **N**ode (MN) notices that the current **A**ccess **P**oint (AP) is no longer reachable, it starts to scan the medium for other available APs. After that the MN authenticates and re-associates itself with the newly discovered AP. These procedures are called layer2 handoff. No additional procedures are required if the new AP belongs to the same subnet as the old one. However, the MN must discover the new **F**oreign **A**gent (FA) serving this subnet, register and authenticate itself with the **H**ome **A**gent (HA) or another special agent through this FA, if the new AP belongs to another subnet. These additional procedures are called layer3 handoff.

The rest of the paper is organized as following: In section 2 we provide the background and the related work. The MIFA protocol is described in section 3. The analytical model to derive the signaling cost and the analysis is given in section 4. After that we conclude with the main results and the future work in section 5.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 12–21, 2005.

2 Background and Related Work

In order to implement the layer3 handoff, several protocols have been proposed. With Mobile IP version 4 (MIPv4) [1], [2] or version 6 (MIPv6) [3], the MN has to be registered and authenticated by the HA every time it moves from one subnet to another. This introduces extra latency to the communication, especially when the HA is far away from the FA. Additionally, the generation of secret keys [4] for the security association is another reason for latency. Even though this is optional with MIP, it is highly recommended for security reasons. In addition, these keys are mandatory for some extensions of MIP, e.g. MIP with routing optimization [5]. Thus these two protocols are suitable for the management of global (macro) mobility.

In order to avoid these sources of extra latency, several approaches have been proposed to support local (micro) mobility. In [6] an approach to use an Anchor FA (AFA) has been proposed. If the MN is away from the home network, it will be initially registered by the HA. During this registration a shared secret between the MN and the FA is established. The FA then acts as an AFA. Thus, in subsequent registrations, the MN is registered at this AFA instead of the HA as long as it remains in the same domain. In this approach an additional tunnel from the AFA to the current FA is established to forward the packets to the MN. However, the forwarding delay on downlink as well as on uplink increases compared to MIP. An additional reverse tunnel is needed from the current FA to the AFA. Additionally a tunnel from the previous FA to the current FA is required in case the smooth handoff is supported [7].

In [8] a Regional Registration for MIPv4 and in [9] a Hierarchical Mobile IPv6 have been proposed. With these protocols the HA is not aware of every change of the point of attachment. This is due to the fact that the handoff procedures will be processed locally by a special node, e.g. a **G**ateway **F**oreign **A**gent (GFA) / **M**obility **A**nchor **P**oint (MAP), instead of the HA when the MN moves inside a certain domain. Thus, the MN communicates with the HA only if it changes this special node. However these protocols need a hierarchical network architecture.

Proposals for low latency handoffs use a trigger originating from layer2 (L2-trigger) to anticipate handoffs prior to a break of the radio link. In [10] methods for pre-registration, post-registration and a combined method have been proposed. Thus, a layer3 handoff is triggered by a L2-trigger. With the pre and post-registration method, the MN scans the medium for other APs if the strength of the signal received from the current AP deteriorates or if the error rate increases. If another AP is available and this AP belongs to another subnet, a L2-trigger is fired. This trigger contains the IP address of the new FA or another address from which the IP address can be derived, e.g. the MAC address. This prompts the MN, when employing pre-registration, to register with the new FA through the old one. Thus, the layer3 handoff is performed while the MN performs layer2 handoff. However, with post registration the MN performs only a layer2 handoff when the L2-trigger is fired. If the link between the current FA and the MN breaks down (receiving **L**ayer**2** **L**ink **D**own trigger (L2-LD) trigger), a bidirectional tunnel is established between the old FA and the new one. As a result the packets destined to the MN will be forwarded to the nFA through the old one. Thus, the MN receives the packets before the registration. With the

combined method, the MN first tries to use the pre-registration method when a L2-trigger is received. If this fails, the MN employs the post-registration method.

Performance studies and an implementation of the pre-registration and post-registration method are described in [11] and [12] respectively. A comparison between the two methods is presented in [13]. The simulation results indicate that the timing of the trigger has a major influence on the handoff latency as well as the packet lose rate. Increased latency results if the L2-trigger for pre-registration is delayed. In case the **Reg**istration **Requ**e**st** (RegRqst) is dropped, it is possible that this method resorts to the standard layer3 handoff methods, e.g. MIP or HMIP. Even though post-registration is faster than pre-registration, the impact of delayed L2-triggers with post-registration is the same as with pre-registration. Due to the missing MIP registration with the post-registration approach, the packet delay is larger (uplink and downlink).

The **S**eamless **M**obile **IP** (S-MIP), proposed in [14] reduces the required registration time through the use of hierarchical network architecture. Additionally it uses the layer2 information to accelerate the layer3 handoff. S-MIP introduces a new entity called **D**ecision **E**ngine (DE) to control the handoff process. When the MN reaches the boundary of the cell, it informs the current **A**ccess **R**outer (AR) about the movement and about the addresses of the newly discovered ARs. The current AR informs the DE and the nARs about the movement. After that the movement of the MN will be tracked by the DE to decide accurately to which AR the MN will move. When the DE determines the new AR it informs the old AR and the other participating ARs about the decision. The packets will be forwarded then to the old and to the new AR until the DE is informed from the new AR that the MN has finished the handoff.

3 Mobile IP Fast Authentication Protocol

In order to avoid the problems of MIP without needing to insert intermediate nodes between the FA and the HA, **M**obile **IP** **F**ast **A**uthentication protocol (MIFA) [15] has been proposed. The basic idea of MIFA is that the HA delegates the authentication to the FA. As a result the MN authenticates itself with the FA and with the HA. However this authentication happens in the FA. Thus the MN sends RegRqst to the FA, which in turn directly replies by sending a **Reg**istration **Reply** message (RegRply) to the MN. After receiving the RegRply, the MN can resume the transmission on the uplink. In downlink a tunnel is established to forward the packets, arriving at the previous FA, to the current FA until the HA is informed about the movement and a tunnel from the HA to the current FA is established to forward the packets directly to the current FA. Thus the delay experienced from the communication between the current FA and the HA is eliminated, similar to the micro mobility protocols, see [16].

The local authentication by FAs relies on groups of neighbouring FAs. Each FA defines a set of neighbouring FAs called a Layer3 Frequent Handoff Region (L3-FHR) [17]. These L3-FHRs can be built statically by means of standard algorithms (e.g. neighbour graph [18] or others [17]), or dynamically by the network itself, by

observing the movements of MNs. Every FA defines its own L3-FHR. There is a security association between the FAs in each L3-FHR. This security association can be established statically, e.g. by the network administrator, or dynamically, e.g. by the network itself as described in [4], [5].

Figure 1 depicts the basic operation of MIFA. While the MN communicates with the current FA, this FA sends notifications to all of the FAs in the L3-FHR the current FA belongs to. These notifications contain the security associations between the MN and the FAs in this L3-FHR on one side and between the FAs and the HA on the other side. These security associations are recorded in soft state and will be used by one FA at the future and deleted from the others. Additionally these notifications contain the characters of the HA and the authentication values (between the MN and the HA) the MN has to generate in the next registration with the next FA. These notifications are authenticated by means of the security associations established between the FAs, for more details see [15].

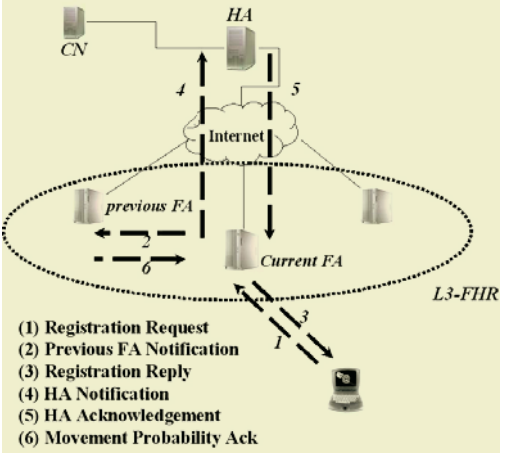


Fig. 1. Basic operation of MIFA

When the MN moves to one of the FAs in the L3-FHR, to which the previous FA belongs to, it sends RegRqst message to this FA. This current FA checks at first the authentication between it and the MN, this authentication will be checked by using the security association sent from the previous FA with the notification. After that the current FA checks the MIFA information, which presents the authentication information between the MN and the HA. The current FA then checks if the requirements requested from the HA can be satisfied, this can be achieved through the check of the HAs characters sent with the notification too. If the authentication succeeds, the FA builds a Previous FA Notification message to inform the previous FA that it has to forward the packets, sent to the MN, to the current FA. After that the current FA sends Registration Reply to the MN, at this time the MN can resume transmission in uplink. Additionally the current FA sends a HA Notification message

to inform the HA about the new binding, the HA in turn establishes a new tunnel to the new FA, after that it intercepts the packets forwarded to the old binding and tunnels them to the new one. Thus the time to inform the HA about the new binding and to establish a new tunnel is eliminated.

In [16] an analytical model to evaluate the performance of MIFA compared to HMIP. This analysis shows that the handoff latency by MIFA is independent of the distance between the current FA and the HA. MIFA performs similar to HMIP when the domain consists of two levels of the hierarchy and outperforms HMIP otherwise. The main advantage of MIFA is that MIFA does not require a hierarchical network architecture as HMIP does. Additionally, MIFA process the handoff procedures locally without introducing any intermediate node between the FA and the HA. Thus MIFA is a protocol to manage the global mobility, same as MIP, locally, same as HMIP.

4 Signaling Cost Function

In this section we will derive the signaling cost function to evaluate the impact of MIFA on the network and compare it to HMIP. The total signaling costs comprise the location update function and the packet forwarding function. We neglect the periodic bindings sent from the MN to refresh the cache in the mobility agents. The total cost will be considered as the performance metric.

4.1 Basic Assumptions

In order to model the signaling cost function we suppose that the MN moves within two domains, domain a and b. Each domain contains 9 mobility agents. To model HMIP, these mobility agents are structured in hierarchical manner. To model the MIFA protocol, the mobility agents are divided into L3-FHRs. For simplicity we use a pre-defined mobility model for both cases, which presents the movement between the mobility agents in an active session. We assume that the MN moves along the path shown in Figure 2. The time that the MN will spend inside the region of each

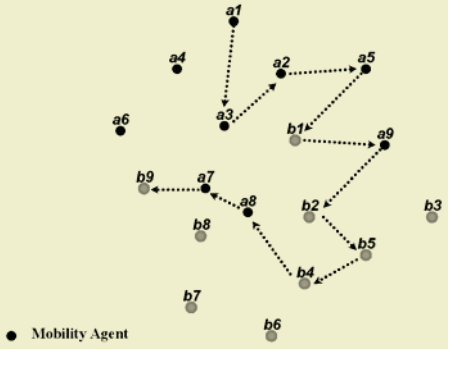


Fig. 2. Locations of the mobility agents

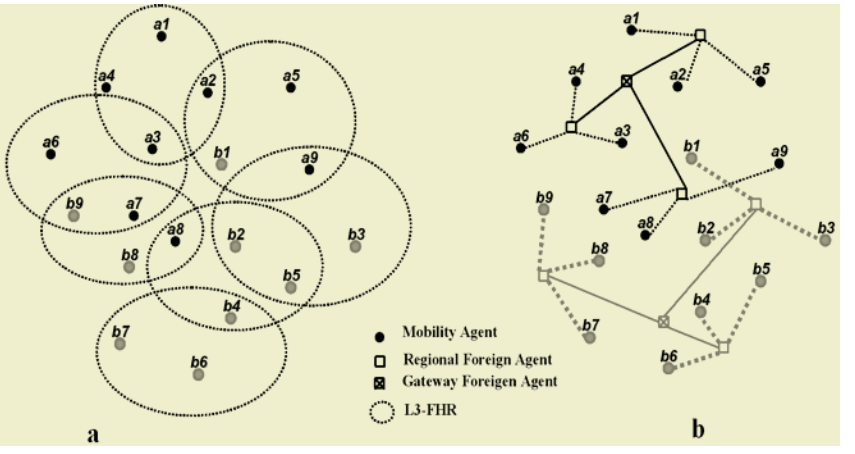


Fig. 3. Network topology

mobility agent is randomly with average T_a. Figure 3b depicts the network topology in the case of HMIP. Each domain contains 9 mobility agents where sets of three mobility agents are controlled by a **R**egional **F**oreign **A**gent (RFA). The RFAs are controlled by a GFA. Figure 3a plots the mobility agents divided into L3-FHRs for the case of MIFA.

4.2 Location Update Cost

We define the following parameters to compute the location update cost: $C_{x,y}$ is the transmission cost of the location update between node x and node y. $D_{x,y}$ denotes the distance between the two nodes x and y with respect to the number of the hops. a_x represents the processing cost of the location update at node x.

4.2.1 Location Update Cost for HMIP

The location update cost function $C_{HMIP\text{-}LU}$ in case of HMIP is given in equation (1):

$$C_{HMIP\text{-}LU}=M^* C_{HMIP\text{-}LU\text{-}HA} + N^* C_{HMIP\text{-}LU\text{-}GFA} + G^* C_{HMIP\text{-}LU\text{-}RFA} \quad (1)$$

$C_{HMIP\text{-}LU\text{-}HA}$ is defined in equation (2) and denotes the location update cost when the MN registers with the HA. M represents the number of home registrations done by the MN while moving on the defined path. $C_{HMIP\text{-}LU\text{-}GFA}$ is defined in equation (3) and represents the location update cost when the MN registers with the GFA. N presents the number of times the MN has registered with the GFA. $C_{HMIP\text{-}LU\text{-}RFA}$ is given in equation (4) and expresses the location update cost when the MN registers with the RFA. G denotes the number of registrations with the RFA.

$$C_{HMIP\text{-}LU\text{-}HA}=2^*(C_{MN,FA}+C_{FA,RFA}+C_{RFA,GFA}+C_{GFA,HA})+2^*(a_{FA}+a_{RFA}+a_{GFA})+a_{HA} \quad (2)$$

$$C_{HMIP\text{-}LU\text{-}GFA}=2^*(C_{MN,FA}+C_{FA,RFA}+C_{RFA,GFA}) + 2^*(a_{FA}+a_{RFA}) +a_{GFA} \quad (3)$$

$$C_{HMIP\text{-}LU\text{-}RFA}=2^*(C_{MN,FA}+C_{FA,RFA}) + 2^*a_{FA} + a_{RFA} \quad (4)$$

The transmission cost $C_{x,y}$ on the wired network is proportional to the distance $D_{x,y}$ with proportional constant d_D. Thus we can write

$$C_{x,y} = d_D * D_{x,y} \tag{5}$$

while this cost on wireless link is z times more than on the wired one. Thus, we can derive this cost from equation (6):

$$C_{MN,FA} = z * d_D \tag{6}$$

4.2.2 Location Update Cost for MIFA

The location update cost $C_{MIFA\text{-}LU}$ using MIFA can be derived from equation (7):

$$C_{MIFA\text{-}LU} = B*(2*(C_{MN,FA} + C_{FA,oFA} + C_{FA,HA}) + 3*a_{FA} + a_{oFA} + a_{HA}) \tag{7}$$

where B denotes the number of the registrations the MN has executed.

4.3 Packet Delivery Cost

In order to evaluate the packet delivery cost, we assume the following parameters: $T_{x,y}$ denotes the transmission cost of the packet delivery between node x and node y. v_x represents the processing cost of the packet delivery at node x.

4.3.1 Packet Delivery Cost for HMIP

When using HMIP, the packets will be forwarded from the **C**orresponding **N**ode (CN) to the HA, which forwards them to the GFA. The GFA in turn forwards these packets to the serving RFA, which forwards them to the current FA. The current FA sends the packets then to the MN. Thus, there is extra cost for the packet delivery. We consider the packet delivery cost a packet incurs on its path from the CN to the MN. The packet delivery cost function $C_{HMIP\text{-}PD}$ is given in equation (8):

$$C_{HMIP\text{-}PD} = v_{HA} + v_{GFA} + v_{RFA} + v_{FA} + T_{CN,HA} + T_{HA,GFA} + T_{GFA,RFA} + T_{RFA,FA} \tag{8}$$

The transmission cost $T_{x,y}$ is proportional to the distance $D_{x,y}$ with proportional constant d_U. Thus we can write

$$T_{x,y} = d_U * D_{x,y} \tag{9}$$

The load on the GFA for the processing depends on the number of MNs in the domain and on the number of RFAs beneath it, while the load on the RFA depends on the number of MNs served from this RFA and on the number of FAs beneath it. Supposing the number of MNs in each subnet is w, the number of FAs served by each RFA is k, which equals the number of RFAs served by the GFA. The IP routing table lookup is based normally on the *longest prefix matching*. Thus, for the traditional *Patricia trie* [19], the packet processing cost functions in the GFA and the RFA can be computed from equations (10) and (11) respectively.

$$v_{GFA} = l_1 * k * L_a * (q_1 * w * K^2 + g_1 * \log(k)) \tag{10}$$

$$v_{GFA} = l_2 * k * L_a * (q_2 * w * K + g_2 * \log(k)) \tag{11}$$

where, L_a is the arrival rate of the MN. q and g are weighting factors of the router visitor list and the table lookups. l is a constant expressing bandwidth allocation cost.

The processing cost at HA and FA, which results from the encapsulation and decapsulation of packets, can be derived from the equations (12) and (13), respectively.

$$v_{HA} = y_1 * L_a \tag{12}$$

$$v_{FA} = y_2 * L_a \tag{13}$$

where, y_1, y_2 are constants expressing the packet delivery cost at HA and FA.

4.3.2 Packet Delivery Cost for MIFA

The packet delivery cost using MIFA can be computed from equation (14).

$$C_{MIFA\text{-}PD} = v_{HA} + v_{FA} + T_{CN,HA} + T_{HA,FA} \tag{14}$$

4.4 Total Signaling Cost

The total signaling cost is the sum of the location update cost and the packet delivery cost. Thus, we can write:

$$C_{HMIP} = C_{HMIP\text{-}LU} + C_{HMIP\text{-}PD} \tag{15}$$

$$C_{MIFA} = C_{MIFA\text{-}LU} + C_{MIFA\text{-}PD} \tag{16}$$

4.5 Analytical Model

In order to analyze and to compare the two protocols, we assume that the costs for the packet processing at the mobility agents are available. a_{FA}, a_{RFA}, a_{GFA} and a_{HA} can be seen as the time required to process the location update message in FA, RFA, GFA and HA respectively. d_D and d_U present the delay required to send the location update message and the data message for a one hop. These values can be derived from the network by deploying certain measurements. l_1, l_2 ,k, q_1, q_2 , g_1, g_2 , y_1 and y_2 are designed values. Table 1 lists the used parameters in this model:

Table 1. The parameters used in this model

a_{FA}	a_{oFA}	a_{RFA}	a_{GFA}	a_{HA}	l_1	l_2	q_1	q_2	g_1	g_2	d_D	d_U	K
10 µsec	10 µsec	15 µsec	20 µsec	25 µsec	0,01	0,01	0,3	0,3	0,7	0,7	0,5 msec	1 msec	3

$D_{CN,HA}$	$D_{HA,GFA}$	$D_{GFA,RFA}$	$D_{RFA,FA}$	$D_{FA,oFA}$	$D_{HA,FA}$	T_a	w	y_1	y_2	z
10	10	2	2	2	10	10	25	10 µsec	10 µsec	10

We define *CMR* as the Call to Mobility Ratio, which expresses the ratio of the packet arrival rate to the mobility rate. Thus, we can write

$$CMR = T_f * L_a \quad (17)$$

Where, T_f is the residence time in the region of a certain mobility agent. Figure 4 depicts the packet delivery cost in a time unit as a function of *CMR*. The arrival rate L_a varies inside the range from 1 to 1000 packet per second. From this figure we notice that MIFA is more efficient than HMIP especially for large *CMR* values. Thus, MIFA is more adequate for real-time applications.

In the following, we will try to observe the total signalling cost of a session. We assume that the session will take 80 sec. The total signalling cost for the two protocols is depicted in figure 5. We can clearly see that MIFA outperforms HMIP.

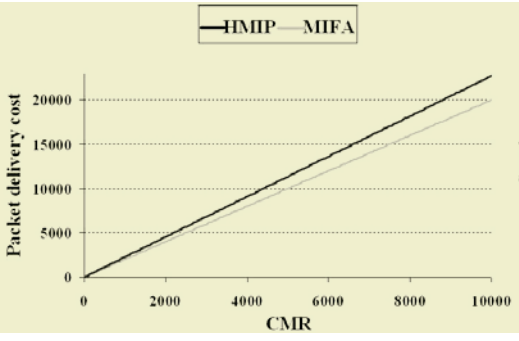


Fig. 4. Packet delivery cost

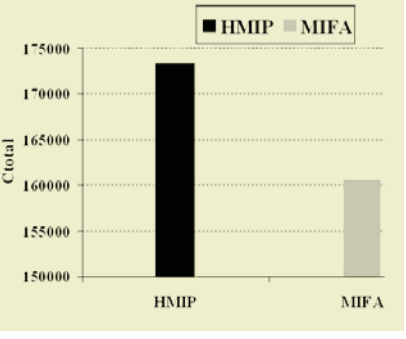


Fig. 5. Total signalling cost

5 Conclusion

In the paper we have designed an analytical model to evaluate the signaling cost of MIFA compared to HMIP. Our analysis shows that MIFA outperforms HMIP with respect to signaling cost. This is because MIFA eliminates the packet delivery costs resulting from the triangular routing by HMIP (from HA via GFA and RFA to the current FA). Additionally, the handoff latency using MIFA does not depend on the distance between the current FA and the HA, similar to HMIP. Thus MIFA performs similar to HMIP when the MN moves within a domain consisting of two hierarchy levels only and outperforms HMIP otherwise. Thus MIFA is more adequate for the real-time applications. Currently, we develop an analytical model to evaluate the signaling costs when using another mobility models and we try to determine the impact of the other parameters on the signaling cost.

References

1. C.E. Perkins: MOBILE IP - Design Principles and Practices. 1998.
2. C. Perkins, Ed: IP Mobility Support for IPv4. RFC: 3344 August 2002.
3. D. Johnson, C. Perkins, J. Arkko: Mobility Support in IPv6. < draft-ietf-mobileip-ipv6-23.txt >, October 2003.

4. C.E. Perkins, P.R. Calhoun: Generalized Key Distribution Extensions for Mobile IP. < draft-ietf-mobileip-gen-key-00.txt >. 2 July 2001.
5. D. B. Johnson, N. Asokan: Registration Keys for Route Optimization. < draft-ietf- mobileip-regkey-03.txt >. 14 July 2000.
6. G. Dommety, Tao Ye: Local and Indirect Registration for Anchoring Handoffs. < draft-dommety-mobileip-anchor-handoff-01.txt >. July 2000.
7. C.E. Perkins, K.-Y. Wang: Optimized Smooth Handoffs in Mobile IP. Proceedings of the Fourth IEEE Symposium on Computers and Communications, July 1999.
8. E. Gustafsson, A. Jonsson, Charles E. Perkins: Mobile IPv4 Regional Registration. < draft-ietf-mobileip-reg-tunnel-08.txt >, November 2003.
9. H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier: Hierarchical Mobile IPv6 mobility management (HMIPv6). < draft-ietf-mobileip-hmipv6-08.txt >, June 2003.
10. K. El Malki et al.: Low Latency Handoffs in Mobile IPv4. draft-ietf-mobileip-lowlatency-handoffs-v4-04.txt, June 2002.
11. C. Blondia, et al.: Low Latency Handoff Mechanisms and Their Implementation in an IEEE 802.11 Network. Proceedings of ITC18, Berlin, Germany, 2003.
12. O. Casals, et al .: Performance Evaluation of the Post-Registration Method, a Low Latency Handoff in MIPV4. Proceedings of IEEE 2003 International Confernence on Communications (ICC 2003), Anchorage, May 2003.
13. C. Blondia, et al: Performance Comparison of Low Latency Mobile IP Schemes. Proceedings at WiOpt'03 Modeling and Optimization in Mobile Ad Hoc and Wireless Networks, INRIA, Sophia Antipolis, pp. 115-124, March 2003.
14. R. Hsieh, Z.-G. Zhou, and A. Seneviratne: S-MIP: A Seamless Handoff Architecture for Mobile IP. In Proceedings of INFOCOM, San Francisco, USA, 2003.
15. A. Diab, A. Mitschele-Thiel: Minimizing Mobile IP Handoff Latency. 2nd International Working Conference on Performance modelling and Evaluation of Heterogeneous Networks (HET-NETs'04), Ilkley, West Yorkshire, U.K., July 26 - 28, 2004.
16. A. Diab, A. Mitschele-Thiel, J. Xu: Perfomance Analysis of the Mobile IP Fast Authentication Protocol. Seventh ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2004), Venice, Italy, October 4-6, 2004.
17. S. Pack, Y. Choi: Fast Inter-AP Handoff using Predictive-Authentication Scheme in a Public Wireless LAN. Networks 2002, Aug. 2002.
18. S.K. Sen, et al.: A Selective Location Update Strategy for PCS Users. ACM/Baltzer J. Wireless Networks, September 1999.
19. H.-Y. Tzeng, T. Przygienda: On Fast Address-Lookup Algorithms. IEEE J. Selected Areas in Comm. (JSAC), vol. 17, no. 6, pp. 1067-1082, June 1999.

Profile System for Management of Mobility Context Information for Access Network Selection and Transport Service Provision in 4G Networks*

Ivan Armuelles Voinov[1], Jorge E. López de Vergara[2],
Tomás Robles Valladares[1], and David Fernández Cambronero[1]

[1] Dept. of Telematic Systems Engineering,
Technical University of Madrid (UPM), Madrid, Spain
{ivan, robles, david }@dit.upm.es
[2] Departmento de Ingeniería Informática,
Universidad Autónoma de Madrid (UAM), Madrid, Spain
jorge.lopez_vergara@uam.es

Abstract. High level services and data transport service provision are facing important advancements towards a more flexible business models and Internet services with the internetworking of several complementary access technologies using IP. This imposes new requirement in users' mobile terminal, such as intelligent discovery and selection of access networks, vertical handover support and roaming. The result of such integration of networks, also known as "4th Generation Networks", will require that transport network and user's terminal coordinates for providing of a "persistent" transport service. This paper presents the conceptual high level description of a context-based management distributed service and profiling system based in OWL, the W3C ontology language, in order to provide this persistent transport for mobile user's data in a heterogeneous environment of IP-based network.

1 Introduction

Lately, mobile communications are facing important advancements towards a more flexible business model and fresh Internet-like services, with the introduction of new 2.5G/3G of mobile communication systems. Additionally, several complementary access technologies are evolving, for instance, broadband WLAN and broadcast systems are becoming available; systems like Bluetooth are being developed for ad hoc and short-range connectivity. The interconnection of all these access networks (ANs) using the Internet Protocol (IP) has been envisioned as the mean by which the user will be in some sense "always best connected".

The results of such evolution, also known as "4th Generation Networks", are related to more flexible service provision characterized by dynamic user registration, flexible charging/accounting models; advanced profile management of users, networks and terminals; context aware systems and others [1].

* This work has been partially developed during the MIND and ANWIRE IST projects cofunded by the European Community.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 22–31, 2005.

This envisioned future imposes basic requirements in mobile user's terminal such as cross-layer and TCP/IP-based design; QoS, multi-homing and mobility support; adaptation and re-configuration at different level of the protocol stack, intelligent discovery and access network selection, ad hoc connectivity and roaming to other different networks [2]. In this paper we present the conceptual high level description of a context-based management distributed service and profiling system in order to provide a persistent transport for mobile user's applications in a heterogeneous environment of IP-based network.

The paper is organized as following: section 2 describes the model and concepts introduced for defining a persistent transport service in 4G scenarios; section 3 presents our perspective of the use of mobility context information management system in 4G networks and an ontology-based solution; section 4 provides the high level description of the management system; section 5 discuss different alternatives for the deployment of a mobility context management service for persistent transport provision in future scenarios; section 6 summarizes the conclusions.

2 Concepts for 4G Networks Description

In future 4G networks, it will be possible to get access to services and to use distributed applications independently of the time and the user location. When moving, a user will access different wireless services via his/her Mobile Terminal (MT). We define "wireless services" (WSs) as a set of functions and facilities related to applications and communications infraestructures offered to consumers by providers, and providing consumers with requested resources according to a service agreement.

WSs will be provided by a wide range of players which will offer their functions to other players or users. In order to avoid complex classification of these players, we differentiate them by the main facilities that they offer. These belong to the class of "infrastructure service provider" which serves data transportation (e.g., access network service and proxy service providers) or to the class of "application service provider" that correspond to upper layers.

A mobile user will access to wireless services depending on its activities and preferences and its MT capabilities. In order to make an abstraction of the relationship between the mobile user and its MT when accessing to WSs, we introduce the concept of "Mobile Customer" (MC). A MC is a specific terminal configuration and application settings according to the current user's role and features of the selected access network.

During the day, the user will develop "roles", i.e., different activities and relationships with other individuals; in each role, the user might react differently. The user's roles are represented by its respective "profiles" of WSs usage.

3 Thoughts on Context-Aware Services Context and Profiles

In future heterogeneous environments of Wireless and Mobile Networks it will be possible to find situations where the MT shall be able to take its own decisions related

to its adaptation and configuration [3]. The MT should be aware of the user activities patterns. These patterns are governed by user‘s roles which state his/her preferences when developing different activities. Such preferences and user context information are usually declared in a profile.

3.1 Context Management for Persistent Transport Service Provision

The management of the knowledge of user context aspects such as the user identity, its activity and the respective location and time is one of the main issues of “Context-aware Systems”. A system is context-aware if it uses context to provide relevant information and/or service to the user. Context is defined as “any information that can be used to characterize the situation of an entity” [4]; an entity could be any element like a person, place or object. The use of context-aware system has been traditionally focused to the adaptation of services and applications according to its location, the collection of nearby people and objects, as well as changes of the objects over time.

In the area of mobile communication services, the Open Mobile Alliance (OMA) and 3GPP had defined architectures for network services and content adaptations in the Application and Content Providers taking in consideration the capabilities of the terminal device and the user preferences [5][6].

More recently, the exchange of context information has been proposed for the proactive configuration of IP-based access network elements in order to improve the handover process of the MT. In the IETF, the Seamoby WG had proposed a Context Transfer Protocol that exchange context information such as security context, policy, QoS, header compression, and AAA information to preserve the session flow in intra-domain handover [7].

We consider that context-aware system concepts can also be exploited for the orchestration of MT configuration and adaptation when providing the “persistent transport service” to application flows.

3.2 Approaches of Context Information Representation by Profiling

Markup languages have been proposed as a basis for context information representation, based on the Resource Description Framework (RDF) or RDF Schema (RDF-S), which improves RDF semantics. For instance, the Composite Capability/Preference Profiles (CC/PP) [8] uses RDF for context information. It defines a “two-level” hierarchy of components and attribute/value pairs for describing a profile of device capabilities and user preferences that can be used to guide the adaptation of the contents presented to that device. CC/PP specification does not mandate a particular set of components or attributes. For instance, the User Agent Profile (UAProf) [5], includes hardware and software characteristics, application/user preferences, and WAP and network features.

3GPP defines a General User Profile (GUP) [6], a collection of user related data which affects the way in which an individual user experiences high level services. The GUP model information is suitable for representing our “mobile customer” concept, because its structure provides a better organization for introducing complex

relationships between elements than the RDF CC/PP approach. However, it uses XML Schema instead of RDF-S.

W3C works on semantic web languages as well, and defines the Web Ontology Language (OWL, not an acronym) [9]. An ontology can be defined as "an explicit and formal specification of a shared conceptualization" [10]. OWL expands RDF-S information model adding more vocabulary along with a formal semantics for describing properties and classes, relations between classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes. Based on this language, OWL-S [11] specifies an ontology of "services" that makes possible functionalities with a high degree of automation to discover, invoke, compose, and monitor Web resources.

After considering the benefits that OWL can offer for the context information representation by means of profiles, we have used it to define different kind of profiles (for users, MTs, access network, etc.) following the GUP structure and leveraging OWL features. Even more, we have used OWL-S as a part of the design of a manager of mobility context information that assists the MT in both the access network selection and the persistent transport service configuration.

4 Context-Aware System for Intelligent Access Network Selection and Terminal Configuration

In this section we present a high level description of the "Mobility Context Manager Service", a control plane functional component of the mobile terminal architecture proposed in section 6. It develops three functionalities: the generation of mobility context information, the access network selection and the persistent transport service configuration, i.e., the adaptation and/or re-configuration control of the user data plane when supporting new and ongoing sessions.

4.1 Mobility Context and Persistent Transport Manager Service

The description (presentation, modeling and grounding) of a distributed implementation of the Mobility Context Manager Service (MCMS) (Fig. 1) is made with OWL-S. A high level implementation of the service has been developed with the use of the OWL-S Editor [12], an extension of the Protege-2000 [13].

The first task of the MCMS is to generate the mobility context information, i.e., the available ANs and their features in the current MC location. The raw information produced by AN advertisements is transformed into AN Profiles (ANPs), which feed the Mobility Context Generator component. The last is in charge of the creation of an updated Mobility Context Profile (MCP). All profiles have been defined in OWL.

The composite process model which describes the main activities performed by the Mobility Context Generator (MCG) is shown Fig. 2.a. The MCG retrieve each ANP in a repeat-while loop. Each ANP is collocated in a provisional list (a container). Finally, the list is processed into a MCP and saved in a repository.

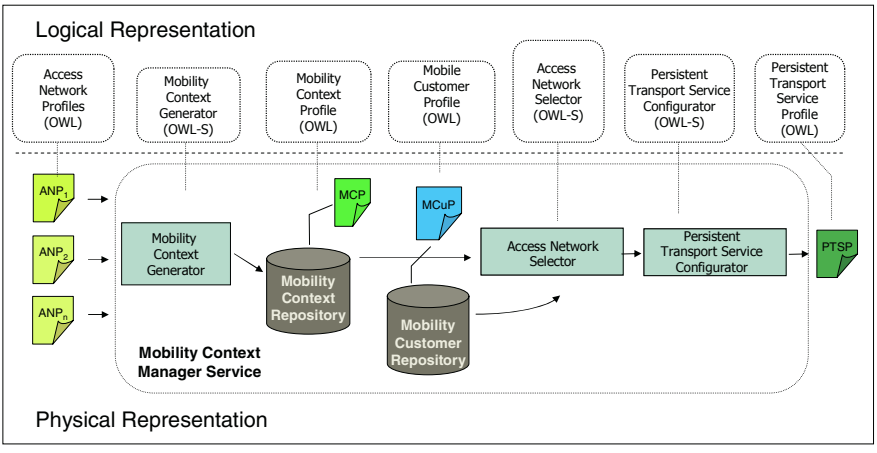


Fig. 1. Mobility Context Manager Service

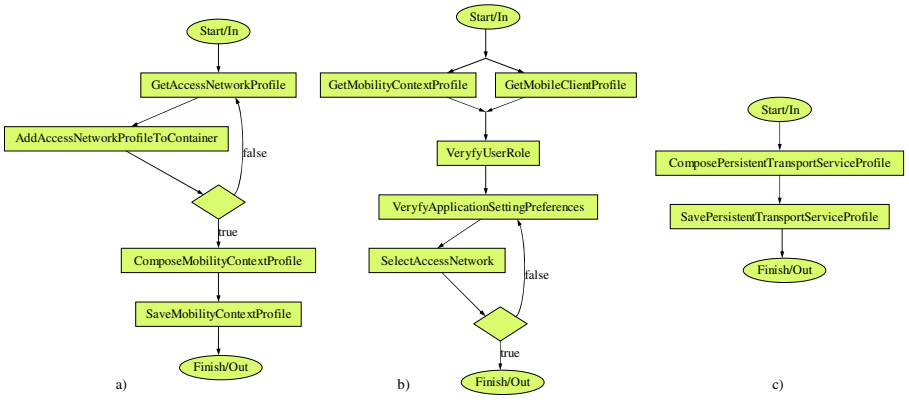


Fig. 2. Process models of Mobility Context Manager Service

The current MCP is used for the Access Network Selection (ANS) process (Fig. 2.b). The ANS is done based in the MCP and the Mobile Customer Profile (MCuP) both of them stored in their respective repositories. The ANS process gets both profile and first it evaluates the current user role and the list of application preferred settings specified in the MCuP according to the role. Then, the ANS process compares the list of application settings with the features of ANs in the MCP, until it finds a suitable network. If there is no a best AN then the ANS process resolves to select one based on default applications settings. Then, the MC establishes the AN association.

The last function of the MCMS is the Persistent Transport Service Configurator (PTSC). This component is in charge of the mobile terminal configuration when a new association will be established or there is any change in the mobility context that

can modify the current PTS behaviour. The PTSC receives a message form the ANs process advertising the association and requesting for configuration information. The PTSC replies with a Persistent Transport Service Profile (PTSP) (Fig. 2.c), which contains parameters for the configuration of different required components and functionalities of the MT protocol stack. The PTSC interacts with different management systems and servers in the access network in order to retrieve these configuration parameters.

4.2 Information Model of the Mobility Context Manager Service

We model different context-information by means of profiles. These are used as inputs and output of the described Mobility Context Manager Service.

The Access Network Profile (ANP) describe the features of one AN organized in a set of profile components groups (PCGs) (Fig. 3.a). The main features included in ANPs were derived from different reports and papers of several standardization groups, such as WWRF, IETF EAP WG, IEEE and 3GPP2, which have deliberated on AN selection issues [14][15].

Each PCG describe a feature that could be relevant during the AN selection before the association establishment: AN identification, QoS (available bandwidth), cost, security/authentication methods, list of intermediary network providers between the current visited AN and the home network, roaming agreements, middleboxes, etc. After the collection of these profiles, the MCG produces a Mobility Context Profile (MCP) which stores context-information of the surrounding heterogeneous environment of ANs. Each PCG is focused in a particular feature such as cost, QoS, etc. and the respective offers of each advertised AN (Fig. 3.b).

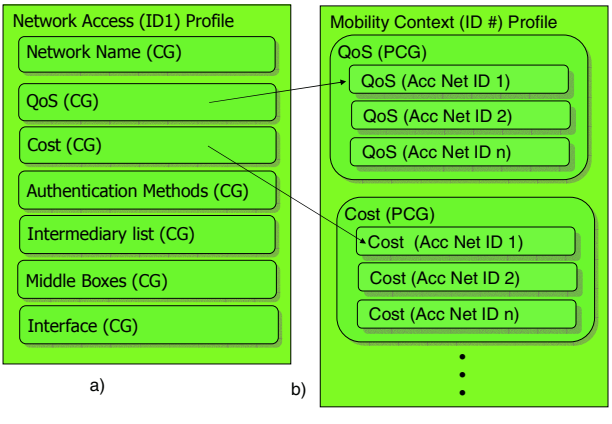


Fig. 3. Access Network Profiles and matching Mobility Context Profile

The MCP is used along with the Mobile Customer Profile (MCuP) by the Persistent Transport Service (PTS) Configurator to produce a PTS profile that guide the adaptation and re-configuration of the terminal protocol stack. The MCuP is com-

posed of a set of PCGs that store information with respect to the user and its preferences per role, default and preferred software settings and terminal hardware and software capabilities. The MCuP is the union of user, application and terminal subprofiles. The "User PCG" includes the criteria of preference used by the user in each of its roles when executing a particular application. The "Application PCG" presents the list of parameter values that must be used for the execution of an application according to the criteria established in the user role.

The last profile of our information model is the Persistent Transport Service Profile (PTSP) which consist of a series of PCGs that enumerate all the parameters that must be set per protocol layer and terminal functional component for the new (or ongoing) data session. All the profiles were edited with the OWL plug-in [16].

5 Alternative Deployments of Mobility Context Manager Services

In future 4G scenario the MC should develop for its own the ANs discovery and selection, the mobility context generation and the persistent transport service provision. This MC's autonomy will be possible only if the mobile device has enough resources and a new business model allows the free association of the MC to any ANs [17]. In this case, the MCMS could be developed in the MC.

Current business model requires the user subscription to an Access Network Provider (ANPr), who establishes the association parameters to allow the access to WSs by the MC. This association could be established out of the domain of the original ANPr by means of vertical handover mechanisms and roaming agreements with other ANPrs. In this kind of scenario, the deployment of the MCMS will require the distribution of its components among different entities as the MC, the home and visited ANPs. The Persistent Transport Service Profile (PSTP) could be generated with the collaboration of home and visited ANs and submitted to the MC for its configuration.

Fig. 4 presents a simplified sequence diagram of an scenario in which the MC has the ability to perform the Mobility Context Generation (MCG) and Access Network Selection (ANS) tasks, while the Persistent Transport Service Configuration (PTSC) is provided by an agent in the visited AN. The ANs advertise their features which are collected by the MT network interfaces. The MCG composes a mobility context profile that is used during the ANS process. In this step, the ANS process evaluates the user and application needs and compares them with the MCP selecting the best offered AN. Then, the MC associates itself with the chosen AN (sequence not shown in the figure). The MC localizes the PTSC component service in the access network following the URL announced in the AN profile, or alternatively, it could ask for it to a UDDI registry. The MT sends a message to the PTSC requesting a specific PTS profile. The PTSC interacts with other servers to collect all the configuration parameters that will be necessary to access the pretended service over the AN. Finally, MT receives an answer with the requested PTSP which allows it to access the WSs.

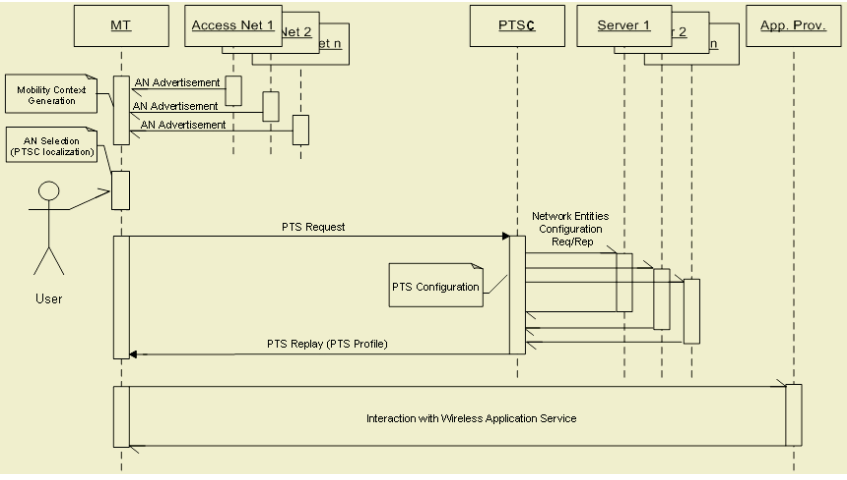


Fig. 4. Interaction between the components of Mobility Context Manager Service

To provide the required flexibility of the service deployment, we take advantage of semantic web solutions. Offering the mobility context management service at high level allows overcoming interoperability problems in heterogeneous systems.

6 Terminal Architecture Proposal for 4G Networks

We have proposed a TCP/IP-based mobile terminal architecture, suitable for future heterogeneous integrated networks, based on a generic reference model (BRENTA) [18]. The architecture is divided in two main planes, the User Data Plane and the Control Plane, as shown in Fig. 5.

In the user data plane ("or network plane"), different kind of applications from Internet legacy applications (Web, mail, etc) to adaptive and QoS-aware applications relay in a generic interface, the "Common QoS Interface" (CQoSI) [3] to request Hard and QoS support independently of the underlying QoS technologies. The "QoS Processor" translates these parameters to network level QoS parameters according to the available QoS Service Provider (QoSSP). It provides an adaptable and re-configurable transport service to active sessions during the MC displacement, and implements, maintains, and handles specific QoS signaling. The "Transport Service Provider" encompasses transport protocols (UDP or TCP), QoSSPs and networking functionalities, and is in charge of the association with access network which offer Internet connection services based on IP parameters (IP addresses, default routers, etc). An "Enhanced Network to Link layer interface" provides a standard link layer interface for different underlying link technology.

In the Control Plane, the MCMS orchestrates the functionalities and resources of the mobile terminal as commented before.

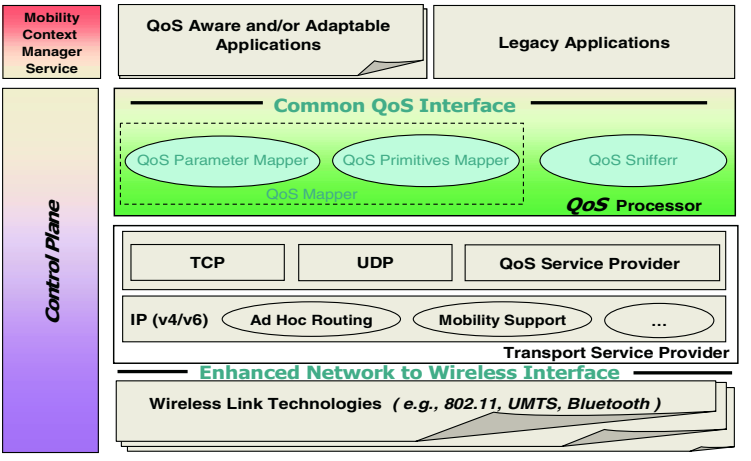


Fig. 5. Proposed TCP/IP-based Terminal architecture for 4G networks

7 Conclusions

Future 4G Networks impose fundamental requirements in mobile user's terminal such as cross-layer and TCP/IP-based design; QoS, multi-homing and mobility support; adaptation and re-configuration at different level of the protocol stack, etc. In this paper we had presented a high level description of a context-based management distributed service and profiling system proposed for allowing the user terminal to support a persistent transport service for mobile user's applications in such networks. The distributed system (the MCMS), is part of the terminal control plane and it is used for the configuration control of the terminal during vertical handover process and roaming. The MCMS is a context-based service composed of three basic components for the collection of information related to the surrounding access networks, the access network selection and the terminal configuration for persistent transport service. These components exchange context-based information of the mobile costumer in form of profiles, and we have proposed the use of ontology languages, such as OWL, to define our information model after the analysis of several approaches. Even more, we have described each service component by means of OWL-S, an ontology based on OWL to discover, invoke, compose, and monitor service resources. This method allows a faster and easer deployment of MCMS in future scenarios where mobile terminal should be more autonomous. Lastly we propose a TCP/IP-based terminal architecture for 4G networks.

References

1. Pereira, J. "European Approach to Fourth Generation –A personal perspective". International Forum on 4th Generation Mobile Communications. London, UK. May, 2002.
2. I.Ganchev et al. "Requirements for an Integrated System and Service 4G Architecture", IEEE Semiannual Vehicular Technology Conference, May 17-19, 2004, Milan, Italy.

3. I. Armuelles, T. Robles, D. Fernández. “Componentes Funcionales de Middleware para un Terminal Móvil de Redes de Comunicaciones Móviles de 4G”. XIX Simposium Nacional de la URSI. Barcelona. España. Septiembre, 2004.
4. Dey, A. K. (editor), "Providing architectural support for building context-aware applications," Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Georgia, USA, 2000.
5. “User Agent Profile”. Version 2. Open Mobile Alliance. May, 2003.
6. “3GPP Generic User Profile (GUP); Stage 2; Data Description Method; (Release 6)”. Technical Specification. 3GPP TS 23.241 V1.0.0. December, 2003.
7. J. Loughney (editor) et al. “Context Transfer Protocol”. Internet Draft. IETF Seamoby WG. August 2004.
8. “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0”. W3C Recommendation 15 January 2004.
9. D. L. McGuinness et al. “OWL Web Ontology Language Overview”. W3C Recommendation. February, 2004.
10. R. Studer, V.R. Benjamins, D. Fensel: Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering. 25. (1998) 161-197
11. “OWL-S: Semantic Markup for Web Services”. W3C Member Submission 22 November 2004
12. G. Denker, D. Elenius, and D. Martin. “OWL-S Editor”. Third International Semantic Web Conference - ISWC 2004, Hiroshima, Japan. November, 2004.
13. N. F. Noy et all. “Creating Semantic Web Contents with Protege-2000”. IEEE Intelligent Systems 16(2):60-71, 2001.
14. J. Arkko and B. Aboba, (editors). “Network Discovery and Selection Problem”. Internet-Draft. IETF EAP WG. July 17, 2004
15. R. van Eijk, J. Brok, J. van Bemmel, B. Busropan, “Access Network Selection in a 4G Environment and the Roles of Terminal and Service Platform”, WWRF #10, 27-28 October 2003, New York.
16. H. Knublauch, R. W. Fergerson, N. F. Noy, M. A. Musen.“The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications”. ISWC 2004, Hiroshima, Japan. November, 2004.
17. M. O'Droma and I. Ganchev. “New Access Network Techno-Business Model for 4GWW”. In Proceedings of 4th ANWIRE International Workshop on Wireless Internet and Reconfigurability, 7 pages, 14 May 2004, Athens, Greece.
18. A. Kassler et al. “Enabling Mobile Heterogeneous Networking Environments With End-to-End User Perceived QoS - The BRAIN vision and the MIND approach”. European Wireless Conference 2002. Florency, Italy. February, 2002.

Replic8: Location-Aware Data Replication for High Availability in Ubiquitous Environments

Evangelos Kotsovinos and Douglas McIlwraith

University of Cambridge Computer Laboratory
evangelos.kotsovinos@cl.cam.ac.uk

Abstract. File replication for uninterrupted availability is affected by the localised nature of network failures, particularly in ubiquitous, mobile environments; nearby nodes often get disconnected together, as a result of switching equipment faults, or of local wireless network unavailability – for instance, failure of a base station, or loss of network connectivity when a train enters a tunnel.

In this paper we propose replic8, a substrate for location-aware file replication, mitigating the effect of localised network failures by storing replicas at network locations selected for being far away. We demonstrate that, compared to storage of replicas at random network locations, replic8 achieves high data availability, and requires lower numbers of replicas to maintain that.

1 Introduction

Mobile and ubiquitous computing proposals envisage a world in which heterogeneous devices are interconnected to share data with ease and reliability. Connectivity of mobile devices such as mobile phones, PDAs and laptop computers is a commodity, due to the nature of the medium which is used.

The *transience* of such devices raises important research challenges in supporting uninterrupted presence of resources. Mobile devices often encounter network unavailability; this can be due to location, as when a user is passing through a valley where reception of her mobile phone network is weak, or when a train is going through a tunnel. It can also be due to failures of network software or hardware, such as network switches and transparent proxies, or base stations – for instance, as a result of poor weather conditions. *File replication* has been used for several years to provide significant benefits in terms of uninterrupted availability of files [1]. More recently its importance for mobile and ubiquitous environments has been realised [2].

In mobile networks we observe that such failures are usually *localised*; if a mobile phone cannot receive signal at a particular location it is highly unlikely that another one at the same location – connected to the same network – will, and if a PDA is not able to connect to a hotel lounge's wireless LAN it is improbable that other PDAs and laptops in the same lounge can. The probability of a mobile

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 32–41, 2005.

device being available, given that a second mobile device is unavailable, is higher if the two devices are a certain distance apart.

To address the localised nature of network failures, we propose *replic8*, a system for enhancing file availability in such environments. We use a location-based metric for determining which servers are "suitable" for hosting the replicas, based on the probability that they will fail together. Then, we replicate the file on nodes where that probability is low. In this paper we use physical distance between nodes as the aforementioned metric, there is no architectural restriction, however, to prevent other metrics from being used.

This paper presents the design and implementation of our system, and initial evaluation results obtained in an internal deployment. We present the system architecture, focusing on the components of the distributed architecture and the operations supported, in Section 2. The prototype deployment and evaluation setup is described in Section 3, along with our initial experimental results. Section 4 positions our system among related research work. Finally, Section 5 concludes and outlines our future plans for the deployment of our system in a mobile environment.

2 System Architecture

Servers participating in the replic8 substrate are termed *nodes*. Various types store replicas and run management software for communication with other nodes and distributed components. *Users* contact nodes for adding or retrieving files.

The node at which a data item is added to the system is called the *owner node* for that item, and the node that a user contacts for retrieving a file is called the *client node*. The owner node determines which nodes are suitable for hosting the replicas – based on their location – and passes replicas to other nodes. The nodes that store replicas of a data item are termed the *replica nodes* for that item. The set of nodes including both the owner node and replica node(s) for a given data item shall be known as the *data nodes* for that item.

A number of research challenges need to be addressed: as users need to be able to add, remove, modify, and retrieve files by contacting any of the replic8 nodes, all nodes need access to information about the availability and *position of other nodes*, as every time a file is inserted location-based decisions need to be made on where its replicas are to be stored. At the same time, nodes need to be able to obtain information about the *location of replicas*, to allow discovery of data items. *Consistency management* issues are also raised, as concurrent modifications of the same data item may be attempted from different nodes.

2.1 Overview

To distribute node position and replica location information we employ special groups of nodes – each one comprising nodes selected to be far from each other, for higher fault-tolerance; then we run each of the required services on a group of nodes in a replicated fashion. This is not dissimilar to super-peer nodes [3]

on peer-to-peer networks, but where designers of such networks may strive to assign super-peer 'status' to nodes most capable of handling the load, replic8 seeks nodes which are least the likely to fail together. We use three such groups of nodes:

- **Directory service nodes** participate in file discovery by storing and providing information about the files that exist in the system.
- **Group service nodes** identify data nodes for each data item in the system and keep track of the location of replicas for each file.
- **Position service nodes** store position information about the nodes in the system and assess the suitability of nodes to hold particular replicas.

Updates sent to any of a group of service nodes are immediately *multicast* to other members of the group. This ensures that all members of the group maintain the same information. If acknowledgement is received from all members then the update is complete, else the original recipient of the update enters a phase of *continual transmission* to the unavailable node, until acknowledgement. This ensures the missing node receives the update soon after it becomes available again. Specific group node operations are discussed in Section 2.2.

In our prototype implementation service nodes are selected randomly – as long as they are further apart than a specified minimum distance, in a future implementation, however, we envisage using a distributed election algorithm appropriate for mobile and ubiquitous environments [4, 5].

Service nodes inform others of the same group when they are disconnecting. Each node also periodically multicasts a heartbeat message, to make sure others can find out if it disappears abnormally, without prior notification.

Since initial results are obtained in a trusted environment, we ignore the possibility of malicious intent. This will need to be re-addressed as we deploy on more large scale distributed networks, perhaps using our experience on trust and reputation management systems [6] and honesty detection mechanisms [7].

2.2 Operations

In this section we describe the internal actions that are taken every time a user calls one of replic8's interfaces to add, retrieve, modify, or remove a file. We also discuss two operations that take place periodically in the background, namely replica relocation and consistency management.

File Addition. Users contact any replic8 node, which becomes the owner node for the file to be added. Additions to the system require generation of a file identifier, and replication on suitably remote nodes. The former is carried out with the help of the directory service, which generates a unique file ID, stores its mapping to the file's name, and returns the ID to the user – operation 2 in Figure 1. The latter is completed by communication with the position service, which recommends a list of servers according to physical proximity – operation 3. Nodes periodically submit advertisements to the position service specifying their current status and location – operation 1.

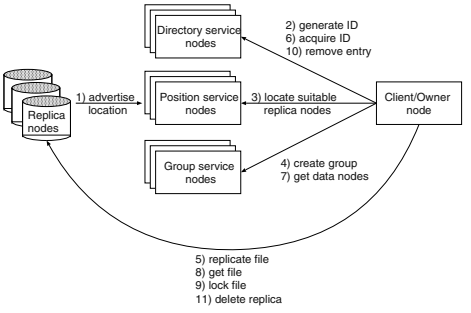


Fig. 1. Overview of the replic8 architecture

The owner node decides, based on the position service's recommendations, on the set of replica nodes to be used and replicates the file there – operation 5. Successful completion triggers communication with the group service to commit the relationship between the file and its data nodes – operation 4.

If a proposed replica node is unable to receive the replica then the owner node enters a state of retransmit, trying to contact the unavailable node until it responds. If after a period of time no response is received the owner node attempts to find another suitable set of replica nodes. If there is no other suitable set of replica nodes then the data is replicated on the available nodes only and inconsistency is resolved at a later stage.

File Discovery and Retrieval. Users request the retrieval of a file providing its human-readable filename. The client node first acquires the unique file ID for the requested file from the directory service – operation 6. On receipt of the ID, the client node contacts the group service to request the data nodes for this file – operation 7, then tries to retrieve the file from each one of the data nodes until the owner node or the first available replica node returns the file – operation 8.

File Modification. Firstly, the client node that has been contacted retrieves the identifier for the file to be modified, which is carried out by communication with the directory service – operation 6. Then, support from the group service is required to provide the list of data nodes for the file – operation 7.

The client node then contacts all data nodes that store copies of the file. If all data nodes are available, the client requests an exclusive *write lock* on the given file on all those nodes – operation 9. If only a subset of the data nodes for that file can be contacted, updates go ahead as above, but only at the nodes contained in the subset. Resolution of the arising inconsistency is described below.

To prevent deadlock we use *soft-state* file locks; each lock is associated with an expiration time and cannot be retained by a node indefinitely. Once confirmed by all involved, updated copies are written to the data nodes and the locks released.

File Removal. Removing a file from the system involves initially deleting the entry for the file in the directory service, which is carried out by the client node

in communication with the directory service – operation 10. This ensures that any further searches for the file will not return any results.

Any replicas of the file stored in remote nodes then have to be deleted by submitting replica deletion requests – operation 11. We follow a "lazy" approach towards replica removal; nodes that receive such requests mark files as "to be deleted" but do not actively delete them unless the allocated space for storing replicas is filling up. This allows for file recovery and better file removal performance, without compromising consistency.

Replica Relocation Management. Nodes advise the position service about changes in their position regularly – operation 1. As we are focusing on ubiquitous and mobile environments, it is necessary that our architecture is able to adapt to change of location and availability of mobile nodes. When a replica node gets closer to the owner node, replic8 needs to locate another, more suitable replica node, copy the file there, and remove the now nearby, old replica node from the data nodes group for the file.

This is carried out periodically by the owner node for the file, which requests recommendations from position service and determines whether there is a significantly better set of replica nodes that may be used. The period at which this operation is undertaken, as well as the threshold above which a change of replica nodes for a file is required, present two important trade-offs; higher frequency of this operation and low restructuring threshold lead to higher data availability – as the replica nodes are most of the time suitable – but incur higher network traffic and computational load for the replic8 nodes.

Consistency Management. As we are employing an optimistic replication scheme, it is necessary that replic8 performs internal management operations periodically to ensure consistency of replicas. Detection and resolution of inconsistencies for a file is ultimately a responsibility of the owner node for the file, and is achieved using Version Vectors [8]. Version Vectors are well understood and it is known that they cannot resolve consistency when copies of a file of the same version have been modified in isolation from the others. Any attempt at resolution will result in one or more modifications to the file being lost. If this situation occurs in replic8 manual intervention is required, and we provide methods to signal this to the file owner affected.

3 Evaluation

In this section, we present our initial results in assessing replic8's performance in terms of achieving higher data availability than a random replication strategy, where files are copied to arbitrary nodes, in an environment where localised failure is common. All experiments were carried out on an Intel Pentium III Mobile CPU, clocked at 866MHz, with 128Mb of main memory. Java 1.4.1-02 was used on a Windows 2000 platform.

To measure replic8's effectiveness, we launched a system comprising a thousand replic8 nodes, and associated a random position in a 2D Cartesian coordi-

nate space with each. The 20x20-sized coordinate space used was split in four hundred 1x1-sized sub-spaces called *quadrants*, each *contains* the nodes whose position coordinates fall inside that quadrant. In our experiments we do not consider varying network conditions, such as route changes or packet loss.

We also present results indicitave of replic8's *network overhead* for file maintainance compared to that of a system which is not location aware. Both systems implement functionality to maintain file consistency, enable file addition and file retrieval, but unlike replic8, the system incapable of location aware operation did not implement services to respond to file movement by relocating replicas.

The simulation started with *file addition*, where files were inserted in the system and replicated accordingly. After that, the simulation entered an iteration comprising *failure generation*, measurement of the number of *unavailable files* in the system, and *node relocation*, as explained below.

File Addition. Suitable replica nodes for the files inserted were discovered, and the files were replicated on these nodes. File addition only happened once, and the files remained in the system throughout its operation after that point. All quadrants were initially set to active.

We added a thousand files, or one on each replic8 node. The *minimum distance* that replicas should be held at, based on which the position service recommends suitable replica holders to the owner nodes at file addition, had been set to seven.

Failure Generation. The first step of the simulation iteration involved injecting localised network failures. We did so by making neighbouring quadrants fail together according to a simple localised failure generation algorithm, and compared the number of available files in the system using random replication to that achieved using replic8. A file is considered *available* if it can be found on at least one currently connected replic8 node.

The failure generation algorithm we used works as follows; each time the failure generation algorithm is called, a single quadrant is picked randomly from the coordinate space and set as disabled. This ensures that all nodes it contains are immediately marked as disconnected. At the same time, neighbouring quadrants may be disabled too, making up a disconnected area of a radius equal to the *localisation coefficient*. In our experiments, the coefficient has been varied from one to twenty.

Node Relocation. Nodes move around the coordinate space in random directions by a fixed *walk distance* and register and unregister with quadrants accordingly. In the tests we carried out, to simulate a highly mobile environment nodes moved around by a unit – ensuring most change quadrant every iteration.

After each relocation step, each owner node ensures that all nodes previously used as replica holders for its file have not got prohibitively close, then a new failure generation step is invoked. This cycle is continued until a number of iterations is reached.

3.1 Results

File Availability. In the aforementioned setting, we have executed the simulation iteration twenty times for each combination of values for number of replicas and localisation coefficient. We have calculated the average number of available files out of the one thousand files initially added to the system as the size of the disconnected area increases. Figure 2 compares the availability achieved using replic8 to that using a scheme where replica nodes were selected at random.

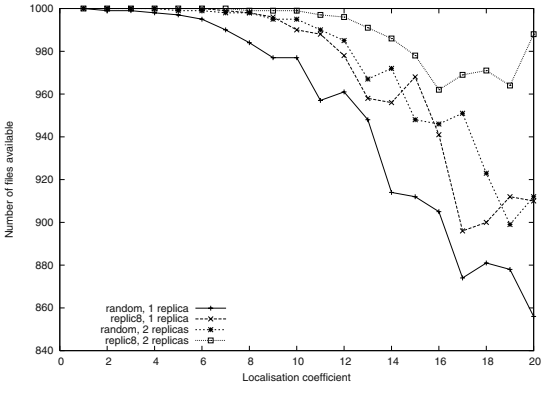


Fig. 2. File availability as the size of a disconnected area grows

We have plotted, along the x-axis, the *localisation coefficient*, representing the maximum size of the disconnected area around a randomly selected quadrant. The y-axis displays the *availability* of files – the number of files accessible through at least one of their data nodes. Availability has been measured in two cases; in the first case, only one replica of each file was being held – so two copies of the file existed in the system, including the original one. In the second case, each file was copied twice on replica nodes.

The graph shows that using replic8, for localisation coefficient 15, only 32 files are unavailable, compared to 88 using random replication. Even by maintaining an extra copy of the file, random replication does not surpass replic8, leading to 52 files being unavailable; replic8 achieves availability of all files but 22 using two replicas per file. In general, *replic8 using one replica performs almost as well as random replication using two replicas* – in some cases even better. This is an important result given that storage space can be restricted on mobile devices.

The graph shows that file availability decreases as the size of the disconnected area grows for both file replication schemes – replic8 and random replication. This is inevitable, as increasing the size of the disconnected area causes more nodes to be disconnected. Furthermore, the frequently changing direction of the plot is due to the randomness present in the simulation; since nodes move

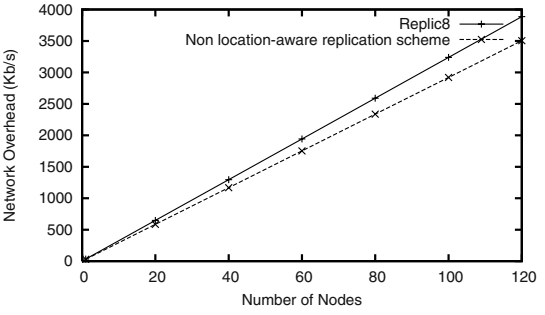


Fig. 3. Network overhead as the number of nodes increase

arbitrarily and failure is random, convergence to an average availability for a given localisation factor is not guaranteed. We believe it is exaggerated by the comparatively large size of the quadrants to the entire area.

Overall, replic8 provides significant file availability benefits in environments where node disconnection is highly localised, such as – potentially – in ubiquitous, mobile, and wireless networks.

Network Overhead. Figure 3 compares *average* network overhead with replic8, and with a non location-aware replication scheme which does not respond to the movement of nodes. The overhead illustrated is due to file maintainance when the system is populated with a varying number of nodes. Each node owns a 500k file and is responsible for its consistency with replicas as well as, in replic8's case, file movement between nodes when node positions become unsuitable.

Both plots exhibit linear behaviour as the number of nodes increase. This is due to increased load on the network from consistency checking and, in replic8, movement of files. These trends are linear since both aforementioned operations occur with a fixed period.

The regularity of node failure also effects this load as, upon service failure, service nodes enter a state of permanent retransmit until all members of a service group are available. Node failure frequency affects data holders for a file also, since the more often data holders for a file are unavailable, the more likely updates will occur optimisitically, at a subset of the data nodes, requiring expensive additional clean-up operations to regain consistency. These figures were based on individual nodes failing with an *average* probability of 0.001.

With replic8, network overhead grows more rapidly as the number of nodes increase. This is due to additional overhead in maintaining an extra service, namely regular publication of position from nodes, plus the load exhibited from file relocation as nodes move around the system. For 120 nodes we calculate an increased network overhead of around 11%

While Figure 3 demonstrates load for file maintainance, it does not illustrate the cost of individual operations. We calculate that replic8 creates a load of

1099Kb on the network introducing a single 500k file to the system, in comparison to a load of 1008Kb for a *non-location-aware* system – an increased load of around 9%. File removal and retrieval in replic8 costs the same whether or not the solution is location aware.

Replic8 provides, with minor network overhead, significant availability improvements over random replication – allowing the use of fewer replicas. This lets us trade storage overhead for a small network overhead in environments where storage is a commodity.

4 Research Context

Distributed storage systems have used replication to improve data availability in unreliable network conditions. Ficus [9] adopts optimistic replication and periodic consistency checks – like replic8 – for reliable access to files on transient devices, and delegates the decision as to how many file replicas are to exist and where, to the clients. Bayou [10] targets heterogeneous environments and provides high availability via a read-any/write-any scheme of data access. A different approach for data replication, based on an economic model for managing the allocation of storage and queries to servers, is employed in Mariposa [11].

All above systems have presented important solutions to problems such as disconnected operation, replica discovery, and conflict resolution. Our work specifically targets the issue of improving file availability in *localised network failure* conditions. replic8 addresses the problem by basing the decisions on where replicas are to be stored on location, or other application-specific metrics.

Fluid Replication [12] employs location-based replication in static environments, ensuring that replicas are stored near the original for reduced latency. Replic8 differs by targeting environments where localised replication would be disastrous; a single failure may either bring down an entire mobile service cluster or a client and its local replicas.

Replic8's service nodes are inspired by super-peer nodes found in peer to peer systems [3]. The simple replication strategy followed is sufficient to validate our approach, as shown by the evaluation results, and avoids requiring the storage and processing of excessive amounts of data on mobile devices.

5 Conclusions and Future Work

In this paper we have presented replic8, a system for intelligent file replication to achieve high data availability in environments where device disconnections are non-arbitrary. We have shown replic8 to operate more effectively than random replication in an environment simulating highly localised failures of a mobile network – while requiring fewer replicas. We have also demonstrated that this can be achieved with small network overhead above a *non-location-aware* system.

We plan to deploy replic8 on our COMS[1] and XenoServers [13] and obtain more accurate results and experience on the performance and fault-tolerance of our system. Furthermore, we plan to augment distributed storage systems such as CODA [14] with replic8 to achieve higher file availability with fewer replicas under conditions of non-arbitrary node disconnection.

References

1. Walker, B., Popek, G., English, R., Kline, C., Thiel, G.: The LOCUS Distributed Operating System. In: Proc. of the 9th ACM Symposium on Operating Systems Principles. (1983)
2. Ratner, D., Reiher, P., Popek, G.J., Kuenning, G.H.: Replication Requirements in Mobile Environments. Mobile Network Applications **6** (2001) 525–533
3. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. In: Proc. of the 19th Intl. Conf. on Data Engineering. (2003)
4. Malpani, N., Welch, J.L., Vaidya, N.: Leader Election Algorithms for Mobile Ad Hoc Networks. In: Proc. of the 4th Intl. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. (2000)
5. Vasudevan, S., DeCleene, B., Immerman, N., Kurose, J., Towsley, D.: Leader Election Algorithms for Wireless Ad Hoc Networks. In: Proc. of the 3rd DARPA Inf. Survivability Conf. and Exposition (DISCEX- III). (2003)
6. Dragovic, B., Hand, S., Harris, T., Kotsovinos, E., Twigg, A.: Managing Trust and Reputation in the XenoServer Open Platform. In: Proc. of the 1st International Conference on Trust Management (iTrust 2003). (2003)
7. Fernandes, A., Kotsovinos, E., Ostring, S., Dragovic, B.: Pinocchio: Incentives for Honest Participation in Distributed Trust Management. In: Proc. of the 2nd International Conference on Trust Management (iTrust 2004). (2004)
8. Parker Jr., D., Popek, G., Rudisin, G., Stoughton, A., Walker, B., Walton, E., Chow, J., Edwards, D., Kiser, S., Kline, C.: Detection of Mutual Inconsistency in Distributed Systems. IEEE Transactions on Software Engineering SE-9, (1983)
9. Guy, R.G., Heidemann, J.S., Page, Jr., T.W.: The Ficus Replicated File System. SIGOPS Oper. Syst. Rev. **26** (1992) 26
10. Terry, D.B., Theimer, M.M., Petersen, K., Demers, A.J., Spreitzer, M.J., Hauser, C.: Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In: Proc. of the 15th Symp. on Oper. Sys. Principles (SOSP-15). (1995)
11. Sidell, J., Aoki, P.M., Sah, A., Staelin, C., Stonebraker, M., Yu, A.: Data replication in mariposa. In: Proc. of the 12th International Conference on Data Engineering, IEEE Computer Society (1996) 485–494
12. Noble, B., Fleis, B., Kim, M., Zajkowski, J.: Fluid replication. In: Proc. of Netstore '99, the Network Storage Symposium. (1999)
13. Hand, S., Harris, T.L., Kotsovinos, E., Pratt, I.: Controlling the XenoServer Open Platform. In: Proc. of the 6th International Conference on Open Architectures and Network Programming (OPENARCH). (2003)
14. Kistler, J.J., Satyanarayanan, M.: Disconnected operation in the coda file system. In: Proc. of the 13th ACM Symp. on Oper. Sys. Principles (SOSP-13). Volume 25., Asilomar Conference Center, Pacific Grove, U.S., ACM Press (1991) 213–225

[1] Cambridge Open Mobile System, `http://www.cl.cam.ac.uk/coms/`

Refined PFTK-Model of TCP Reno Throughput in the Presence of Correlated Losses

Roman Dunaytsev, Yevgeni Koucheryavy, and Jarmo Harju

Institute of Communications Engineering,
Tampere University of Technology,
P.O. Box 553, FIN-33101, Tampere, Finland
{dunaytse, yk, harju}@cs.tut.fi

Abstract. This paper presents a simple and accurate analytical model of TCP Reno throughput as a function of loss rate, average round trip time and receiver window size based on PFTK-model. The presented model refines previous work by careful examination of fast retransmit/fast recovery dynamics in the presence of correlated losses and taking into consideration slow start phase after timeout. The accuracy of the proposed model is validated against simulation results and compared with those of PFTK-model. Simulation results show that our model gives a more accurate estimation of TCP Reno throughput in the presence of correlated losses than PFTK-model.

1 Introduction

Transmission Control Protocol (TCP) is the de facto standard protocol for the reliable data delivery in the Internet. Recent measurements show that from 60% to 90% of today's Internet traffic is carried by TCP [1]. Due to this fact, TCP performance modeling has received a lot of attention during the last decade [2].

One of the most known and wide referenced analytical models of TCP throughput of a bulk transfer is the model proposed by J. Padhye et al. in [3], also known as PFTK-model. This model describes steady-state throughput of a long-lived TCP Reno bulk transfer as a function of loss rate, average round trip time (RTT) and receiver window size. It assumes a correlated (bursty) loss model that is better suited for FIFO Drop Tail queues currently prevalent in the Internet.

Unfortunately, this model does not capture slow start phase after timeout and uses simplified representation of fast retransmit/fast recovery dynamics in the presence of correlated losses as having negligible effect on TCP Reno throughput. As it will be shown later, such simplifications can lead to overestimation of TCP Reno throughput. Since new analytical TCP models are often compared with PFTK-model (e.g., [4], [5], [6]) and use its resultant formula (e.g., [7], [8]), such inaccuracy in throughput estimation can lead to inaccurate results or incorrect conclusions.

In this paper, we propose a simple and more accurate steady-state TCP Reno throughput prediction model. This is achieved by careful examination of fast retransmit/fast recovery dynamics in the presence of correlated losses and taking into consideration slow start phase after timeout.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 42–53, 2005.

The reminder of the paper is organized as follows. Section 2 describes assumptions we made while constructing our model. Section 3 presents a detailed analysis of the proposed model. Section 4 describes model validation experiments, presents an analysis of the accuracy of our model and the one proposed in [3]. Finally, Section 5 concludes the paper.

2 Assumptions

The refined model we develop in this paper has exactly the same assumptions about endpoints and network as the model presented in [3]. We assume that the sender uses TCP Reno congestion control algorithm based on [9] and always has data to send. Since we are focusing on TCP performance, we do not consider sender or receiver delays and limitations due to scheduling or buffering. Therefore, we assume that the sender sends full-sized segments whenever the congestion window (*cwnd*) allows, while the receiver window (*rwnd*) is assumed to be always constant. We model TCP behavior in terms of "rounds" as done in [3], where a round starts when the sender begins the transmission of a window of segments and ends when the sender receives an acknowledgement (ACK) for one or more of these segments. It is assumed that the receiver uses delayed acknowledgement algorithm according to [10]. When modeling data transfer, we assume that segment loss happens only in the direction from the sender to the receiver. Moreover, we assume that a segment is lost in a round independently of any segments lost in other rounds, but at the same time segment losses are correlated within a round (i.e., if a segment is lost, all the remaining segments in that round are also lost). Such bursty loss model is a simplified representation of IP-datagram loss process in routers using FIFO Drop Tail queuing discipline. We assume that the time needed to send a window of segments is smaller than the duration of a round; it is also assumed that probability of segment loss and the duration of a round are independent of the window size. This can only be true for flows that are not fully utilizing the path bandwidth (i.e., in case of high level of statistical multiplexing).

3 The Model

According to [9], segment loss can be detected in one of two ways: either by the reception at the sender of "triple-duplicate" ACK or via retransmission timeout expiration. Similarly to [3], let us denote the first event as a TD (triple-duplicate) loss indication, and the second as a TO (timeout) loss indication. As in [3], we develop our model in several steps: when the loss indications are exclusively TD (Section 3.1); when the loss indications are both TD and TO (Section 3.2); and when the window size is limited by the receiver window (Section 3.3).

3.1 TD Loss Indications

In this section, we assume that all loss indications are exclusively TD and that the window size is not limited by the receiver window. In this case, according to [3], the

long-term behavior of TCP Reno flow may be modeled as a cyclic process, where a cycle (denoted in [3] as a TD Period, TDP) is a period between two TD loss indications. For the i-th cycle $(i = 1,2,...)$ let Y_i be the number of segments sent during the cycle, A_i be the duration of the cycle and W_i be the window size at the end of the cycle. Considering $\{W_i\}_i$ to be a Markov regenerative process with renewal reward process $\{Y_i\}_i$, we can define the long-term steady-state TCP throughput B as

$$B = \frac{E[Y]}{E[A]}. \tag{1}$$

Fig. 1 shows the evolution of congestion window size during the i-th cycle according to [3].

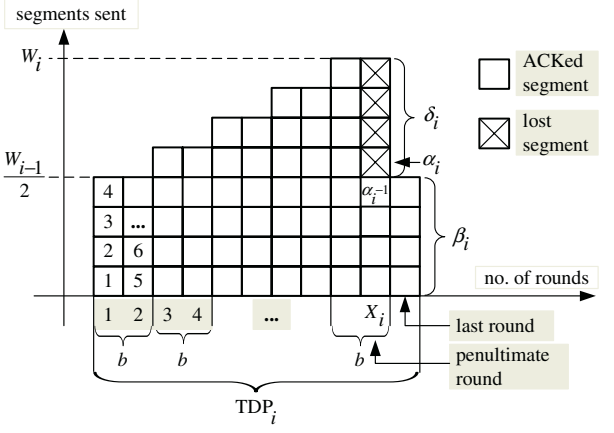


Fig. 1. Segments sent during the i-th cycle (TD Period) according to [3]

A cycle starts immediately after a TD loss indication, hence the current *cwnd* (expressed in segments) is set to $W_{i-1}/2$. The receiver sends one ACK for every b-th segment that it receives (according to [10], $b = 2$), so *cwnd* increases linearly with a slope of $1/b$ segments per round until the first segment loss occurs. Let us denote by α_i the first segment loss in the i-th cycle and by X_i the round where this loss occurs (see Fig. 1). According to the sliding window algorithm, after the segment α_i, $(W_i - 1)$ more segments are sent before a TD loss indication occurs and the current cycle ends.

Let us consider the evolution of congestion window size in the i-th cycle after the first TD loss indication. Taking into account the assumption about correlated losses within a round (i.e., if a segment is lost, so are all following segments till the end of the round), all segments following α_i in the round X_i (denoted in Fig. 1 as the pe-

nultimate round) are lost as well. Let us define δ_i to be the number of segments lost in the round X_i and β_i to be the number of segments sent in the next (and the last) round (X_i+1) of the i-th cycle (see Fig. 1). Similarly to [3], we assume that random variables β_i and δ_i are uniformly distributed from zero to (W_i-1) and from one to W_i correspondingly. Thus, taking into account that $\beta_i = W_i - \delta_i$ we have

$$E[\beta] = \frac{E[W]-1}{2}, \quad E[\delta] = \frac{E[W]+1}{2}. \tag{2}$$

After a TD loss indication the sender enters the fast retransmit/fast recovery phase and performs a retransmission of the lost segment. The slow start threshold ($ssthresh$) and the current value of $cwnd$ are updated according to [9] as

$$ssthresh = \max(FlightSize/2, 2), \quad W' = ssthresh + N_{DupACK}, \tag{3}$$

where $FlightSize$ is the number of segments that has been sent, but not yet acknowledged; W' is the value of $cwnd$ during fast recovery phase; N_{DupACK} is the number of received duplicate ACKs.

Since $E[N_{DupACK}] = E[\beta]$, $E[FlightSize] = E[W]$ and using (2), we can determine $E[W']$ as

$$E[W'] = E[ssthresh] + E[N_{DupACK}] = \frac{E[W]}{2} + \frac{E[W]-1}{2} = E[W] - \frac{1}{2}. \tag{4}$$

As $E[W'] < E[FlightSize]$, it is expected that the sender will not send new segments in the fast recovery phase. After the successful retransmission of the segment α_i the sender will receive new ACK, indicating that the receiver is waiting for the segment (α_i+1). As a result of receiving this new ACK, the phase of fast retransmit/fast recovery ends and according to [9] the new value of $cwnd$ is set as $W = ssthresh$, where $ssthresh$ is from (3). Since $FlightSize$ is still larger than the new value of $cwnd$, the sender cannot transmit new segments, therefore this ACK will be the single. As the sender will not be able to invoke the fast retransmit/fast recovery algorithms again, then it will wait for the expiration of retransmission timeout (RTO), which was set after the successful retransmission of the segment α_i (in accordance with [11], step 5.3). After the RTO expiration, the values of $cwnd$ and $ssthresh$ are set as $W = 1$ and $ssthresh = \max(FlightSize/2, 2)$, and the slow start phase begins.

Thus, in the presence of correlated losses and when the first loss is detected via a TD loss indication, the following sequence of steps is expected:

- initialization of the fast retransmit and fast recovery algorithms, retransmission of the first lost segment;

- awaiting for the RTO expiration, which was set after the successful retransmission of the first lost segment;
- initialization of the slow start algorithm.

Our observation is well agreed with the results from [12], showing that TCP Reno has performance problems when multiple segments are dropped from one window of segments and that these problems result from the need to wait for the RTO expiration before reinitiating data flow. Moreover, empirical measurements from [3] show that the significant part of loss indications (in average 71%) is due to timeouts, rather than TD loss indications.

In order to include the fast retransmit/fast recovery phase and the slow start phase, we define a cycle to be a period between two TO loss indications (besides periods between two consecutive timeouts). Therefore, a cycle consists of the slow start phase, congestion avoidance phase, fast retransmit/fast recovery phase and one timeout. An example of the evolution of congestion window size is shown in Fig. 2, where the congestion avoidance phase (TD Period in [3]) is supplemented with the slow start phase at the beginning and the fast retransmit/fast recovery phase with one timeout at the end.

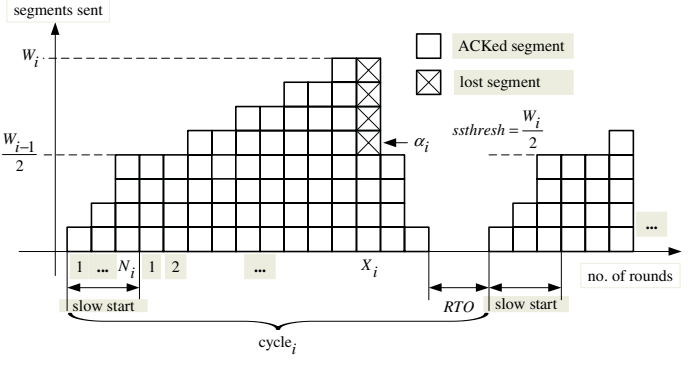


Fig. 2. Evolution of congestion window size during the i-th cycle, supplemented with the slow start phase at the beginning and the fast retransmit/fast recovery phase with one timeout at the end of the congestion avoidance phase

Observe that $Y_i = \alpha_i + W_i$, thus we have

$$E[Y] = E[\alpha] + E[W]. \tag{5}$$

The expected number of segments sent in a cycle up to and including the first lost segment is given in [3] as

$$E[\alpha] = \sum_{k=1}^{\infty} (1-p)^{k-1} \cdot p \cdot k = \frac{1}{p}, \tag{6}$$

where p is the probability that a segment is lost, given that it is either the first segment in its round or the preceding segment in its round is not lost.

As in [3], let r_{ij} be the duration of the j-th round of i-th cycle ($i, j = 1, 2, \ldots$). If we assume r_{ij} to be random variables independent of $cwnd$, then we have

$$E[A] = E[r] \cdot \left(E[N] + E[X] + 2 + \frac{E[RTO]}{E[r]} \right), \tag{7}$$

where $E[r] = \overline{RTT}$; $E[RTO] = \overline{RTO}$; $E[N]$ is the expected number of slow start rounds.

In order to derive $E[X]$ and $E[W]$, let us consider the evolution of $cwnd$ as a function of number of rounds. Similarly to [3], we assume that $W_{i-1}/2$ and X_i/b are integers. Therefore, we have

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{b} - 1, \quad i = 1, 2, \ldots \tag{8}$$

Then the number of segments sent during the congestion avoidance (CA) phase of the i-th cycle can be defined as

$$Y_i^{CA} = \sum_{k=0}^{\frac{X_i}{b}-1} \left(\frac{W_{i-1}}{2} + k \right) \cdot b + \beta_i = \frac{X_i \cdot W_{i-1}}{2} + \frac{X_i}{2} \cdot \left(\frac{X_i}{b} - 1 \right) + \beta_i . \tag{9}$$

Combining (8) and (9), we obtain

$$Y_i^{CA} = \frac{X_i}{2} \cdot \left(W_i + \frac{W_{i-1}}{2} \right) + \beta_i . \tag{10}$$

Similarly to [3], we assume $\{X_i\}_i$ and $\{W_i\}_i$ to be mutually independent sequences of i.i.d. random variables. After the transformation of (8) we get

$$E[X] = b \cdot \left(\frac{E[W]}{2} + 1 \right). \tag{11}$$

Hence

$$E\left[Y^{CA}\right] = \frac{E[X]}{2} \cdot \frac{3 \cdot E[W]}{2} + E[\beta] = \frac{3 \cdot b \cdot E[W]}{4} \cdot \left(\frac{E[W]}{2} + 1 \right) + \frac{E[W] - 1}{2} . \tag{12}$$

According to [7], the number of segments sent during the slow start (SS) phase can be closely approximated by a geometric series. At the same time it is known from [9], that the receiver sends an immediate duplicate ACK when out-of-order segment arrives (i.e., $b = 1$). Then we can approximate

$$Y^{SS} = 1 + 2 + 2^2 + ... + 2^{N-1} = \sum_{k=1}^{N} 2^{k-1} = 2^N - 1. \quad (13)$$

The required number of slow start rounds to send Y^{SS} segments can be expressed as

$$N = \log_2 \left(Y^{SS} + 1 \right). \quad (14)$$

Taking into account, that in the slow start phase of the i-th cycle $cwnd$ growths exponentially from one to $ssthresh = W_{i-1} / 2$, from (13) we have

$$\frac{E[W]}{2} = 2^{E[N]-1}. \quad (15)$$

Combining (13), (14) and (15), we obtain

$$E\left[Y^{SS} \right] = E[W] - 1. \quad (16)$$

By substituting (16) in (14) and taking into consideration (3), we get the expected number of slow start rounds as

$$E[N] = \max \left(\log_2 E[W], 2 \right). \quad (17)$$

Based on (5), (12) and (16) and taking into account the retransmitted segment in the fast retransmit phase, the following system of equations can be defined as

$$\begin{cases} E[Y] = \dfrac{1}{p} + E[W] \\ E[Y] = E[W] - 1 + \dfrac{3 \cdot b \cdot E[W]}{4} \cdot \left(\dfrac{E[W]}{2} + 1 \right) + \dfrac{E[W] - 1}{2} + 1 \end{cases} \quad (18)$$

Solving this system of equations for $E[W]$, we get

$$E[W] = -\left(\frac{2 + 3 \cdot b}{3 \cdot b} \right) + \sqrt{\frac{8 + 4 \cdot p}{3 \cdot b \cdot p} + \left(\frac{2 + 3 \cdot b}{3 \cdot b} \right)^2}. \quad (19)$$

In order to show that the slow start phase will enter in the congestion avoidance phase before the first segment loss occurs, we have to prove that $E\left[Y^{SS} \right] < E[\alpha]$ (i.e., $E[W] - 1 < \frac{1}{p}$). Solving this inequality, we get $\frac{3 \cdot b}{p} + 9 \cdot b \cdot p + 12 \cdot b > 4$. The last inequality holds since $p > 0$ and $b \geq 1$.

By substituting (11) and (17) in (7), we have

$$E[A] = \overline{RTT} \cdot \left(\max \left(\log_2 E[W], 2 \right) + b \cdot \left(\frac{E[W]}{2} + 1 \right) + 2 + \frac{\overline{RTO}}{\overline{RTT}} \right). \quad (20)$$

Combining (1), (5), (6), (19) and (20), we obtain

$$B = \frac{\frac{1}{p} + E[W]}{\overline{RTT} \cdot \left(\max\left(\log_2 E[W], 2\right) + b \cdot \left(\frac{E[W]}{2} + 1 \right) + 2 + \frac{\overline{RTO}}{\overline{RTT}} \right)}, \tag{21}$$

where $E[W]$ is given in (19).

3.2 TD and TO Loss Indications

A TO loss indication happens when segments (or ACKs) are lost and less than three duplicate ACKs are received. Note that in this case there will be no fast retransmit/fast recovery phase in a cycle. Similarly to [3], we define W_{ij} to be the window size at the end of the j-th cycle $(i,\ j = 1,2,...)$, A_{ij} to be the duration of the j-th cycle, Z_i^{TO} to be the duration of a sequence of timeouts, Z_i^{TD} to be the duration of time interval between two consecutive timeout sequences, $S_i = Z_i^{TD} + Z_i^{TO}$. The number of transmitted segments during the last cycle and the duration of the last cycle can be approximated as $\left(E[Y]-1\right)$ and $\left(E[A]-\overline{RTT}\right)$ (where $E[Y]$ is from (5) and $E[A]$ is from (20)).

From [3] we can define long-term steady-state TCP throughput B as

$$B = \frac{E[n]\cdot E[Y] + E[R] - 1}{E[n]\cdot E[A] + E\left[Z^{TO}\right] - \overline{RTT}} = \frac{E[Y] + Q\cdot\left(E[R]-1\right)}{E[A] + Q\cdot\left(E\left[Z^{TO}\right] - \overline{RTT}\right)}, \tag{22}$$

where $E[R]$ is the expected number of segments sent during timeout sequence; $E\left[Z^{TO}\right]$ is the expected duration of timeout sequence; $Q = 1/E[n]$ is the probability that a loss indication ending a cycle is a TO.

The probability that a loss indication is a TO under the current congestion window size w is given in [3] as

$$\hat{Q}(w) = \min\left(1, \frac{\left(1-(1-p)^3\right)\cdot\left(1+(1-p)^3\cdot\left(1-(1-p)^{w-3}\right)\right)}{1-(1-p)^w}\right), \tag{23}$$

which can be approximated for small values of p as

$$\hat{Q}(w) \approx \min\left(1, \frac{3}{w}\right), \quad Q \approx \hat{Q}\left(E[W]\right), \tag{24}$$

where $E[W]$ is given in (19).

According to [3], $E[R]$ can be defined as

$$E[R] = \frac{1}{1-p}. \tag{25}$$

Note that in contrast to [3], the duration of the first timeout from the sequence of consecutive timeouts is incorporated in the duration of a cycle. Therefore, the duration of the sequence of timeouts (excepting the first timeout) is

$$L_k = \begin{cases} \left(2^k - 2\right) \cdot RTO, & \text{when } k \in [2, 6], \\ \left(62 + 64 \cdot (k-6)\right) \cdot RTO, & \text{when } k \geq 7, \end{cases} \tag{26}$$

and the expectation of Z^{TO} is

$$\begin{aligned} E\left[Z^{TO}\right] &= \sum_{k=2}^{\infty} L_k \cdot p^{k-1} \cdot (1-p) = \\ &= \overline{RTO} \cdot \frac{2 \cdot p + 2 \cdot p^2 + 4 \cdot p^3 + 8 \cdot p^4 + 16 \cdot p^5 + 32 \cdot p^6}{1-p}. \end{aligned} \tag{27}$$

Combining (5), (20), (23) and (27), we obtain

$$B = \frac{\frac{1}{p} + E[W] + \hat{Q}\left(E[W]\right)\frac{p}{1-p}}{\overline{RTT}\left(\max\left(\log_2 E[W], 2\right) + b\left(\frac{E[W]}{2} + 1\right) + 2 + \frac{\overline{RTO}}{\overline{RTT}}\right) + \hat{Q}\left(E[W]\right)\left(\overline{RTO}\frac{f(p)}{1-p} - \overline{RTT}\right)}, \tag{28}$$

where

$$f(p) = 2 \cdot p + 2 \cdot p^2 + 4 \cdot p^3 + 8 \cdot p^4 + 16 \cdot p^5 + 32 \cdot p^6. \tag{29}$$

3.3 The Impact of Receiver Window Size

Let us denote by $W_{\max}$ the receiver window size and by $E[W_u]$ the unconstrained window size. Similarly to [3], we assume that $E[W_u] < W_{\max}$ leads to $E[W_u] \approx E[W]$ (where $E[W]$ is from (19)) and $E[W_u] \geq W_{\max}$ leads to $E[W] \approx W_{\max}$. Thus, using derivation from [3] and taking into account that

$$E\left[Y^{CA}\right] = \frac{3 \cdot b \cdot \left(W_{\max}\right)^2}{8} - \frac{b \cdot W_{\max}}{4} + E[V] \cdot W_{\max} + \frac{W_{\max} - 1}{2}, \tag{30}$$

we obtain the following system of equations

$$\begin{cases} E[Y] = \dfrac{1}{p} + W_{\max} \\ E[Y] = W_{\max} - 1 + \dfrac{3 \cdot b \cdot \left(W_{\max}\right)^2}{8} - \dfrac{b \cdot W_{\max}}{4} + E[V] \cdot W_{\max} + \dfrac{W_{\max} - 1}{2} + 1 \end{cases} \tag{31}$$

Hence, the expected number of rounds when the window size remains constant is

$$E[V]=\frac{4+b\cdot p\cdot W_{\max}+2\cdot p}{4\cdot p\cdot W_{\max}}-\frac{1}{2}-\frac{3\cdot b\cdot W_{\max}}{8} \tag{32}$$

and

$$E[X]=\frac{b\cdot W_{\max}}{8}+\frac{4+b\cdot p\cdot W_{\max}+2\cdot p}{4\cdot p\cdot W_{\max}}-\frac{1}{2}. \tag{33}$$

Therefore

$$E[A]=\overline{RTT}\cdot\left(\max\left(\log_2 W_{\max},2\right)+\frac{b\cdot W_{\max}}{8}+\frac{4+b\cdot p\cdot W_{\max}+2\cdot p}{4\cdot p\cdot W_{\max}}+\frac{3}{2}+\frac{\overline{RTO}}{\overline{RTT}}\right). \tag{34}$$

In conclusion, the complete expression of TCP throughput can be represented by the following expression

$$B=\begin{cases}\dfrac{\frac{1}{p}+E[W]+\hat{Q}(E[W])\frac{p}{1-p}}{\overline{RTT}\left(\max\left(\log_2 E[W],2\right)+b\left(\frac{E[W]}{2}+1\right)+2+\frac{\overline{RTO}}{\overline{RTT}}\right)+\hat{Q}(E[W])\left(\overline{RTO}\frac{f(p)}{1-p}-\overline{RTT}\right)}, & \text{when } W_{\max}>E[W],\\ \dfrac{\frac{1}{p}+W_{\max}+\hat{Q}(W_{\max})\frac{p}{1-p}}{\overline{RTT}\left(\max\left(\log_2 W_{\max},2\right)+\frac{bW_{\max}}{8}+\frac{4+bpW_{\max}+2p}{4pW_{\max}}+\frac{3}{2}+\frac{\overline{RTO}}{\overline{RTT}}\right)+\hat{Q}(W_{\max})\left(\overline{RTO}\frac{f(p)}{1-p}-\overline{RTT}\right)}, & \text{when } W_{\max}\le E[W].\end{cases} \tag{35}$$

4 Model Validation Through Simulation

In order to validate the proposed model and compare it with the one presented in [3], we compared the results obtained from the both analytical models against simulation results obtained from ns-2 [13]. We performed experiments using the well-known single bottleneck ("dumbbell") network topology. In this topology all access links have a propagation delay of 1 ms and a bandwidth of 10 Mbps. The bottleneck link is configured as a Drop Tail link and has a propagation delay of 8 ms, bandwidth of 2 Mbps and a buffer size of 50 packets. To model TCP Reno connection we used Agent/TCP/Reno as a TCP Reno sender, Agent/TCPSink/DelAck as a TCP receiver with delayed acknowledgement algorithm and FTP as an application for transmitting infinite amount of data. We set TCP segment size to be 1460 bytes and maximum receiver window size ($W_{\max}$) to be 10 segments.

It has been noted in [14] that Web-traffic tends to be self-similar in nature and it was shown in [15] that superposition of many ON/OFF sources whose ON/OFF times are independently drawn from heavy-tailed distributions such as Pareto distribution can produce asymptotically self-similar traffic. Thus, we modeled the effects of competing Web-like traffic and high level of statistical multiplexing as a superposition of

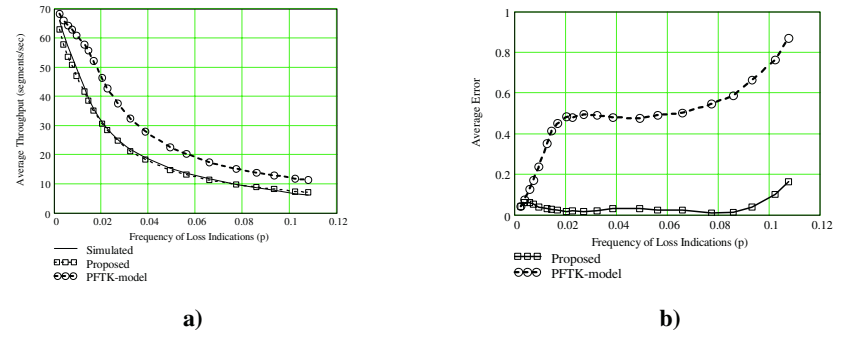
many ON/OFF UDP sources. Similarly to [16], in our experiments we set the shape parameter of Pareto distribution to be 1.2, the mean ON time to be 1 second and the mean OFF time to be 2 seconds. During ON times the UDP sources transmit with the rate of 12 kbps. The number of UDP sources was varied between 220 and 420 with the step of 10 sources. For each step we ran 100 simulation trials with a simulation time of 3600 seconds for each trial. In order to remove transient phase at the beginning of simulation, the collection of data was started after 60 seconds from the beginning of the simulation.

As in [3], in order to estimate the value of p, we used the ratio of the total number of loss indications to the total number of segments sent as an approximate value of p. Fig. 3a compares our model and the one presented in [3] against the simulation results. It easy to see, that the predicted values of throughput by the proposed model are much closer to the simulation results.

To quantify the accuracy of the both analytical models we computed the average error using the following expression from [3]:

$$\text{Average error} = \frac{\sum_{\text{observations}} \frac{\left| B(p)_{\text{predicted}} - B(p)_{\text{observed}} \right|}{B(p)_{\text{observed}}}}{\text{number of observations}}. \tag{36}$$

As shown in Fig. 3b, the proposed model has the average error smaller than 0.05 over a wide range of loss rates.



a) **b)**

Fig. 3. Average throughput (a) and average error (b) of the proposed and PFTK models

5 Conclusion

In this paper we developed an analytical model for predicting TCP Reno throughput in the presence of correlated losses. The model is based on the one proposed in [3] and improves it by taking into consideration a fast retransmit/fast recovery dynamics and slow start phase after timeout. The presented model has the average error smaller than 0.05 over a wide range of loss rates with the mean of 0.03, while the one, pro-

posed in [3] performs well when the loss rate is quite small and significantly overestimates TCP Reno throughput in the middle-to-high loss rate range.

References

1. M. Fomenkov, K. Keys, D. Moore, and k. claffy. Longitudinal study of Internet traffic in 1998-2003. Technical Report, Cooperative Association for Internet Data Analysis (CAIDA), 2003.
2. J. Olsen. Stochastic modeling and simulation of the TCP protocol. PhD thesis, Uppsala University, Sweden, 2003.
3. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. IEEE/ACM Transactions on Networking, vol. 8, no. 2, pp. 133-145, April 2000.
4. S. Fortin and B. Sericola. A Markovian model for the stationary behavior of TCP. Technical Report RR-4240, IRISA-INRIA, France, September 2001.
5. B. Sikdar, S. Kalyanaraman, and K. Vastola. An integrated model for the latency and steady state throughput of TCP connections. Performance Evaluation, vol. 46, no. 2-3, pp. 139-154, October 2001.
6. O. Bogoiavlenskaia, M. Kojo, M. Mutka, and T. Alanko. Analytical Markovian model of TCP congestion avoidance algorithm performance. Technical Report C-2002-13, University of Helsinki, Finland, April 2002.
7. N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. Proc. IEEE INFOCOM'00, vol. 3, Tel Aviv, pp. 1742-1751, March 2000.
8. S. Fu and M. Atiquzzman. Modeling TCP Reno with spurious timeouts in wireless mobile environment. Proc. 12-th International Conference on Computer Communications and Networks, Dallas, October 2003.
9. M. Allman, V. Paxson, and W. Stevens. TCP congestion control. IETF RFC 2581, April 1999.
10. R. Braden. Requirements for Internet hosts. IETF RFC 1122, October 1989.
11. V. Paxson and M. Allman. Computing TCP's retransmission timer. IETF RFC 2988, November 2000.
12. K. Fall and S. Floyd. Simulation-based comparison of Tahoe, Reno and SACK TCP. ACM SIGCOMM Computer Communication Review, vol. 26, no. 3, pp. 5-21, July 1996.
13. UCB/LBNL/VINT. The network simulator - ns-2. http://www.isi.edu/nsnam/ns/
14. K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols and self-similar network traffic. Technical Report 1996-016, Boston University, USA, August 1996.
15. W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. IEEE/ACM Transactions on Networking, vol. 5, no. 1, pp. 71-86, February 1997.
16. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. A model-based TCP-friendly rate control protocol. Proc. NOSSDAV, Basking Ridge, pp. 137-151, June 1999.

Examining TCP Parallelization Related Methods for Various Packet Losses

Qiang Fu and Jadwiga Indulska

School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, QLD 4072, Australia
{qiangfu, jaga}@itee.uq.edu.au

Abstract. The diversity of the networks (wired/wireless) prefers a TCP solution robust across a wide range of networks rather than fine-tuned for a particular one at the cost of another. TCP parallelization uses multiple virtual TCP connections to transfer data for an application process and opens a way to improve TCP performance across a wide range of environments - high bandwidth-delay product (BDP), wireless as well as conventional networks. In particular, it can significantly benefit the emerging high-speed wireless networks. Despite its potential to work well over a wide range of networks, it is not fully understood how TCP parallelization performs when experiencing various packet losses in the heterogeneous environment. This paper examines the current TCP parallelization related methods under various packet losses and shows how to improve the performance of TCP parallelization.

1 Introduction

TCP parallelization uses a set of parallel TCP connections to transfer data for an application process and opens a way to improve TCP performance across a wide range of environments - high bandwidth-delay product (BDP) [1] [2] [11] [12], wireless [5] [6] as well as conventional networks [3] [4].

In [5], a brief test, which involved bit errors due to the purposely degraded RF performance of the earth station in satellite systems, showed that given the same effective window of 192Kbytes better throughput was achieved when a large number of connections were used. Unfortunately there was no further analysis. FEC is widely used for wireless data transmission. It is well known that a small amount of FEC gives the most efficient throughput gain (but not the best throughput) [6]. To fully utilize the available bandwidth, a large amount of FEC has to be added. This wastes a lot of bandwidth. Therefore, opening multiple TCP connections with a small amount of FEC could be an efficient way.

To eliminate the complexity of maintaining multiple connections, some solutions use a single TCP connection to emulate the behavior of a set of standard TCP connections. MulTCP [7] is a typical example, which makes one logical connection behave like a set of standard TCP connections to achieve weighted proportional fairness. Although the purpose of MulTCP is not to achieve high performance in

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 54–64, 2005.

high-BDP networks, its essence has been inherited by Scalable TCP [8], HighSpeed TCP [9] and FAST TCP [10] which are designed for high-BDP networks.

Most of the solutions mentioned above focus on wired networks, especially high-BDP networks and the solutions implemented in the wireless environments do not fully explore the potential of TCP parallelization. In this paper, we examine some of the solutions and explore the potential of TCP parallelization under various packet losses which reflect the heterogeneity of wired/wireless networks. We have a particular interest in MulTCP [7] and the fractional method in [11] [12]. The fractional method can be a representative of the solutions using parallel connections to improve effectiveness in utilizing while maintaining fairness. MulTCP can be a representative of using a single connection to emulate parallel connections.

2 Analytical Models and Simulation Environment

We have developed the bandwidth model shown in (1) based on a set of parallel TCP connections [13] [14]. W_n, n, p, MSS and RTT denote the aggregate window of a set of parallel connections, the number of parallel connections, packet loss rate, Maximum Segment Size and Round-Trip Time, respectively. c is referred to as window reduction factor. In response to a packet loss the window of the involved connection is cut down by $1/c$. m is referred to as window increase factor. The aggregate window (W_n) opens m packets per RTT. The simplicity of the model clearly shows the relationship between the key elements such as window increase factor (m) and window reduction factor (c) and how these elements impact the throughput and aggressiveness of the set of parallel connections.

$$BW_n = \frac{(1/p) * MSS}{(W_n/cmn) * RTT} = \sqrt{\frac{m(2cn-1)}{2p}} \times \frac{MSS}{RTT} \tag{1}$$

Without considering the impact of Fast Recovery and timeouts, the model in (1) cannot show if a single logical connection can achieve the equivalent throughput when emulating a set of parallel connections and if a set of multiple connections can give a better performance when emulating a single standard connection. In (2), a model is presented which takes TCP's Fast Recovery into account. More details about this model are available in [13].

$$\left[2\frac{(cn-1)W_n}{cn} + \frac{m(n_e-1)}{n_e} + \frac{W_n}{cn}\right] \times \left(\frac{W_n}{cmn} - \frac{n_e-1}{n_e}\right) \times \frac{1}{2} + \left[2\frac{(cn-1)W_n}{cn} + \frac{m(n_e-1)}{n_e}\right] \times 1 \times \frac{1}{2} = \frac{1}{p}$$

$$BW_n = \frac{(1/p) * MSS}{\left\{1 + \left(\frac{W_n}{cmn} - \frac{n_e-1}{n_e}\right)\right\} * RTT} \tag{2}$$

In (2), n_e represents the number of connections used to emulate a set of TCP connections. The model in (2) is more accurate than the one in (1) to predict the performance of a set of standard TCP connections [13]. Because the model does not consider the effects of timeouts, we do not expect that it can predict the performance well for a single connection emulating a set of standard TCP connections. Similar to

fast recovery, in this case timeouts could significantly affect the throughput performance (compared to the direct use of a set of standard TCP connections). However, the model can show how the performance is affected in these two cases.

Table 1. Simulation parameters

TCP Scheme	*Reno*
Link Capacity	28.6/8Mbps
Link Delay	100ms (5+90+5)
Packet Size (MSS)	1,000bytes
Buffer Size	Unlimited/Limited
Error (*p*)	Uniform/Two-state Markov (0.01/0.001)
Simulation Time	100/1,000s

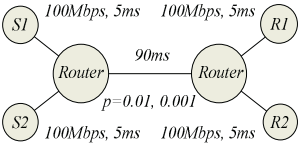


Fig. 1. Simulation topology

The simulations are run on the topology shown in Fig. 1. NS-2 is used to perform all the simulations. Table 1 is a concise list of the system parameters selected for the simulations. The studied connections are established between sending host *S1* and receiving host *R1*. The cross or competing traffic is created between sending host *S2* and receiving host *R2*. A buffer is used on the bottleneck link between the two routers. The buffer size is set unlimited for studying uniform/bursty losses and set limited when investigating buffer overflow losses.

3 Uniform Errors

3.1 Fractional Method

In [11] and [12], a *fractional approach* is introduced. In this scheme, each one of n flows increases its congestion window by one packet per n packets acknowledged and only one of n parallel streams will decrease its window by half in response to a packet loss. Referring to (1), we define Fractional Multiplier (FM) as: $FM=n/m$. If we take the n flows as a whole the approach requires the aggregate window to increase 1 packet per RTT, that is, $m=1$ and thus $FM=n$. This method does reduce the aggressiveness of the parallel connection by increasing the aggregate window 1 packet per RTT as a single standard TCP connection. However, it only reduces the aggregate window by $1/2n$ rather than 1/2 in response to a packet loss.

A *combined approach* is also introduced in [11] and [12]. The method combines a single standard TCP connection with a set of parallel connections that opens their window very conservatively. The parallel flows are modified based on the previously mentioned *fractional approach* with a little difference. Each of the n conservative flows increases its congestion window by one packet for every FM_C*n packets it receives. *FM_C* is referred to as combined fractional multiplier. The *FM_C* with a value >1 will ensure that the set of fractional streams is less aggressive than if they are modified by the *fractional approach*.

We carried out simulations based on these two approaches. In the simulations, the *fractional* and *combined approaches* compete with 5 standard TCP connections, respectively. We use 5 standard TCP connections as the cross traffic because this approach is also used in [11] and [12]. Fig. 2 and Fig. 3 compare the modeled and the simulation results. The model in (1) is used. For Fig. 2, the packet loss rate and the bottleneck bandwidth are set to 0.001 and 8Mb/s, respectively, to guarantee that the network is congested from time to time. For Fig. 3, the packet loss rate is set to 0.01 and the bottleneck bandwidth is 28.6Mb/s. The loss rate and bandwidth are high enough so that the bottleneck bandwidth cannot be fully utilized.

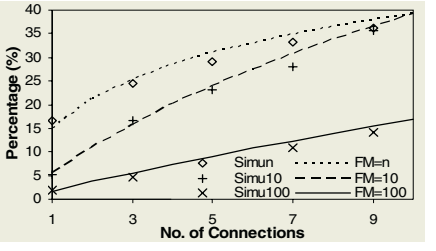


Fig. 2. Throughput share (p=0.001)

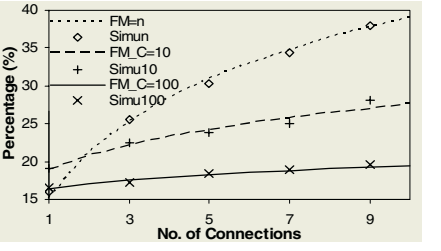


Fig. 3. Throughput share (p=0.01)

If all the schemes used in the figures had the same aggressiveness, then the studied connections would have a throughput share of 1/(1+5), that is, 16.67%. Fig. 2 shows the performance of the *fractional approach* with various *FM* schemes. Fig. 3 shows the throughput share achieved by the *fractional approach* and the *combined approach*. It shows the modeled results match the simulation results well. The good match indicates that the fairness for these two approaches is determined by their effectiveness to increase throughput because the model in (1) is actually a reflection of this effectiveness. These two approaches do not fundamentally address the concern how to maintain fairness in presence of congestion and improve effectiveness in absence of congestion.

3.2 Single Connection Emulating TCP Parallelization

MulTCP can be perceived as a single connection that emulates a set of standard TCP connections and is similar to the single connection based approach that we will examine later. For instance, if a single connection emulates a set of n standard TCP connections, then in (1) and (2) m=n and c=2n. The model in (1) cannot show the performance difference between a single connection emulating a set of parallel connections and the direct use of a set of parallel connections, because it ignores TCP's fast recovery. Therefore the model in (2) is used. In the figures, *n_emu_* denotes n_e in (2), which is the actual number of connections used to emulate the behavior of a set of parallel connections. In Fig. 4, the bottleneck bandwidth is set to 28.6Mb/s while the packet loss rate is set to 0.01. This makes the bottleneck bandwidth always underused. In Fig. 5, the packet loss rate is reduced to 0.001 to

examine the fairness performance in the congested network. The figures examine the single connection based approach (*n_emu_*=1) and the 5 connection based approach (*n_emu_*=5) that emulate the behavior of *n* standard TCP connections. The cross traffic is made up by the same number (that is, *n*) of standard TCP connections. Therefore, for desired fairness the throughput share achieved by the studied approaches should be 50%. However, Fig. 4 shows that the single connection based approach fails to achieve the desired fairness. By using 5 connections, we find that the fairness is much improved. We notice the difference between the modeled and the simulation results. We attribute the difference to the timeouts ignored in the model. The similar performance is also observed in Fig. 5.

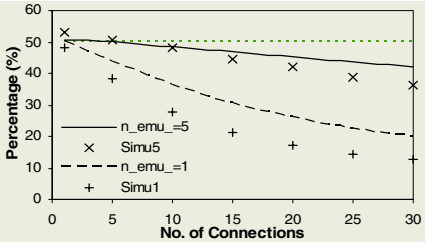


Fig. 4. Throughput share (*p*=0.01)

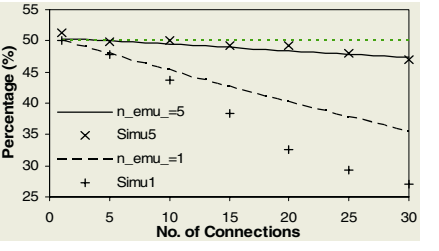


Fig. 5. Throughput share (*p*=0.001)

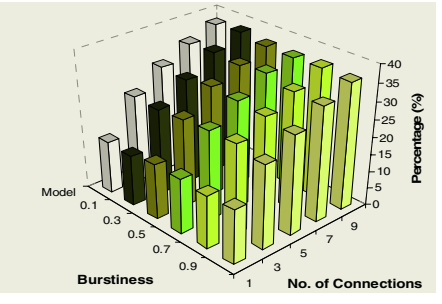
4 Bursty Errors

Depending on the bit interleaving depth and channel characteristics, the burstiness of packet losses in wireless links tends to vary. Serious bursty losses could cause synchronization of packet losses between the parallel connections. In this section we study the impacts of burstiness of packet losses on TCP parallelization. We use a discrete-time Markov chain with two states (Bad and Good) to model the error characteristics of the wireless link. Such a model is often used in the literature to analyze losses on wireless links [6] [15] [16] [17]. A packet is lost if the packet is transmitted over the link while the link is in the Bad state, otherwise it is supposed to be correctly received. Suppose that the link is currently in the Bad state for a given time unit. The burstiness is represented by the probability that the link stays in the Bad state for the next time unit. Note that the time unit can be replaced by a packet.

In this section, we set the bottleneck bandwidth to 8Mbps. The average probability of time unit being in the bad state, if not mentioned, is set to 0.01. The length of the time unit can destroy up to 4 back-to-back packets in the bad state. The burstiness of packet losses varies from 0.1 to 0.9. Please note that even if the burstiness is set to 0.1, the loss pattern is significantly different from the uniform losses. This is because one time unit in the bad state can destroy up to 4 back-to-back packets and the uniform distribution of the time units in the bad state requires a burstiness of 0.01.

4.1 Fractional Method

Because of the similarity of the *combined approach* and *fractional approach*, we only examine the performance of the *fractional approach*. Except for the packet loss pattern and bottleneck bandwidth, the simulation environment is same to the corresponding section for the uniform packet losses. In Fig. 6 the packet loss rate is set to 0.001 to study the performance when the network is congested while in Fig. 7 it is set to 0.01 to study the performance when the bottleneck bandwidth is underused. The figures show the similar performance across various burstinesses and between the two different packet loss rates. Furthermore, the simulation results match well the modeled results. Because the simulation performance under the uniform losses matches well the modeled performance too, there is no clear difference between the throughput share performance under uniform and bursty losses.



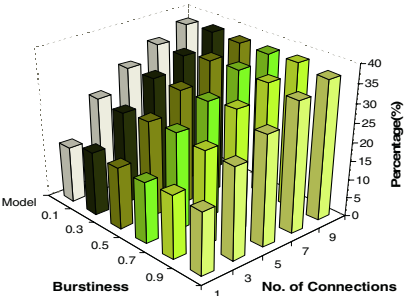


Fig. 6. Throughput share vs. Burstiness (0.001) **Fig. 7.** Throughput share vs. Burstiness (0.01)

4.2 Single Connection Emulating TCP Parallelization

Same to the corresponding section for the uniform losses, the single connection based approach and the 5 connection based approach are examined. The desired throughput share achieved by the studied approaches should be 50%. The setting of bottleneck bandwidth (8Mbps) and packet loss rate (0.01) enables the bottleneck to change from underused to congested as the emulated number of connections increases. Fig. 8 shows the throughput shares achieved by the single connection based approach are similar for the burstiness ranging from 0.1 to 0.9. The similar situation is observed for 5 connection based approach in Fig. 9. Similar to the observations under uniform losses, the 5 connection based approach can achieve a throughput share much closer to 50% than the single connection based approach. We also notice that the throughput share achieved by the studied approach under bursty losses is significantly and consistently lower than under uniform losses. This indicates that under bursty losses it is more difficult for the single connection based approach to achieve a performance equivalent to the direct use of a set of parallel connections. So is it for the 5 connection based approach.

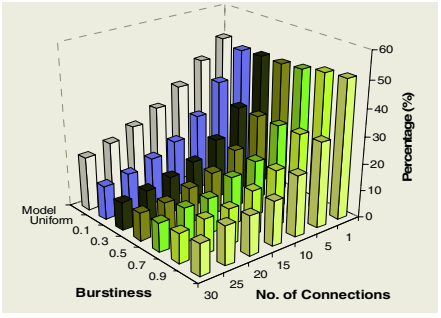


Fig. 8. Throughput share vs. Burstiness (1)

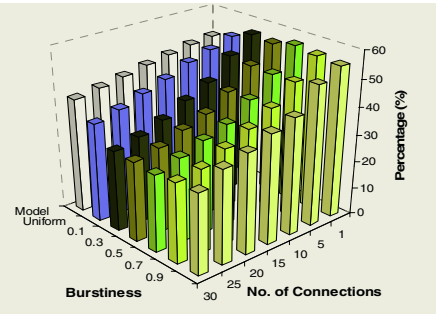


Fig. 9. Throughput share vs. Burstiness (5)

5 Buffer Overflow Errors

In this section, we use a limited buffer on the bottleneck link between the two routers to study the effect of buffer size on the throughput performance.

5.1 Fractional Method

Only the *combined approach* is examined, because of its similarity to the *fractional approach*. The simulations are carried out for 100 and 1000 seconds. In the figures, the legend items with a suffix of "s" indicate the simulation time is 100s, otherwise it is 1000s. The buffer size is indicated by the number in the legend items and the unit is packet. For instance, B1000s means that the buffer size is 1000 packets and the simulation time is 100s. Fig. 10 shows dramatically different performance with the various buffer sizes and simulation times. In terms of simulation time, the throughput share achieved with the 100 second simulation is much lower than if the simulation time is 1000s. In terms of buffer size, a small buffer size gives the *combined approach* a better performance than a large buffer size. Recall that the set of parallel connections increases its aggregate window less than 1 packet per RTT. This loss of aggressiveness is compensated by the more aggressive window reduction factor: in response to a packet loss its aggregate window will only be closed by $1/2n$ (n denotes the number of parallel connections in the set). Let us assume that the buffer size is very large so that for a given time there are no packet losses. In this case, the set of parallel connections cannot take the advantage of its more aggressive window reduction factor and only the less aggressive window increase factor stands. As a result, the *combined approach* will give a throughput share less than the one predicted by the model.

As the buffer size decreases, packet losses occur increasingly often. This brings back the advantage of the aggressive window reduction factor that the set of parallel connections is supposed to enjoy. Therefore, if for a given simulation time the buffer size is small enough or for a given buffer size the simulation time is long enough so that there are regular packet losses, then the throughput share achieved by the *combined approach* is expected to match well the modeled results. However, it does

not happen in the figure. The reason is that the analysis so far is limited to the assumption that all the packets have the equal probability of a packet loss and this assumption does not hold. By using the combined fractional multiplier (*FM_C*), the set of parallel connections is much less aggressive than the standard TCP connection and only opens its aggregate window by 1/10 packets per RTT. Therefore, the standard TCP connections create burstier traffic and are more susceptible to packet losses. When the simulation time is short the set of parallel connections has not yet fully enjoyed the advantage of the lower loss rate before the simulation run is completed, because of its too conservative window increase factor. When the simulation run is finished, its aggregate window has not yet been able to reach its full (maximum) size. As the simulation time gets longer, its aggregate window becomes larger and yields higher throughput before a packet loss occurs. This is why the throughput share with the simulation time of 1000s is much higher than the one achieved when the simulation time is 100s. As the set of parallel connections has a lower packet loss rate, it is understandable that the throughput share achieved by the *combined approach* is higher than the modeled throughput share if the buffer size is small enough or the simulation time is long enough to guarantee regular packet losses.

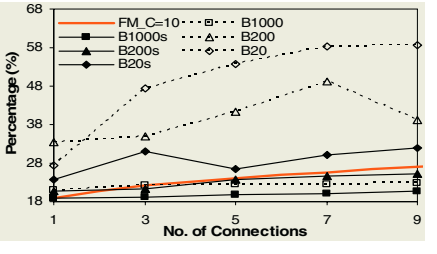


Fig. 10. Throughput share

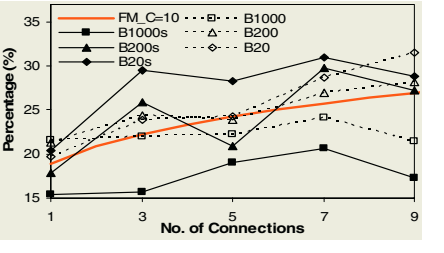


Fig. 11. Throughput share with UDP traffic

The traffic in Fig. 10 is self-similar traffic, because all the traffic is created by competing TCP flows. In Fig. 11, we introduce UDP traffic, which is created by a single UDP ON/OFF source (exponential). The sending rate is 1.6Mbps when the source is ON and thus takes up 20% bandwidth. Both the ON and OFF periods are set to 250ms. It shows that with the simulation time of 100s the performance of the *combined approach* is not stable and is similar to the performance in Fig. 10. However, when the simulation time is 1000s, the performance converges towards the curve predicted by the model. This means that the UDP traffic changes the packet loss pattern of the competing flows. The UDP traffic has a pacing effect on the TCP traffic and thus closes the difference of packet loss rate between the standard TCP flows and the set of parallel connections. As a result, the throughput share achieved by the *combined approach* is close to the modeled throughput share.

5.2 Single Connection Emulating TCP Parallelization

In contrast to the fractional method, the single connection based approach is more aggressive than the standard TCP connections. As a result, the finding is opposite to the one observed for the fractional method. Fig. 12 shows the performance of the single connection based approach with and without UDP traffic. The legend items with a suffix of "b" indicates that the UDP traffic mentioned in the previous section is introduced. It shows that high throughput share is achieved by the single connection based approach with large buffer size because large buffer size means lower packet loss rate. Furthermore, the UDP traffic makes the throughput share higher than if no UDP traffic is introduced. Although the single connection based approach has the same level of aggressiveness as the total competing TCP flows, it creates more back-to-back packets for the same number of returning ACKs. Therefore its traffic is burstier and thus more likely to cause buffer overflow (packet losses). Once again, the UDP traffic closes the difference of packet loss rate between the competing TCP flows, and thus, the throughput share achieved by the single connection based approach with UDP traffic is higher than without UDP traffic. With UDP traffic introduced, Fig. 13 compares the single connection based approach with the 4 connection based approach. The legend items with a suffix of "_1" denote the single connection based approach. Besides the advantage of the use of a small number of connections to emulate a set of standard TCP connections shown in the previous sections under random packet losses, the 4 connection based approach can reduce traffic burstiness.

The 4 connection based approach can not only improve its throughput share, but also the efficiency to utilize the bottleneck bandwidth. Fig. 14 shows the total throughput achieved by the studied approach along with its competing TCP flows. For dark color columns the 4 connection based approach is the studied approach while the light color columns on the immediate right indicate the performance of the corresponding single connection based approach. For example, 4/20 indicates that the 4 connection based approach is performed with a buffer size of 20 packets while 1/20 indicates the performance of its corresponding single connection based approach. It clearly shows that the 4 connection based approach always has a better utilization of the bottleneck bandwidth than the single connection based approach. The smaller the buffer size is, the larger the improvement gain on the utilization is.

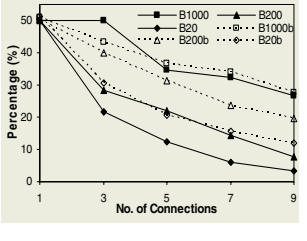


Fig. 12. Throughput share: UDP traffic vs. no UDP traffic

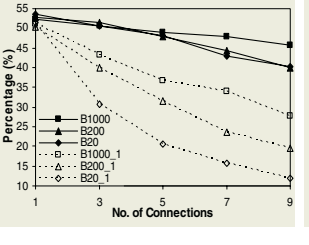


Fig. 13. Throughput share with UDP traffic

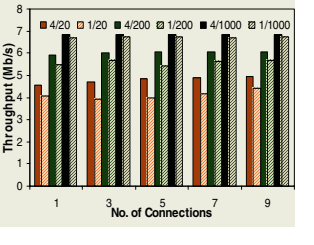


Fig. 14. Utilization of bottleneck bandwidth

6 Conclusions

In this paper, we analyzed TCP parallelization related methods for various packet losses. For the fractional method, the analysis shows that our model is capable to predict the performance of the method under uniform, bursty and buffer overflow losses. It shows that the fractional method does not really address the concern how to maintain fairness in presence of congestion and improve effectiveness in absence of congestion: the fairness and the effectiveness are achieved at the cost of each other. For a given context our model can help determine the balance point, where the desired effectiveness/fairness can be achieved at an acceptable cost of one another. We also examined how buffer size and competing traffic affect the performance of the fractional method. It shows that the performance can be different over various buffer sizes. The achieved fairness and effectiveness can be deviated significantly from the expected level. In [11] [12], the *combined approach* was considered more suitable than *fractional approach*. Our model and analysis do not show that there is a mechanism in favor of the *combined approach*.

For the single connection based approach represented by MulTCP [7], the analysis shows that the single connection based approach cannot achieve the equivalent performance by the direct use of the set of parallel connections. The model, which takes fast recovery into account, clearly shows how the performance is affected by using the single connection based approach. In high speed and error-prone networks, it is too risky to use this method. It can cause tremendous performance penalty due to unnecessary congestion control and timeouts. With the same level of aggressiveness, the single connection based approach creates burstier traffic than TCP parallelization, and thus is more prone to buffer overflow losses. The model shows that using a small number of connections to emulate a set of parallel TCP connections can achieve a performance close to the performance of the set of parallel connections. This method has the advantages of TCP parallelization while reducing the complexity of managing a large number of connections. The method can reduce the burstiness of its traffic and thus performance degradation by buffer overflow losses. Compared to the single connection based approach, it can not only improve the throughput performance, but also the efficiency to utilize the bottleneck bandwidth.

References

1. B. Allcock, ed al., “Data Management and Transfer in High-Performance Computational Grid Environments”, *Parallel Computing*, 28(5), 2002.
2. R. Grossman, ed al., “Experimental Studies Using Photonic Data Services at IGrid 2002”, *Future Computer Systems*, 19(6), 2003.
3. H. Balakrishnan, ed al., “An Integrated Congestion Management Architecture for Internet Hosts”, *ACM SIGCOMM*, Sept. 1999.
4. L. Eggert, J. Heidemann, and J. Touch, “Effects of Ensemble–TCP”, *ACM Computer Communication Review*, 30(1), 2000.
5. M. Allman, H. Kruse and S. Ostermann, “An Application-Level Solution to TCP’s Satellite Inefficiencies”, *WOSBIS*, Nov. 1996.

6. C. Barakat, E. Altman, “Bandwidth Tradeoff between TCP and Link-Level FEC”, *Computer Networks*, 39(2), June 2002.
7. J. Crowcroft and P. Oechslin. “Differentiated end-to-end Internet services using a weighted proportionally fair sharing TCP”, *Computer Comm. Review*, 28(3), 1998.
8. Tom Kelly, “Scalable TCP: Improving performance in highspeed wide area networks”, *Computer Communication Review* 32(2), April 2003.
9. Sally Floyd, “HighSpeed TCP for Large Congestion Windows”, *RFC 3649*, Dec. 2003.
10. C. Jin, D. Wei and S. Low, “FAST TCP: motivation, architecture, algorithms, performance”, *INFOCOM*, 2004.
11. T. Hacker, B. Noble and B. Athey, “The Effects of Systemic Packet Loss on Aggregate TCP Flows”, *Supercomputing*, 2002
12. T. Hacker, B. Noble, B. Athey, “Improving Throughput and Maintaining Fairness using Parallel TCP”, *INFOCOM*, 2004.
13. Q. Fu, J. Indulska, “The Impact of Fast Recovery on Parallel TCP connections”, *HET-NETs* 2004, Ilkley, UK.
14. Q. Fu, J. Indulska, “Features of Parallel TCP with Emphasis on Congestion Avoidance in Heterogeneous Networks”, *Advanced Wired and Wireless Networks*, pp. 205-228, eds. T. Wysocki, A. Dadej and B. Wysocki, Springer-Verlag 2004.
15. H. Chaskar, T. V. Lakshman, and U. Madhow, “On the Design of Interfaces for TCP/IP over Wireless”, *MILCOM*, 1996.
16. A. Chockalingam, M. Zorzi, and R.R. Rao, “Performance of TCP on Wireless Fading Links with Memory”, *ICC*, 1998.
17. E.N. Gilbert, “Capacity of a Burst-Noise Channel”, Bell Sys. Tech. Journal, Sept. 1960.

The Interaction Between Window Adjustment Strategies and Queue Management Schemes

Chi Zhang[1] and Lefteris Mamatas[2]

[1] School of Computer Science, Florida International University,
Miami, FL 33139, USA
czhang@cs.fiu.edu
[2] Department Of Electrical and Computer Engineering,
Demokritos University, Xanthi 67100, Greece
emamatas@ee.duth.gr

Abstract. In this paper, we investigate extensively the joint network dynamics with different AIMD window-adjustment parameters on end-hosts, and different queue management schemes (i.e. DropTail vs. RED) in routers. We reveal that with DropTail buffer, although smooth TCPs causes less queuing-delay jitter, its average queuing delay is significantly higher than that of responsive TCPs. The direct implication of this discovery is that when mobile users of media-streaming and short messages share a bottleneck link, the energy consumption for sending short messages can increase severely if media-streaming users adopt smooth TCPs. With RED, on the other hand, smooth TCPs not only lead to smaller queue oscillation, the average/max queue length is smaller as well.

1 Introduction

In computer networks, commodity routers and switches often use FIFO buffers to multiplex packets from different flows. Computer networks thus rely on the congestion control algorithms on end-hosts to 'probe' available bandwidth, avoid persistent congestion, and achieve system fairness. The congestion control algorithm of TCP [1] is based on the Additive Increase / Multiplicative Decrease (AIMD) window adjustment strategy [2], to reach satisfactory system equilibrium in a distributed fashion.

While TCP congestion control is appropriate for bulk data transfers, media-streaming applications find the standard multiplicative decrease by a factor of 2 upon congestion to be unnecessarily severe, as it can cause serious throughput oscillations [3]. Authors in [6] investigated the impact of transport protocols on real-time application QoS. Since throughput smoothness is crucial to the subjective performance of multimedia applications, TCP-friendly protocols [3,10] have been proposed with two fundamental goals: (i) to achieve smooth downward adjustments; this is done by increasing the window decrease ratio during congestion, and (ii) to compete fairly with TCP flows; this is approached by reducing the window increase step according to a steady-state TCP throughput equation. TCP friendly protocols

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 65–74, 2005.

favor *smoothness* by using a gentle backward adjustment upon congestion, at the cost of lesser *responsiveness* - through moderated upward adjustments. In this research, we will study one family of TCP-friendly protocols: TCP(α,β) protocols [10], which parameterize the additive increase value α and multiplicative decrease ratio β of AIMD. The authors in [10] incorporated α and β into a TCP throughput equation, and derived a rough guide for appropriately selecting α and β to achieve TCP friendliness:

$$\alpha = 4(1-\beta^2)/3 \tag{1}$$

Based on experiments, they propose a β = ⅞ as the appropriate ratio for downward window adjustments upon congestion (i.e. smoother than standard TCP). With β = ⅞, equation (1) gives an increase value α=0.31 (i.e. less responsive than TCP). We are interested in the family of *TCP-friendly* TCP(α,β) protocols that follow equation (1), because they make tradeoffs between responsiveness and smoothness, and provide a good opportunity to acquire interesting and useful insights into the strategy of window adjustments: By tuning the protocol parameters α and β, we can watch the trends of protocol behaviors under various network and traffic conditions. We categorize three classes of TCP-friendly TCP(α, β) protocols: (i) Standard TCP(1, ½); (ii) Responsive TCP is TCP(α, β) with *relatively* low β value and high α value; (iii) Smooth TCP is TCP(α, β) with *relatively* high β value and low α value.

At the router side, Active Queue Management (AQM) has been recommended for overcoming the two important drawbacks of the straightforward "Drop-Tail" buffer [4]: (i) Lock-Out: In some situations drop-tail buffer allows a single connection or a few flows to monopolize queue space, preventing other connections from getting resources. (ii) Full Queues: Drop-tail strategy tends to keep the queue in a (almost) full status for a long period of time. A persistent large queue increases the end-to-end delay. Random Early Detection (RED) [4] is an AQM scheme dropping packets from among various flows randomly before the gateway's queue overflows, when the average queue length starts to build up.

While there are extensive research works on the joint dynamics of one congestion control strategy with DropTail/RED, little has been done on the interactions between *different* congestion control strategies with *different* queue management schemes. In this paper, we investigate extensively the joint network dynamics with different (α,β) parameters and different queue management schemes (i.e. DropTail vs. RED). We reveal that (i) With DropTail buffer, although smooth TCP causes less queuing-delay jitter, its average queuing delay is significantly higher than that of responsive TCP. The direct implication of this discovery is that when mobile users of media-streaming and short messages share a bottleneck link, the energy consumption for sending short messages can increase severely if media-streaming applications adopt smooth TCPs. (ii) With RED, on the other hand, smooth TCP not only has smaller queue oscillations, the average or max queue length is smaller as well. (iii) Although RED can control the magnitude of queue oscillations, the frequency of queue oscillations can increase, especially with the fast additive increase speed of responsive TCP. (iv) With multiple bottlenecks and different levels of capacity aggregations, the system

fairness is better with responsive TCP when routers use DropTail buffers, or better with smooth TCP when routers use RED.

The rest of the paper is organized as follows. In section 2, we give an intuitive analysis of the dynamics of AIMD control and queue fluctuations. In section 3, the experiment methodology and metrics are given. Section 4 provides detailed results and analysis. Section 5 concludes the paper.

2 The Dynamics of Congestion Control with a Drop Tail Buffer

Previously (see [11]) we extended the network model of [2]. Consider a simple network topology shown above, in which link bandwidths and propagation delays are

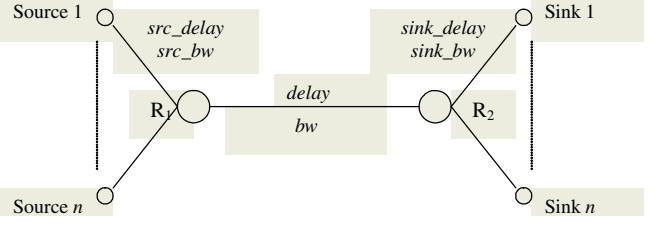


Fig. 1. A simple network topology

labeled. n TCP flows share a bottleneck link with capacity of bw, and the round trip propagation delay is RTT_0. To capture the overall system behavior, we define the aggregated window size at time t as: $cwnd(t) = \sum cwnd_i(t)$, where $cwnd_i(t)$ is the window size of the i^{th} flow. Consequently, the system throughput at time t can be given by the following equation:

$$throughput(t) = \frac{cwnd(t)}{RTT(t)} = \frac{cwnd(t)}{RTT_0 + qdelay(t)} \tag{2}$$

where $qdelay(t)$ is the queuing delay at the bottleneck router R_1.

Consider the time period when all flows are in the additive increase stage. If $cwnd(t)$ is below the point *knee* [2], where $cwnd_{knee} = RTT_0 \cdot bw$, then there is no steady queue build-up in R_1 (i.e. $RTT(t) = RTT_0$), and according to (2), the throughput grows in proportion to *cwnd*. The bottleneck capacity is not fully utilized until *cwnd* reaches $cwnd_{knee}$.

If *cwnd* increases further beyond $cwnd_{knee}$, however, the bottleneck queue builds up steadily, with a saturated bottleneck link. If

$$cwnd(t) = cwnd_{knee} + \Delta w(t) \quad (\Delta w(t) > 0) \tag{3}$$

then $\Delta w(t)$ packets will linger in the queue. The TCP flows continue to additively expand their window sizes, until the queue length $\Delta w(t)$ reaches the buffer size, i.e. when *cwnd* touches the point *cliff*, where $cwnd_{cliff} = (RTT_0 + max\ qdelay) \cdot bw$. TCP

senders then multiplicatively decrease their congestion windows, after packet losses due to buffer overflow are detected.

The above analysis demonstrates that increasing *cwnd* beyond the knee does not enhance further the system throughput, but only results in increasing queuing delay. Moreover, in order to prevent the system from operating below the knee where bandwidth is underutilized, and meanwhile maintain adequate AIMD oscillations (which affects the speed to converge to fairness [9]), an efficient window decreasing ratio should be

$$\beta = \frac{cwnd_{knee}}{cwnd_{cliff}} = \frac{1}{1+k} \quad where\ k = \frac{BufferSize}{RTT_0 \cdot bw} \tag{4}$$

Furthermore, in [11] we confirmed that in a real system, packet losses may not occur to all flows when the bottleneck buffer overflows, even with drop tail buffers. The selection of which flows to drop is random by nature. With RED, random congestion indications are explicitly performed. Therefore, downward window adjustments are not synchronized among competing flows. We revealed that with unsynchronized multiplicative decreases, the convergence speed to fairness is very slow, if measured by the worst-case fairness [11].

3 Experimental Methodology

We implemented our experiment plan on the ns-2 network simulator. The network topology is shown in Figure 1 in section 2. The propagation delay of access links is 10ms, while the delay of the bottleneck link is 15ms. The capacity of the bottleneck link (*bw*), access links to source/sink nodes (*src_bw/sink_bw*) is 10 Mbps. For simulations of heterogeneous (wired and wireless) networks, ns-2 error models were inserted into the access links to the sink nodes. The Bernoulli model was used to simulate link-level errors with configurable bit error rate (BER). The connection time was 100 seconds.

We selected and evaluated four protocols across a spectrum of TCP-friendly TCP(α, β) protocols, from smooth TCP to responsive TCP: TCP(0.31, 0.875), TCP(0.583, 0.75), TCP(1, 0.5) (standard TCP) and TCP(1.25, 0.25). The size of the

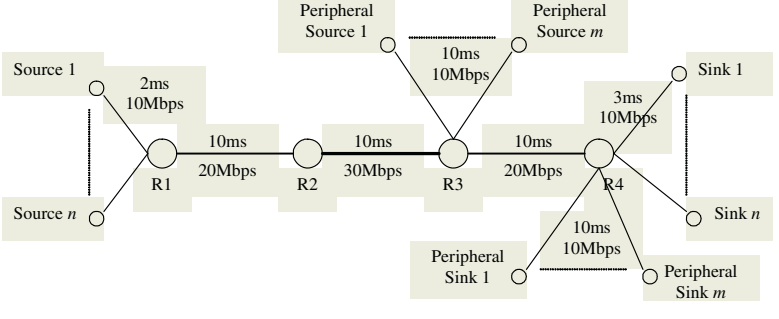


Fig. 2. Network topology with multiple bottlenecks and cross traffic

bottleneck buffer is 100 packets. The settings for RED are as per [5]. Specifically, *max_th = 3·min_th* and *min_th = BufferSiz/6*.

Protocol behaviors were also evaluated with multiple bottlenecks and cross traffic, using the topology in Figure 2. The router R1 is the bottleneck for the main traffic, which includes TCP flows between "source nodes" to "sink nodes". The router R3 is another bottleneck for the competing main traffic and cross traffic, which includes TCP flows between "peripheral source nodes" and "peripheral sink nodes".

The system goodput, defined as the sum of the goodput of all flows, is used to measure the overall system efficiency in terms of bandwidth utilization at the receivers. The queue size of the bottleneck router is traced and sampled every 100ms. Long-term Fairness is measured by the Fairness Index, defined in [2]:

$$FairnessIndex = \frac{\left(\sum_{i=1}^{n} throughput_i\right)^2}{n\sum_{i=1}^{n} throughput_i^2}$$

where $throughput_i$ is the throughput of the i^{th} flow. This Fairness Index provides a sort of "average-case" analysis. In order conduct a "worst-case" analysis and provide a tight bound on fairness, the Worst-Case Fairness is defined as:

$$WorstCaseFairness = \frac{\min_{1 \le i \le n} throughput_i}{\max_{1 \le i \le n} throughput_i}$$

When the system is fair on average but particularly unfair to a very small fraction of flows, the unfairness can only be captured by the worst-case fairness (see [11] for details).

4 Results and Observations

4.1 Queue Length and Its Impact on Energy Consumption of Mobile Devices

We first simulated 10 flows over the simple network topology described in section 3, with a drop-tail buffer. The bottleneck queue lengths over time are depicted in Figures 3-6. As can be seen from the analysis in section 2, the fluctuations of queues reflect the oscillations of sending rates, when the system operates above the knee. The queue fluctuation of the responsive TCP(1.25, 0.25) is so dramatic that sometimes its queue length approaches zero, and the bottleneck link is temporarily underutilized because of the idle queue. On the other hand, although smooth window adjustment leads to smaller jitters in queuing delays, the queuing delay remains high throughout the simulation. Due to the high window-decrease ratio upon congestion, the average queue length of TCP(0.31, 0.875) is much higher than the other protocols. Notably this is also true with another smoothness-oriented TCP-friendly protocol TFRC (see our results in [11]).

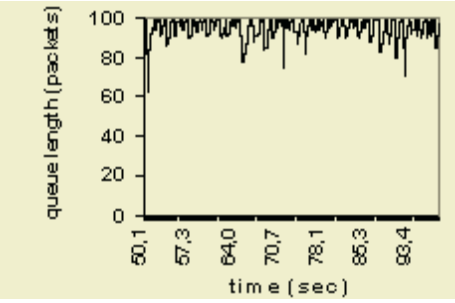


Fig. 3. DropTail queue with TCP (0.31, 0.875)

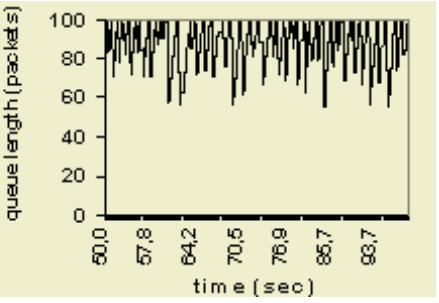


Fig. 4. DropTail queue with TCP (0.583, 0.75)

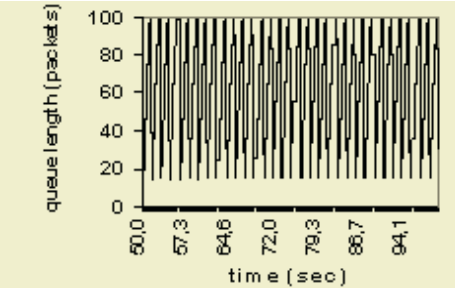


Fig. 5. DropTail queue with TCP (1, 0.5)

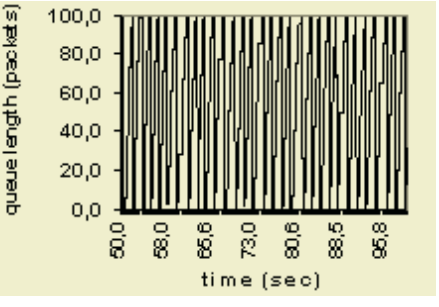


Fig. 6. DropTail queue with TCP (1.25, 0.25)

The queue lengths were also traced with RED configured in routers, shown in Figures 7-10. With smooth TCPs, not only the queue oscillation is smaller, but also the maximum/average queue size is lower. It seems that RED can control the queue growth more effectively with smooth TCP flows. We now give an intuitive analysis why α is the dominant factor in this scenario. Assume the *average* probability that an individual flow experiences packet drops is p_f, which increases with the queue length. The current queue length is determined by the aggregated window size *cwnd(t)*, as shown in section 2. The expected size of the aggregated window in the next RTT will be:

$$\begin{aligned} cwnd(t+RTT) &= \sum_{flow\ j \in MD}(cwnd_j(t)\cdot\beta) + \sum_{flow\ i \in AI}(cwnd_i(t)+\alpha) \\ &= p_f \cdot cwnd(t)\cdot\beta + (1-p_f)\cdot cwnd(t) + (1-p_f)\cdot n\cdot\alpha \\ &= cwnd(t) - p_f\cdot(1-\beta)\cdot cwnd(t) + (1-p_f)\cdot n\cdot\alpha \end{aligned} \tag{5}$$

Intuitively, with a small queue size and hence a small p_f, the α-related third term is the dominant factor for window adjustments. More importantly, as the number of flows increases, the impact of the α-related term increases with n, while the β-related term does not. With the same queue length, the larger n is, the stronger the momentum of queue/window increase. Hence, smooth TCPs with a low α are more "responsive" to the early packet drops by RED.

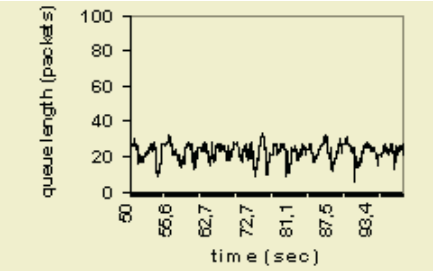


Fig. 7. RED queue with TCP (0.31, 0.875)

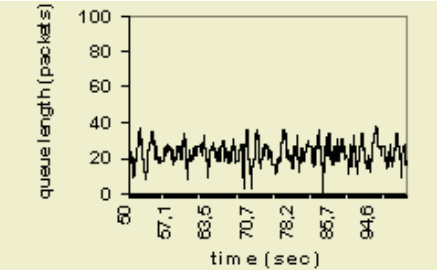


Fig. 8. RED queue with TCP (0.583, 0.75)

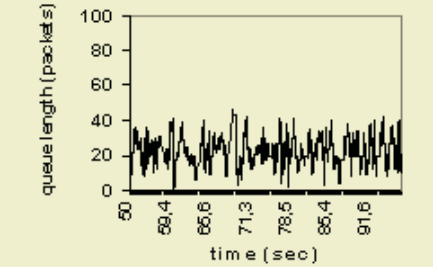


Fig. 9. RED queue length with TCP (1, 0.5)

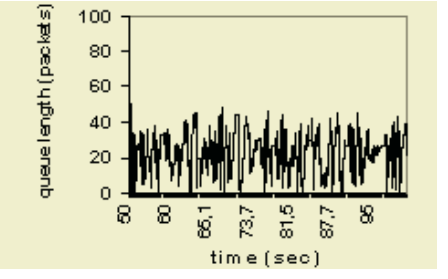


Fig. 10. RED queue with TCP (1.25, 0.25)

After we increase the number of competing flows to 60, the behavior of RED becomes close to DropTail (figures not shown due to the space limit). With a large n, the aggregated increase speed of the system is higher, due to the large $n\alpha$ in equation (5) (even with a small α). The random packet drops by RED between *min_th* and *max_th* cannot effectively reduce the momentum of queue buildup. The average queue frequently touches *max_th*, and RED then drops all packets. The system dynamics is similar to a DropTail buffer, except that the maximum queue is bounded by $max\_th = 0.5 \bullet buffer\_size$.

Implications to Energy Consumptions of Mobile Devices: While the throughput of long flows (e.g. multimedia applications) is determined by the available bandwidth, the connection time of short flows (e.g. short messages of mobile phones) is mainly bounded by RTT. Furthermore, for mobile devices, the extended connection time means higher energy consumptions [7], since they cannot quickly switch back to power-saving mode. Assume that a group of mobile users subscribe to a media-streaming server. Another group of mobile users frequently send/receive short messages (that can fit into one packet) to a short-message server. Media-streaming users use laptops that rely less on batteries, while short messages of mobile phones are more sensitive to energy consumptions. Assume that flows to/from these two servers share a bottleneck link somewhere in the network. Smooth TCP-friendly flows originated from the media-streaming server can cause large persistent queues in the bottleneck buffer, if RED is not deployed, or if RED is deployed but the number

of competing flows is large. Data packets or ACKs from the short-message server to the mobile phones can be significantly delayed. That is, the deployment of smooth TCP for media-streaming can adversely affect the connection time and hence the energy consumption of short messages.

4.2 Network Behaviors over Heterogeneous (Wired/Wireless) Networks

We further repeated the simulations in section 4.1 by inserting random wireless bit errors, with packer error rate 1%. It is interesting to observe that although RED can limit the maximum queue and hence the magnitude of queue fluctuation, it cannot fully control the frequency of queue oscillations (Figures 11-14). Since RED forces TCP senders to adjust before the system reaches the *cliff*, the queue fluctuates more frequently. On the other hand, with smooth TCPs, congestion epochs (the time period between two consecutive multiplicative decreases) are extended. Thus, the growing speed of queue is moderated, and the frequency of queue oscillations is reduced.

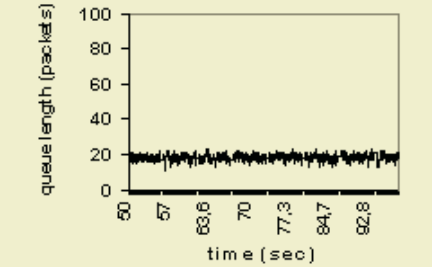


Fig. 11. RED queue Length TCP (0.31, 0.875)

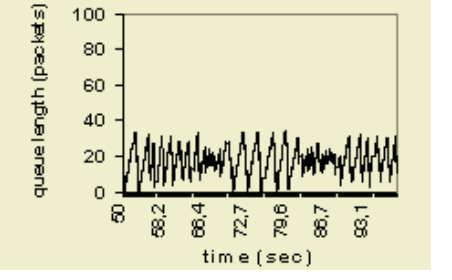


Fig 12. RED queue with TCP (0.583, 0.75)

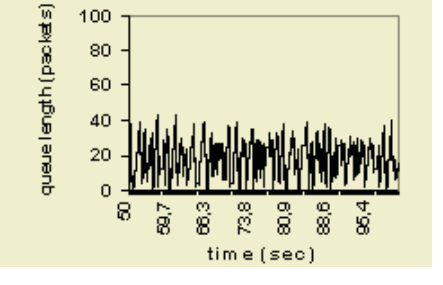


Fig. 13. RED queue with TCP (1, 0.5)

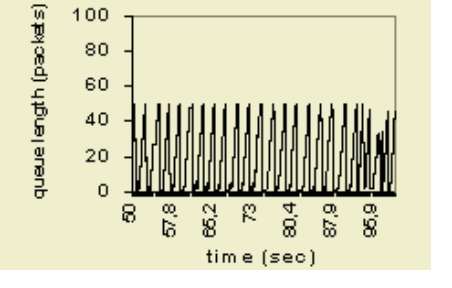


Fig. 14. RED queue with TCP (1.25, 0.25)

4.3 Traffic over Complex Topology with Multiple Bottlenecks

We also tested the system behaviors with different TCP(α, β) protocols over complex topologies, shown in Figure 2 in section 3. Half of the flows form the main traffic, while the other half form the cross traffic. We choose TCP(0.31, 0.875) to represent

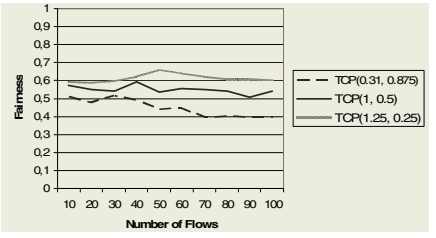


Fig. 15. DroTail Fairness

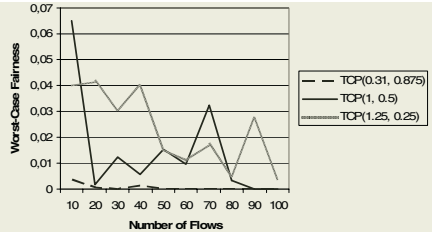


Fig. 16. DroTail worst-case fairness

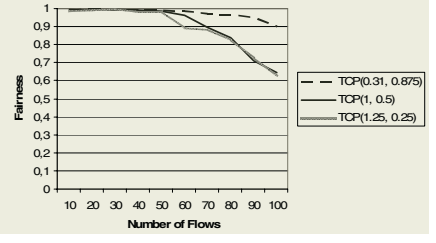


Fig. 17. RED Fairness

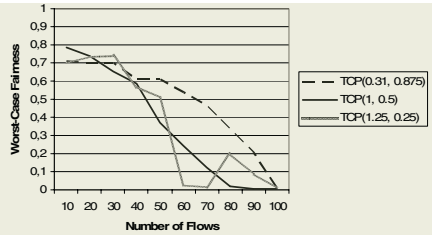


Fig. 18. RED Worst-case Fairness

smooth TCP, and TCP(1.25, 0.25) to represent responsive TCP. We have already shown in [8] that with drop tail buffers and standard TCP, the main traffic consumes more bandwidth than the cross traffic, due to the fact that packets of main-traffic flows are aggregated before entering R2. They are more uniformly distributed in the time domain, therefore having a smaller probability to get dropped, compared to the burstiness of non-aggregated cross-traffic flows. With RED gateways, better system fairness is achieved. In this paper, we further study the system behaviors with different window adjustment strategies and queue management schemes. With DropTail, responsive TCP's fairness (Figure 15-16), especially the worst-case fairness, is much higher than smooth TCPs. Notably the lowest goodput (Figure 16) of individual flow is less then 10% of the highest one. Upon packet losses, responsive TCP flows in the main traffic adjust downwards more dramatically, leaving sufficient space for flows in the cross traffic to grow. That is, with responsive AIMD window adjustments, the system is less likely to be biased against less aggregated flows. With RED turned on in the routers, the system fairness (Figures 17-18) is significantly improved, because RED's early random packet drops discard more packets from flows consuming more bandwidth. However, with large number of competing flows, the fairness is still low. Interestingly, with RED, smooth TCP achieves better fairness than responsive TCP. With smooth TCP, RED can more effectively control the queue growth. With the faster increase speed of responsive TCP, the senders can easily overshoot, and the average queue frequently touches the point max_th, where RED drops all packets and behaves similar to DropTail.

5 Conclusion

We investigated the interaction between different window adjustment strategies, and different queue management schemes in routers. We discussed its impact on the end-to-end delay and fairness, and its implications to the energy consumption of mobile phones.

References

1. M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", RFC2581, April 1999.
2. 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):1-14, 1989.
3. S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", In Proceedings of ACM SIGCOMM 2000, August 2000.
4. S. Floyd, and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, 1(4):397-413, August 1993.
5. S. Floyd, "RED: Discussions of Setting Parameters", November 1997, available from http://www.icir.org/floyd/REDparameters.txt
6. P. Papadimitriou and V. Tsaoussidis, "On Transport Layer Mechanisms for Real-Time QoS", TR-DUTH-EE-2005-10, Feb. 2005
7. C. Jones, K. Sivalingam, P. Agrawal and J. Chen, "A Survey of Energy Efficient Network Protocols for Wireless Networks", ACM Journal on Wireless Networks, vol. 7, No. 4, 2001.
8. V. Tsaoussidis and C. Zhang, "TCP-Real: Receiver-Oriented Congestion Control", Computer Networks Journal (Elsevier), Vol. 40, No. 4, November 2002.
9. V. Tsaoussidis and C. Zhang, "The Dynamics of Responsiveness and Smoothness in Heterogeneous Networks", IEEE Journal on Selected Areas in Communications, March 2005.
10. Y.R. Yang and S.S. Lam, "General AIMD Congestion Control", IEEE ICNP '00, Nov 2000.
11. C. Zhang and V. Tsaoussidis, "Improving TCP Smoothness by Synchronized and Measurement-based Congestion Avoidance", In Proceedings of ACM NOSSDAV 2003, June 2003.

A Novel TCP Congestion Control (TCP-CC) Algorithm for Future Internet Applications and Services

Haiguang Wang and Winston Khoon Guan Seah

Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613
{wanghg, winston}@i2r.a-star.edu.sg

Abstract. In this paper, we present a novel congestion control algorithm for the Transmission Control Protocol (TCP) for the future Internet. Our assumption of future Internet is that, with the increasing quality of service (QoS) requirements, per-flow packet scheduling (per-flow here refers to per TCP or UDP connection) will replace the current first-come-first-serve algorithm used in routers. Based on the assumption, we design a new congestion control algorithm. In our TCP-CC algorithm, each connection adjusts the size of the congestion window according to the size of its packet queue at the bottleneck router. Thus, the queue size for each connection at the bottleneck router is within a controlled range. We show that congestion loss is effectively reduced compared to the current TCP congestion algorithm.

1 Introduction

TCP has been the dominant transport layer protocol of the Internet since 1970s. As a reliable end-to-end data transmission protocol, it is used by many applications such as WWW, FTP, Telnet, Email, etc. It enables the computers on separate networks to communicate and share data with each other. However, with the exponential growth of computer networks, congestion becomes a severe problem [1]. Congestion control algorithms are designed to reduce the packet drops at the bottleneck router.

Historically, TCP's congestion control algorithms are based on the notion that the network is a black box [1] [2]. A TCP sender increases its congestion window size (a value that determines the maximal size of data that can be pumped into the network by the sender) gradually to probe the available network capacity. When packet loss occurs, the size of the congestion window is decreased to relieve the network from congestion [3]. However, using packet loss as a primary indication of congestion may not always be a good scheme. Sometimes, it causes the packet buffer of routers to overflow unnecessarily even if there is only one TCP connection on the network. It is inefficient as the dropped packets have to be retransmitted and bandwidth is wasted. It may also affect the performance of the real-time traffic on the Internet [4]. Therefore, new congestion-control algorithms that avoid using packet loss as primary indication of congestion are required.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 75–85, 2005.

Another motivation for designing a new congestion control algorithm is the change in the nature of Internet applications. The increasing demand for QoS beyond best-effort makes the widely used FCFS packet scheduling algorithm insufficient for the future Internet [5] [6]. It does not allow a router to give some sources a lower delay than others nor prevent malicious sources from eating up excessive bandwidth. Protocols for supporting QoS such as the Resource ReSerVation Protocol (RSVP) [7] require the current FCFS packet scheduling algorithm to be replaced with per-flow ones [8] [9] [10].

In this paper, we propose a new TCP congestion control algorithm**,** which we refer to as **TCP-CC,** for networks that support per-flow packet scheduling. The congestion window of a connection is adjusted dynamically according to the size of its packet queue at the bottleneck router. Thus, queue size for each connection is controlled within a predefined range at the bottleneck router. Compared to the congestion control algorithm used by the current TCP implementations, which increases the congestion window until packet loss occurs, TCP-CC saves the expensive buffer space at the bottleneck router and reduces congestion loss. It is superior to TCP Vegas [11] as it can identify the direction of congestion. While similar to the one used in TCP Santa Cruz (SC) [12], it overcomes the unfairness problem inherent in TCP SC.

The paper is organised as follows. Section 2 discusses the previous work in congestion control for TCP. Section 3 describes our congestion control algorithm. Section 4 shows the results obtained from our test-bed and the performance improvements compared to other algorithms. Finally, Section 5 summaries the paper.

2 Previous Work

Congestion control has been an active research area for TCP since 1986 when the Internet was confronted with the congestion collapse for the first time. Many solutions with [13] [14] [15] or without [1] [11] [12] [16] [18] the involvement of routers have been proposed since then.

For router-based schemes, the router either drops packets when the size of the router buffer reaches a predetermined value (Random Early Detection (RED) gateway) [13] or notifies the sender by setting a flag in the packet (Explicit Congestion Notification (ECN)) [14]. This consequently adjusts the congestion window. The scheme proposed in [15] changes the size of receiver's advertised window such that the packet rate is slowed down when network enters the congestion state. The problem of router-based algorithms is that the router support is needed.

End-to-end congestion control algorithms adjust the packet rate according to some indication from the network. The earliest congestion control algorithm uses packet loss as an indication of congestion [1]. The congestion window is gradually increased until packet loss happens, then the window size is halved. This cycle of increase-drop-decrease period is repeated. The problem of this algorithm has been discussed in Section 1.

An improved congestion control algorithm, TCP Vegas [11], adjusts the congestion window according to the difference between the expected and actual throughput. The

expected throughput is calculated from the minimal round-trip time (RTT) previously observed and the actual throughput is calculated according to the current RTT. When the differences are below α or above β, where α and β represent too little or too much data in flight, the size of congestion window is increased or decreased accordingly. Similar algorithms have been proposed by [16] [18].

The problem of using RTT in congestion control is the traffic fluctuation on return link, which may lead to erroneous decisions in congestion window adjustment. To solve this problem, TCP-SC [12] introduced the concept of relative delay, which is the increase and decrease in delay that packets experience with respect to each other as they propagate through the network. It estimates the number of packets of a connection in the bottleneck router queue using the relative delay and the congestion window is adjusted when too little or too many packets are in the queue. However, in a multiple connection case, bandwidth is shared unfairly among the connections.

The above congestion control algorithms do not consider the features of per-flow packet scheduling. A congestion control algorithm for per-flow packet scheduling has been proposed in [16]. Back-to-back pair packets are sent out and the inter-arrival time of the acknowledgement packets (ACK) are measured to estimate the bandwidth assigned to this connection and the sending rate is adjusted according to the estimated bandwidth. While it has the advantages of per-flow packet scheduling, the problem of this algorithm is that the variation of the ACK's delay on the return link may adversely affect its performance.

3 TCP Congestion Control Algorithm

Our TCP congestion control algorithm is based on the fundamentals of queuing theory and applies the concept of relative delay and Back-to-Back pair packets in queuing-length detection at the bottleneck router.

3.1 Assumptions

The basic assumption of our algorithm is that per-flow packet scheduling algorithm is used in the routers to provide QoS support. Per-flow packet scheduling algorithms have been introduced in [8] [9] [10] and Deficit Round Robin (DRR) [10] is implemented in a product [20]. Our assumption of the future Internet is as follows.

- For a given duration during which congestion occurs, there exists only one bottleneck router on the link between the sender and receiver.
- DRR is used in packet scheduling at the bottleneck router.
- Each TCP connection is treated as a flow.
- When congestion happens, the router drops packets of the flow with longest queue.
- The sender always has data to send.

3.2 The New Algorithm

Similar to TCP-SC [12], the critical point of our TCP-CC algorithm is in calculating the packet-queue-length (PQL) (including the packet that is being served) of a connection at the bottleneck router. According to the Little's theorem:

$$N = \lambda T \tag{1}$$

where N is the average number of customers in a system, λ is the arrival rate, and T is the mean time each customer spends in the system. For a computer network, we can consider the router as the server and the packets are the customers, and if the sender knows the packet arrival rate λ and T, then it can estimate the value of PQL at the bottleneck router. Assuming the packet arrival rate from a connection is λ_{t_j} and the mean time a packet spends at the bottleneck router is T_j when the j^{th} packet reaches the bottleneck router. Then, we can estimate PQL of this connection at the bottleneck router, as follows:

$$N_{t_j} = \lambda_{t_j} * T_j \tag{2}$$

where N_{t_j} is the PQL at time t_j. For a connection, λ_{t_j} can be derived as:

$$\lambda_{t_j} = \frac{\#\ pkts \quad received}{R} \tag{3}$$

where R is the duration between the arrival times of the first and last packets in the period for which λ_{t_j} is being computed.

Therefore, if we can get the value of T_j, then we can know the PQL of this connection. One way is to let the router attach the time in the packet. However, this method may consume expensive computing resource at the router.

The concept of relative delay introduced by TCP-SC can tell us the additional delay experienced by j^{th} packet compare to the i^{th} packet. If we know the delay, T_i, of the i^{th} packet, then we can derive the delay of the j^{th} packet from (4) and (5).

$$T_j = T_i + D^F_{j,i} \tag{4}$$

$$D^F_{j,i} = (R_j - R_i) - (S_j - S_i) \tag{5}$$

where j is greater than i, $D^F_{j,i}$ is the relative delay of the j^{th} packet compare to the i^{th} packet, R_j and R_i are the arrival times of the j^{th} and i^{th} packets at the receiver side, and S_j and S_i are the sending times of the j^{th} and i^{th} packets at the sender side.

The problem lies in determining the value of T_i. Back-to-Back pair packets used in [16] can be used in estimating the value of T_0 as follows. Firstly, after the connection is established, the size of congestion window is set to 2. Thus the Back-to-Back pair packets, namely, packet 0 and 1, are sent out. The bandwidth assigned to this connection at the bottleneck router can then be calculated as follows:

$$b_1 = pkt\_size / R_1 - R_0 \tag{6}$$

where b_1 represent the bandwidth assigned to this connection when packet 1 is served by the bottleneck router. As the time interval between packet 0 and 1 are served is very short, we assume the bandwidth assigned to this connection does not change too much, that is $b_0 \cong b_1$. As packet 0 is the first packet of this connection, no packet is in the queue of this connection at the bottleneck router when it reaches. Thus, T_0 can be estimated as follows.

$$T_0 = \frac{pkt\_size}{b0} \approx \frac{pkt\_size}{b1} = R_1 - R_0 \tag{7}$$

Since R_0 and R_1 are known values, we can get the estimated value of T_0 using (7) and get T_j using (4) and (5), and finally get the value of N_{t_j} through (2).

The congestion window is adjusted according to (8) after each window of data is received. It is similar to the schemes used in TCP Vegas and SC.

$$cwnd = \begin{cases} cwnd + 1 & if\ N_{t_i} < Q_m - \alpha \\ cwnd & if\ Q_m - \alpha < N_{t_i} < Q_m + \beta \\ cwnd - 1 & if\ N_{t_i} > Q_m + \beta \end{cases} \tag{8}$$

cwnd is the size of congestion window. Q_m is the expected value of PQL at the bottleneck router for this connection. Q_m - α and Q_m + β represent the lower and upper bound of the PQL at the bottleneck router.

4 Performance Test

In this section, we examine the performance of our new algorithm, TCP-CC, and compared its performance with TCP Reno [3], Vegas [11] and SC [12]. We first show the performance results for a basic configuration with a single connection case. Then, we show the performance results for multiple-connection case. Finally, we examine the influence of traffic on the reverse link. We have implemented TCP-CC, TCP Vegas and TCP-SC in our test-bed.

4.1 Configuration of Test-Bed

We set up a test-bed as Fig.1 shows. The test-bed consists of 4 computers with Red Hat Linux 7.0 (kernel version: 2.4.10). The computers are connected with 100 Mbps Ethernet links. We have developed a per-flow router emulator with a DRR packet scheduling algorithm. It simulates the bottleneck router with 100 Mbps input link and 1.5 Mbps output link. The size of buffer is set to 22.5 KB (15 data packets) at the router. The last packet in the longest queue will be dropped when congestion happens. Another network emulator, NISTNet [21], is run on the computer labeled as the non-bottleneck router. It is used to simulate the delay at the non-bottleneck router on the Internet. We assume each packet will be delayed 50 ms at the non-bottleneck router. The bandwidth delay product (BDP) is equal to 19.8 KB, or 13.5 data packets. In the test, the receiver fetches a file with a size of 5 MB from sender through a FTP client.

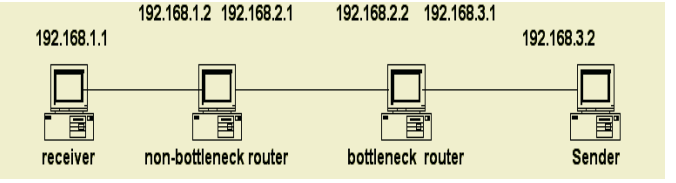


Fig. 1. Configuration of Test-bed

4.2 Performance Results for One Connection Case

Fig.2(a) and (b) shows the growth of the TCP Reno's congestion window and packet queue at the bottleneck router. As TCP Reno uses packet loss as primary congestion indication, it keeps on increasing the size of congestion window until the congestion loss happens. Then, the congestion window is decreased and starts the next round of window increase-drop-decrease period. Even for the one-connection case, packets are dropped periodically, and cause the congestion window and packet queue at bottleneck router to vary in a see-saw oscillation. The delay variance caused by this oscillation may affect the performance of real time and interactive applications [12].

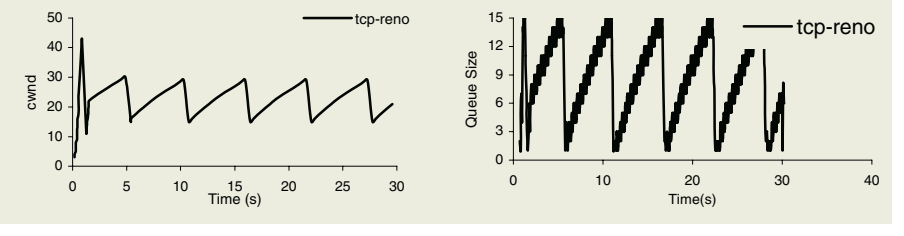


Fig. 2. TCP Reno (1 connection): (a) CWnd {b) Bottleneck Queue

Fig.3(a) and (b) shows the variation of the congestion window and the packet queue at the bottleneck router of our new algorithm. We set the value of α to 1.5 and β to 2.5. The congestion window varies in a very small range from 13 to 15, and the packet queue at bottleneck router varies from 1 to 3 packets. These figures show the main advantage of TCP-CC, that is, the bandwidth of the bottleneck router is almost fully used without causing any network congestion and packet loss, and the queue size at the bottleneck router varies within a controlled range. We have gotten similar results for TCP Vegas and SC in terms of the variation of congestion window and queue size at bottleneck router. Table 1 shows the throughput of TCP Reno, Vegas, SC and TCP-CC. TCP Reno, SC and TCP-CC have similar throughput that is better than TCP Vegas. Although congestion loss happens for Reno, the size of the congestion window is still greater than the value of BDP. For TCP-SC and TCP-CC, after the connection has started for a while, the pipe is kept full. TCP Vegas oscillates at the beginning of the connection and keeps the window size below the value of BDP for a certain duration. Therefore, the throughput is a little lower than that of the others.

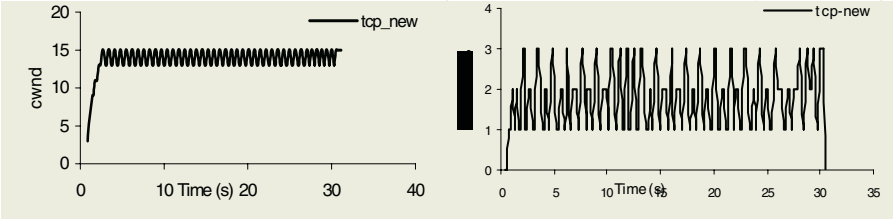


Fig. 3. TCP-CC (1 connection): (a) CWnd (b) Bottleneck Queue

We can also observe from Table 1 that 20 packets have been dropped due to the congestion for TCP Reno. For TCP Vegas, SC and TCP-CC, no packets were dropped since packet loss was not used as primary indication of congestion.

4.3 Performance Results for Multiple Connection Case

In the following, we study the performance of different algorithms in multiple connection cases. In the study, we start multiple connections one after another with an interval of 5 seconds. Each connection downloads 5 M bytes of data from server. We name the connections with their sequence number. For example, we name the first connection as *conn1*, the second as *conn2*, etc.

Table 1. Performance for TCP Reno, SC, Vegas and TCP-CC (1 connection)

Protocol	Throughput	Utilization (%)	Congestion loss
TCP Reno	1.40 Mbps	89.3	20 packets
TCP Vegas (2, 4)	1.36 Mbps	63.3	0
TCP-SC	1.40 Mbps	89.3	0
TCP-CC	1.40 Mbps	89.3	0

Table 2. Performance for 4-connection case

Protocol	Throughput	Utilization (%)	Fairness
TCP Reno	1.43 Mbps	95.3	fair
TCP Vegas (2, 4)	1.43 Mbps	95.3	fair
TCP Santa Cruz	0.88 Mbps	89.3	unfair
TCP-CC	1.43 Mbps	95.3	fair

Table 2 shows the performance results for different algorithms in the four-connection case. The throughputs of TCP Reno, Vegas and TCP-CC are the same as the buffers at the bottleneck router always have packets waiting to be sent. The throughput of TCP-SC is lower than that of the others. The reason is that, in the multiple connection case, the algorithm used by SC for queue size estimation is inaccurate. This further causes the congestion windows to be adjusted inappropriately. For example, as Fig.4(a) shows, for *conn1*, even when the congestion window is

decreased to 2, the estimated queue size is still greater than 2. Thus, the *cwnd* is kept at 2 even when all other connections have completed their download and this causes the network to be underutilized. The reason is that SC computes the current PQL by summing the PQL of the previous window and the PQL introduced by the current window. However, with the change of network status, the old PQL has expired and cannot represent the corresponding value of PQL in the new network status, thus causing the inaccuracy in PQL estimation. Fig.4(b) shows the queue size of *conn1* of TCP SC. It oscillates between 0 and 1 for a while and then drop to 0 as the *cwnd* remained at 2.

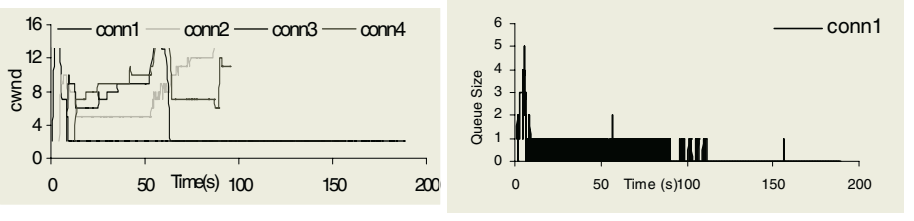


Fig. 4. TCP SC (4 connections): (a) CWnd (b) Queue Size of connection 1

TCP SC also has the fairness problem in multiple connection case. As Fig.4(a) shows, the values of *cwnd* for the connections are difference. Some are as high as 16 while some are as low as 2. The differences cause the bandwidth to be shared unfairly among the connections. Although connection 1 started first, it gets a bandwidth of 0.22 Mbps (1/7 of the total bandwidth) and is the last one to complete the download.

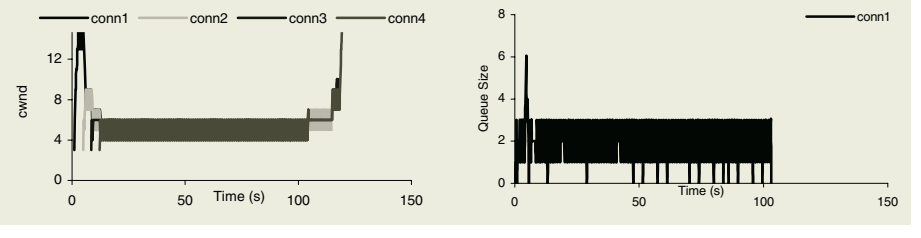


Fig. 5. TCP-CC (4 connections): (a) CWnd (b) Queue Size of connection 1

Compared to TCP-SC, our new algorithm does not have such problems. As Fig.5(a) shows, the values of *cwnd* for the connections are always close to each other. It is adjusted when the network status changes, for example, connections open or close, and tries to keep the PQL between α and β. Fig.5(b) shows the PQL for the *conn1*. Most of time, it varies in the range of 1 to 3, which is close to our objective. Other connections also have similar trends in PQL variation and the bandwidth is shared fairly among different connections.

Table 3 shows the packet drop rate for different algorithms (TCP-SC is not included as it has the fairness problem). TCP Reno drops more packets than other algorithms. Our TCP-CC is the best with no packet dropped in 4-connection case.

Table 3. Congestion Loss in Multiple-connection Case

Protocol	4 connections	6 connections	8 connections
TCP Reno	350 (2.4%)	1092 (5.1%)	1963 (7.0%)
TCP Vegas (2, 4)	0 (0%)	1017 (4.8%)	1916 (6.6%)
TCP-CC	0 (0%)	315 (1.47%)	1286 (4.5%)

When the number of connections increases to 6, the maximum number of packets expected to queue at the bottleneck router is 2.5 * 6 = 15 packets, which is equal to the maximal size of the available buffer at the bottleneck router. Thus, congestion loss happens when some connections do not adjust their window on time. However, the number of packets dropped by TCP-CC is less than one third of the packets dropped by Reno and Vegas. For an 8-connection case, the network is heavily congested. Our algorithm still performs better than Reno and Vegas with a 30% less in packet drop rate. This suggests that TCP-CC can obviously reduce the congestion loss and improve the performance of loss-sensitive applications.

4.4.1 Traffic on Reverse Link

Table 4 shows the performance results of different congestion control algorithms with traffic on the reverse link. The client first starts a connection that fetches data from the server, and subsequently starts another connection which sends data to the server on the reverse link.

Table 4. Throughput Comparison with Traffic on Reverse Link

Protocol	Throughput
TCP Reno	1.36 Mbps
TCP Vegas (2, 4)	1.02 Mbps
TCP-CC	1.33 Mbps

Results in Table 4 shows that the performance of TCP Reno and TCP-CC is better than TCP Vegas with a percentage increase of 33.3% and 30.4% in throughput respectively. The reason is because the ACK delay on reverse link causes TCP Vegas to incorrectly infer the increase of RTT as congestion on the forward link. Thus, *cwnd* is reduced and kept below BDP.

For both TCP Reno and TCP-CC, although there is slight performance degradation due to the delay of the ACK on the reverse link, the degradations are trivial (2.9% and 5%) with TCP Reno performing a little better than TCP-CC (2.3%). However, the cost of the better performance is an over-aggressive *cwnd*-increase algorithm that results in more congestion losses, as shown in Table 3. This violates the objective of congestion control — reducing the congestion loss.

5 Conclusion

In this paper, we have proposed a new congestion-control algorithm (TCP-CC) for the future Internet in which per-flow packet-scheduling algorithm is expected to replace

the FCFS algorithm used by current routers. Compared to other congestion control algorithms, our new algorithm has the following advantages:

- Compared to TCP Reno, TCP-CC adjusts the congestion window based on the queue size of the connection at the bottleneck router instead of the packet loss and this reduces congestion loss significantly.
- Compared to TCP Vegas, TCP-CC achieves better throughput, less congestion loss and is more accurate in identifying the direction of congestion.
- Compared to TCP-SC, TCP-CC has sound fundamentals and solves the fairness problem in multiple connection scenarios.

Our future work is to study TCP-CC in a more complex network environment, which includes the characteristics of web traffic, the variations in packet size etc.

References

[1] V. Jacobson, Congestion avoidance and control, *Proc. SIGCOMM '88*, Stanford, CA, Aug 1988.
[2] V. Jacobson, Modified TCP congestion control avoidance algorithm, *Message to end2end-interest mailing list*, Apr 1990.
[3] W.R. Stevens, *TCP/IP illustrated, volume 1* (Addison-Wesley, 1996).
[4] H. Sawashima, Y. Hori, H. Sunahara and Y. Oie, Characteristics of UDP packet loss: effect of TCP traffic, *Proc. INET'97*, Jun 1997.
[5] S. Keshav and R. Sharma, Issues and trends in router design, *IEEE Communications Magazine*, May 1998, Vol.36, No.5, pp. 144-5.
[6] S. Shenker, Fundamental design issues for the Future Internet, *IEEE Journal of Selected Areas in Communication*, Vol. 13, No. 7, pp. 1176-1188, Sep 1995.
[7] L. Zhang, S. Berson, S. Herzog and S. Jamin, Resource reservation protocol, *RFC2205*, Sep 1997.
[8] A. Demers, S. Keshav and S. Shenker, Analysis and simulation of a fair queuing algorithm, *Proc. ACM SIGCOMM'89*, Vol. 19, pp. 1-12, Oct 1989.
[9] J. Nagle, On packet switches with infinite storage, *IEEE Transaction on Communications*, Vol. 35, Apr 1987.
[10] M. Shreedhar and George Varghese, Efficient fair queuing using deficit round-robin, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 3, Jun 1996.
[11] L.S. Brakmo, S.W. O'Malley and L.L. Peterson, TCP Vegas: new techniques for congestion detection and avoidance, *Proc. ACM SIGCOMM'94*, pp. 24-25, Oct 1994.
[12] C. Parsa and L. Aceves, Improving TCP congestion control over Internets with heterogeneous transmission Media, *Proc. IEEE Conference on Network Protocols (ICNP '99),* Toronto, 1999.
[13] S. Flyod and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, Vol. 4, pp. 397-413, Aug 1993.
[14] V. Jacobson and S. Floyd, TCP and explicit congestion notification, *Computer Communication Review*, Vol. 24, pp. 8-23, Oct 1994.
[15] L. Kalampoukas, A. Varma and K. Ramakrihnan, Explicit window adaptation: a method to enhance TCP performance, *Proc. IEEE INFOCOM'98*, pp. 242-251, Apr 1998.
[16] S. Keshav, Congestion Control in Computer Networks *PhD Thesis, published as UC Berkeley TR-654* , September 1991.

[17] Z. Wang and J. Crowcroft, Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm, *Computer Communication Review*, Vol. 22, pp. 9-16, Apr 1992.

[18] Z. Wang and J. Crowcroft, A new congestion control scheme: slow start and search (Tri-S)", *Computer Communication Review*, Vol. 21, pp. 32-43, Jan 1991.

[19] G. Hasegawa and M. Murata, Survey on fairness issues in TCP congestion control mechanism, *IEICE Transactions on Communications*, E84-B6:1461-1472, Jun 2001.

[20] Cisco 12016 Gigabit Switch Router, available from http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/12016.

[21] NISTNet network emulator, available from http://snad.ncsl.nist.gov/itg/nistnet/.

Performance Evaluation of τ-AIMD over Wireless Asynchronous Networks

Adrian Lahanas[1] and Vassilis Tsaoussidis[2]

[1] Dept. of Computer Science,
University of Cyprus, Nicosia, Cyprus
[2] Dept. of Electrical Engineering,
Demokritos University, Xanthi, Greece

Abstract. The work describes the performance of two congestion control algorithms: AIMD and τ-AIMD. The first is the default mechanism of TCP; the second is a proposed congestion control algorithm that improves fairness of AIMD. We consider asynchronous networks where TCP flows have different propagation delays, a portion of their link is wireless, and they compete for resources over a single bottleneck link. We show that τ-AIMD improves the performance of flows that have long propagation delay and the fairness of the network. In wireless links τ-AIMD outperforms AIMD and the cost of lost packets (or wasted energy) is the same as that of AIMD.

1 Introduction

Additive Increase/Multiplicative Decrease (AIMD) is the algorithm that controls congestion in the Internet [6, 11]. It is coded into TCP and adjusts its sending rate mechanically, according to the 'signals' TCP gets from the network. A lost packet from the data it pumps into the network is considered as a congestion event and therefore, AIMD decreases the sending rate of TCP. Otherwise, when data is transmitted successfully, the AIMD increases the sending rate of TCP by a packet per RTT (round trip time). It is proved that these adjustments bring the network load into an equilibrium and TCP flows converge into a fair state.

Fairness of TCP is an over-argued topic because of the nuances of fairness [4, 12]. In this work we support the *max-min* notion of fairness where all flows get the same treatment (consume the same resources) from the networks. Max-min fairness is discussed in detail in [4]. Judged from the max-min notion of fairness, in asynchronous networks, TCP does not share network resources equally among flows [6, 9]. The problem is caused by short RTT flows which increase their sending rate faster than long RTT ones. Short RTT flows consume more resources from the network, leaving thus less resources for the long RTT flows.

In wireless environments where link disconnections or bit errors are more frequent than in wired networks, the problem of fairness becomes more apparent. In these environments flows experience more packet drops, which they consider as congestion signals, and reduce their sending rate more frequently. Long RTT

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 86–96, 2005.

flows are those that experience the most severe treatment from the network. If we add the wireless link error factor to the limited resources the short RTT flows leave in the system, it is obvious that these flows operate at very low transmission rates and their process of recovery is very time-consuming.

Although fairness of TCP over heterogeneous (wired and wireless) asynchronous networks is an open issue, it has not received enough attention. A solution or an improvement of the problem is important for two reasons: *i*) it will speed-up the recovery process of long RTT flows when multiple errors occur in the network[1], *ii*) when long RTT flows recover faster and have rates close to short RTT flows, they utilise better the network resources or the available bandwidth at the wired and at the wireless portion of the network.

Performance of TCP over wireless links has been studied in a range of works (e.g. [19, 7, 8]) and several mechanisms have been proposed to improve performance over such links. However, the focus of these works has been the wireless hop of the network rather than the global system: flows with heterogeneous RTTs that traverse wired or wireless links. In this work we experiment with a congestion control algorithm, named τ-AIMD, whose purpose is to improve fairness of TCP flows in asynchronous networks. This algorithm derives from AIMD but the rate of the protocol is increased proportionally to its RTT (whereas in TCP it increases a packet per RTT). The long RTT τ-AIMD flows increase their rate faster than they do with AIMD and consume more resources. This scheme however, does not achieve max-min fairness among flows. The experiments in this work are focused mainly on the fairness achieved by AIMD and τ-AIMD in asynchronous networks where a portion of the link is wireless.

The rest of this work is organised as follows: Section 2 describes the related work on fairness and Section 3 gives detailed description of AIMD and τ-AIMD mechanism; Section 4 describes the experimental set-up and Section 5 evaluates the experimental results. Section 6 summarises the work.

2 Related Work

The AIMD was first introduced in [6] where the authors studied the algorithm in terms of fairness and convergence to equilibrium. Jacobson in his work [11] proposed and coded into TCP a mechanism similar to AIMD and studied it in terms of congestion control. The AIMD algorithm is effective in congestion control but, as its authors have stated out, in asynchronous networks the flows do not get the same share [6]. The concept of AIMD itself is quite simple, general and very effective. Although the same concept can be used in different ways to improve a specific performance characteristic of the network (e.g. [9, 13, 14, 2]), the AIMD of TCP is limited into achieving only equilibrium in the network.

[1] Multiple errors might result in sequential decreases of the transmission rate of the flow, which implies a few packet in transit. When in-transit packets are scarce a loss is very costly because it could result in time-outs or might make useless the 3 DACK mechanism of TCP

The proposal in [9] considers the fairness problem of TCP in asynchronous networks. Increasing the sending rate of TCP by $a \cdot r^2$ (where a is a constant and r is the RTT of the flow) can achieve the same rate among the flows and improve fairness. This mechanism is based on AIMD principles but is shown experimentally that the overhead (or the number of lost packets) increases significantly [10]. Choosing the parameters of this mechanism (i.e. constant a) is still an open problem. Analysis of congestion avoidance of TCP in asynchronous systems has shown that when it reaches an equilibrium, the formula

$$F_A^h = \sum_{i \in S} \frac{1}{\tau_i} \log \frac{x_i}{a_I + b_D x_i} \tag{1}$$

where x_i are the rates of each flow and τ_i is the RTT of flow i, is maximized [18]. This is also the fairness characteristic of AIMD in asynchronous systems.

The effect of wireless errors on the throughput performance of TCP has been studied and reported in a series of works (e.g. [20]). The work in [17] reports the effect of the wireless errors in the fairness of TCP. The work in [16] reports the fairness of TCP over 802.11 wireless networks. All these works deal with the performance of TCP over wireless local area networks which is a mere part of a large scale asynchronous network (like Internet). In [5] and [3] is studied the performance of TCP over wireless wide area networks. The improvements that the authors propose deal mainly with the throughput of TCP and are limited at the wireless hop of the network.

3 AIMD and τ-AIMD

AIMD is a distributed algorithm that runs at the transport layer of each end-node with a minimum assistance from the network. The main goal of this algorithm is to control the sending rate of each end-node such that an equilibrium can be reached in the network. Accordingly, each user increases its sending rate linearly when resources are available and decreases exponentially the sending rate as soon as the network is congested. The increase and decrease of the sending rate in TCP is controlled by a parameter called congestion window. This parameter records how many bytes TCP has in transit (i.e. in the network). When the whole window is transmitted successfully TCP increases its window by one packet (i.e. one packet per RTT) and when a packet is lost from the window it decreases to half the congestion window. In addition to AIMD TCP has other mechanisms that assist in congestion avoidance process (e.g. ACK clocking and time-outs) or mechanisms that assist in the recovery process after packets loss (e.g. three DACK, Fast Retransmit, Slow Start, Fast Recovery).

Algorithmically the AIMD can be expressed with the following lines:

AIMD()

1. a_i : *constant* = packet-size()
2. W : *integer* // congestion window
3. *repeat forever*
4. send W bytes in the network
5. receive ACKs
6. *if* W bytes are ACKed
7. $W \leftarrow W + a_i$
8. *else*
9. $W \leftarrow \frac{W}{2}$
10. *end*

END-AIMD

This continuous process of AIMD achieves equilibrium but, does not achieve max-min fairness in asynchronous networks.

τ-AIMD is designed for asynchronous networks: flows might have different RTT or propagation delays (PD). τ-AIMD controls congestion by using the same increase/decrease mechanism of AIMD. To improve fairness τ-AIMD increases the rate of the flows proportionally to the RTT or PD of the flow. Instead of adding a packet every RTT to the window (a_i in the pseudo-code), τ-AIMD adds as many as is the flow's RTT (i.e. $\tau \cdot a_i$, where τ is the RTT of the flow). Algorithmically, τ-AIMD can be expressed by the following lines:

τ-AIMD()

1. a_i : *constant* = packet-size()
2. W : *integer* // congestion window
3. *repeat forever*
4. send W bytes in the network
5. receive ACKs
6. $\tau \leftarrow$ Window-transmission-time()
7. *if* W bytes are ACKed
8. $W \leftarrow W + \tau \cdot a_i$
9. *else*
10. $W \leftarrow \frac{W}{2}$
11. *end*

END-τ-AIMD

It is proved in [14] that an asynchronous system where flows use the τ-AIMD as their congestion control algorithm reaches an equilibrium and a window based fairness (i.e. windows of the flows become equal when they converge to their fair state). Window based fairness does not mean that flows achieve max-min fairness. Nevertheless, window based fairness is closer to max-min fairness than F_A^h fairness of AIMD is.

4 Experimental Methodology

We have implemented τ-AIMD into TCP and validated its performance on Ns2 [21] simulator. We use the smoothed RTT (*srtt* variable of Ns2's TCP) to assign

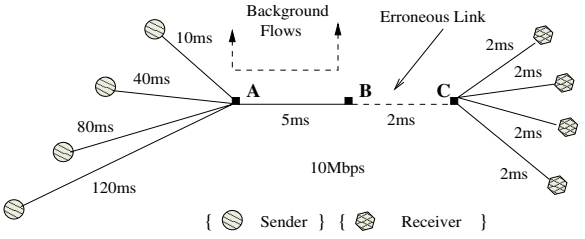


Fig. 1. Single Bottleneck Topology

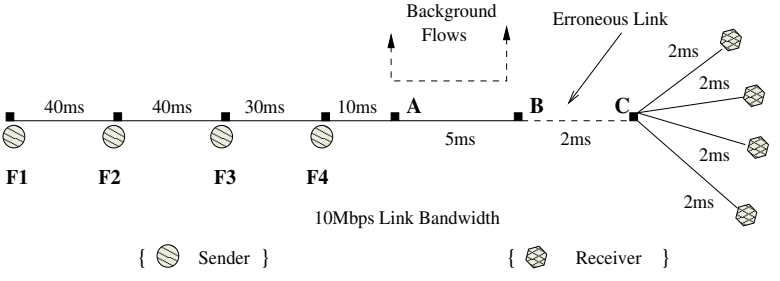


Fig. 2. Linear Topology

values to the τ parameter. Two different sets of experiments are used to validate the performance of the algorithm. In the first set of experiments we set-up a topology with a single bottleneck router A as shown in figure 1. Flows have different propagation delays ranging from 19ms to 129ms and a portion of their link is wireless (from point B to C in each topology). To simulate the wireless link errors we use the On/Off error model presented in [1]. This is a Markov chain model. The packet error rates (PER) in each experiment ranged from 0.01% to 10%.

In the second set of experiments we used a 'linear topology' as that in figure 2. Flows have different PDs and traverse a number of bottleneck routers depending on their PD (e.g. flow F1 traverses 4 bottleneck routers, flow F2 traverses 3, flow F3 traverses 2, and flow F4 traverses 1). In these experiments we were interested in the performance of each algorithm when they traverse a number of bottlenecks and a portion of their link is wireless.

There are four TCP flows in each experiment (four senders and 4 receivers) whose performance we monitored. In addition to these flows we added other background flows at points A and B of each topology that generate forward and backward traffic in order to avoid window and ACK synchronisation. We don't measure the performance of these flows.

The experiments are limited to four flows in order to allow each TCP to expand its window and control the rate mainly with AIMD or τ-AIMD rather than with other mechanisms of TCP (i.e. time-outs and Slow Start). The bottleneck routers use the RED drop policy and are configured with the default Ns2 parameters.

FTP application was attached to each TCP and their task was to send continuously data for a period of 60 seconds - enough to validate the performance of each mechanism. We use *goodput* as a metric to compare the performance of each algorithm. Goodput is defined as the amount of data received at the application over the transmission time. For each PER we conducted 35 experiments and report the average goodput achieved by each flow.

5 Simulation Results

We evaluate first the performance of both algorithms in a network where the only cause of packet loss is congestion and then compare their performance in the presence of errors similar to wireless environments. In figure 3 is displayed the goodput performance of AIMD and τ-AIMD in the single bottleneck topology. The figure shows that when the propagation delay of the flows increases, its goodput performance decreases. The goodput of τ-AIMD flows is higher than the goodput of AIMD when the propagation delay increases whereas, when flows have small propagation delay, the goodput of τ-AIMD is slightly lower. The reason is that long RTT τ-AIMD flows open their window faster than AIMD flows. Therefore, they grab more resources from the network and leave less for the short RTT flows. If the network has less resources the performance of the flows will be lower. In other words, the goodput loss from short RTT flows is converted in goodput gain from long RTT flows. Nevertheless, the fairness performance of the system improves. The figure shows that the goodput of the flow that has delay 80ms from the bottleneck router improves by 16% and the goodput of the flow that has delay 120% ms from the bottleneck improves by 24%.

Figure 4 shows the performance of the algorithms in the linear topology with four bottlenecks and in the absence of wireless errors. The performance of AIMD is similar to its performance in the single bottleneck topology and the gain of long

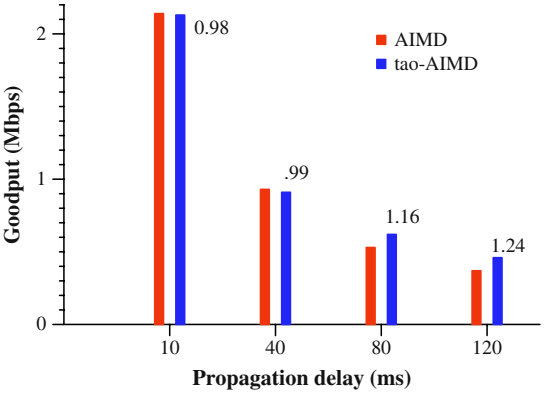


Fig. 3. Goodput of TCP-SACK with AIMD and τ-AIMD congestion control algorithms. The numbers on top of each bar indicate the ratio of τ-AIMD over AIMD goodput

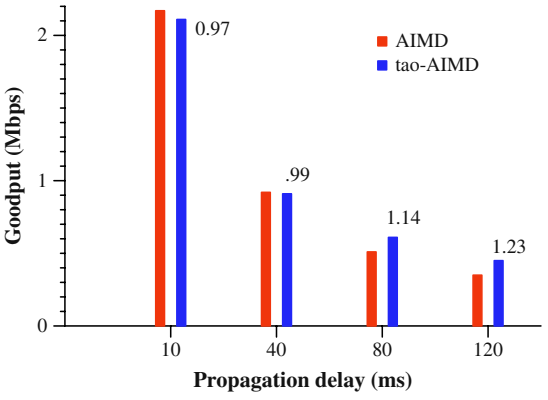


Fig. 4. Goodput of TCP-SACK with AIMD and τ-AIMD in the four bottleneck topology. The numbers on top of each bar indicate the ratio of τ-AIMD over AIMD goodput

RTT τ-AIMD flows over AIMD flows is almost the same. These experiments show the capability of τ-AIMD in improving the fairness and achieving congestion control in asynchronous networks.

In figure 5 is plotted the goodput performance of flows that have delay 10ms and 40ms respectively, from the bottleneck router. The performance is given as a function of the error rate in the wireless channel. When the error rate increases above 0.1% the performance of the flows drops significantly. However, there is not a clear distinction in the performance of each algorithm. The τ-AIMD performs the same as AIMD for short RTT flows because of its implementation in Ns2. The value of 'τ' in the increase formula is calculated by dividing the smoothed RTT variable of TCP by 64, which is almost or bigger than the RTT of the plotted flows[2]. Therefore, τ-AIMD increases its window exactly as AIMD and has the same performance.

When the propagation delay increases the τ-AIMD increases the goodput of long RTT flows. Figures 6 and 7 show that even in the presence of errors characteristic to wireless links, the goodput of τ-AIMD is higher that that of AIMD. This means that the increase mechanism of τ-AIMD benefits also the recovery process of flows after a wireless error event. Expanding faster the window makes the flow exploit better the available bandwidth. When the packet error rates increases up 10% the goodput performance of AIMD and τ-AIMD becomes almost equal. In such error rates the flows probably experience loss of the whole window and result in time-outs or in minimal window values. As a result, the traffic in such high error rates is shaped by time-outs and Slow Start mechanisms rather than AIMD and τ-AIMD.

In the four bottleneck linear topology the performance of AIMD and τ-AIMD is similar to the performance in the single bottleneck topology. Because of the

[2] When the division is equal to zero, we set τ to 1. The source code of the algorithm is available on line at http://www.cs.ucy.ac.cy/~ladrian/software.html.

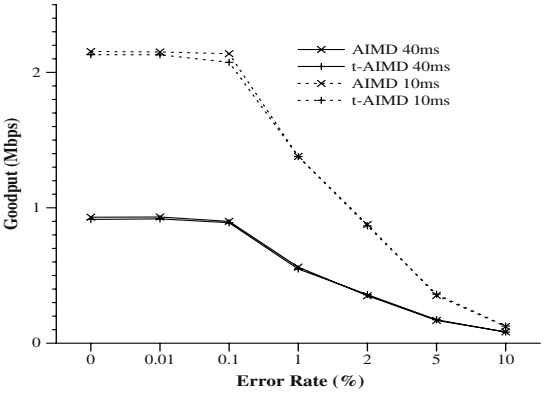


Fig. 5. Goodput of flows when the propagation delay is 10ms vs. goodput when propagation delay is 40ms. Goodput is a function of 'wireless' error rate

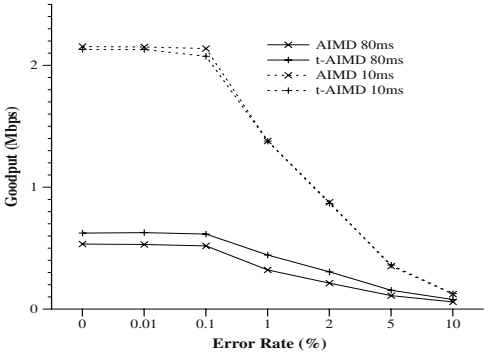


Fig. 6. Goodput of flows when the propagation delay is 10ms vs. goodput when propagation delay is 80ms

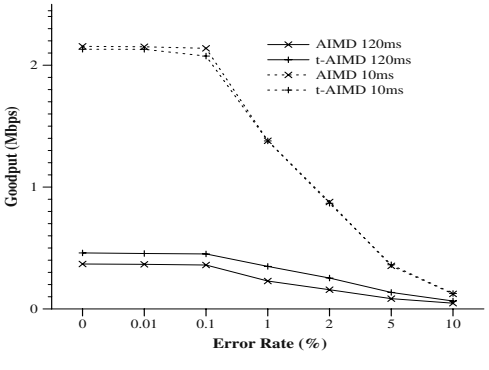


Fig. 7. Goodput of flows when the propagation delay is 10ms vs. goodput when propagation delay is 120ms

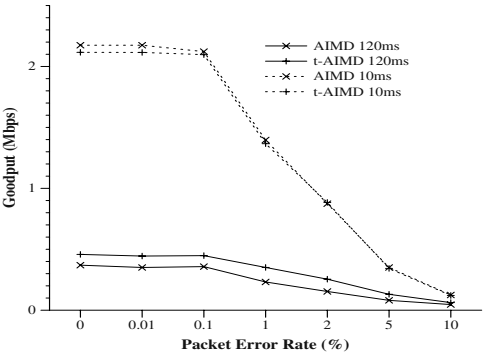


Fig. 8. Goodput of flows when the propagation delay is 10ms vs. goodput when propagation delay is 120ms, in the four bottleneck topology

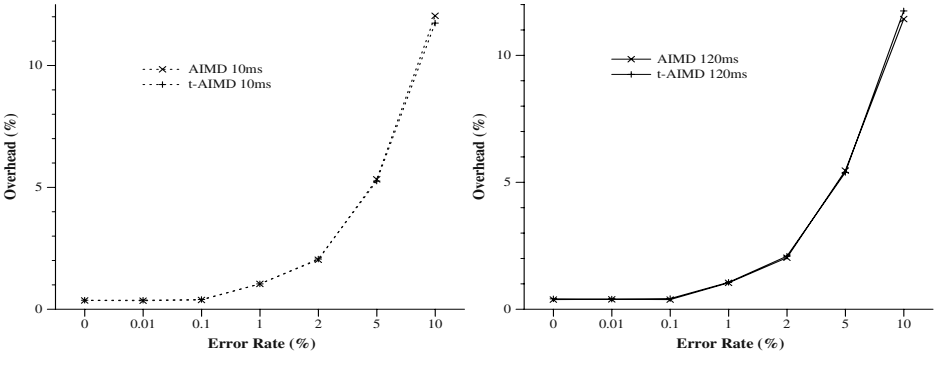


Fig. 9. Left: Overhead of flows when propagation delay is 10ms. Right: Overhead of flows when propagation delay is 120ms

similarity we display only the results with the experiments where flows have delay 10ms and 120ms respectively, from the bottleneck router A. These results are shown in figure 8.

In figure 9 is displayed the *overhead* or the ratio of loosed packets over the received data at the application. It is obvious that the loss increases when the error rate at the wireless portion of the link increases. The experiments show that the overhead of both mechanisms is approximately the same. In erroneous wireless networks the overhead is mainly due to the loss at the wireless portion of the network rather than loss at the bottleneck congested routers. For both protocols the overhead increases almost linearly with the error rate. Since the overhead of both protocols is almost equal we report the overhead of flows that have delay 10ms and 120ms from the bottleneck router A. In the four bottleneck topology these numbers are similar.

An important conclusion that we infer from these experiments is the gain of τ-AIMD flows over the AIMD ones. In asynchronous (wired or wireless) networks

the goodput of τ-AIMD flows is higher than the goodput of AIMD flows. Furthermore, the overhead or lost energy of each protocol is the same. This means that τ-AIMD can achieve higher goodput than AIMD at the same wasted energy cost. This feature of τ-AIMD is important for wireless devices where energy is limited and transmission time effect their battery life.

6 Conclusions

The work studies experimentally the performance of τ-AIMD and AIMD over asynchronous networks. We have shown the potential of τ-AIMD to improve fairness of the network even when a portion of it is wireless. The goodput of long RTT flows is increased whereas the goodput of short RTT flows reduces by the same portion. Although the goodput of long RTT flows increases, the cost of energy loss (or packet loss) is the same as the cost with AIMD. This makes τ-AIMD a protocol suitable for wireless devices where energy is limited and its loss effects the battery time of the device. τ-AIMD flows transmit an amount of data faster than AIMD and at a lower energy cost.

References

1. A. Abouzeid, S. Roy, and M. Azizoglou. Stochastic Modeling of TCP over Lossy Links. *INFOCOM 2000*, March 2000.
2. P. Attie, A. Lahanas, and V. Tsaoussidis. Beyond AIMD: Explicit Fair-share Calculation. In *Proceedings of the ISCC'03*, June 2003.
3. H. Balakrishnan, V. Padmanabhan, and R. Katz. The Effects of Asymmetry in TCP Performance. In *Proceedings of the 3rd ACM/IEEE Mobicom Conference*, pages 77–89, September 1997.
4. D. Bertsekas and R. Gallager. *Data Networks*, chapter 6, pages 493–530. Prentice Hall, 1987.
5. R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt. Flow Aggregation for Enhanced TCP over Wide-Area Wireless. In *Proceedings of the INFOCOM '03*, March 2003.
6. D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1):1–14, June 1989.
7. A. Chockalingam, M. Zorzi, and R. Rao. Performance of TCP on Wireless Fading Links with Memory. In *Proceedings of the IEEE ICC'98, Atlanta, GA*, pages 201–206, June 1998.
8. V. Tsaoussidis *et al.* Energy / Throughput Tradeoffs of TCP Error Control Strategies. In *Proceedings of the 5th IEEE Symposium on Computers and Communications, ISCC*, pages 150–156, July 2000.
9. S. Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks Part1: One-way Traffic. *ACM Computer Communication Review*, 21(5):30–47, October 1991.
10. T. H. Henderson, E. Sahouria, S. McCanne, and R. H. Katz. On Improving the Fairness of TCP Congestion Avoidance. In *Proceedings of the IEEE Globecom*, 1998.

11. V. Jacobson. Congestion Avoidance and Control. In *Proceedings of the ACM SIGCOMM '88*, pages 314–329, August 1988.
12. F. Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.
13. A. Lahanas and V. Tsaoussidis. Exploiting the Efficiency and Fairness Potential of AIMD-based Congestion Avoidance and Control. *Journal of Computer Networks, COMNET*, 43:227–245, 2003.
14. A. Lahanas and V. Tsaoussidis. τ-AIMD for Asynchronous Networks. In *Proceedings of the ISCC'03, Antalya, Turkey*, June 2003.
15. T. Lakshman and U. Madhow. The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. *IEEE/ACM Transactions on Networking*, 5:336–350, June 1997.
16. S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha. Understanding TCP Fairness over Wireless LAN. In *Proceedings of the INFOCOM '03*, March 2003.
17. D. Vardalis and V. Tsaoussidis. On the Efficiency and Fairness of Congestion Control Mechanisms in Wired and Wireless Networks. *The Journal of Supercomputing, Kluwer Academic Publishers*, 23, November 2002.
18. M. Vojnovic, J.Y. Le Boudec, and C. Boutremans. Global Fairness of Additive-Increase and Multiplicative-Decrease with Heterogeneous Round-Trip Times. In *Proceedings of the IEEE INFOCOM'00*, pages 1303–1312, March 2000.
19. G. Xylomenos and G. Polyzos. TCP and UDP Performance over a Wireless LAN. In *Proceedings of the IEEE INFOCOM*, pages 439–446, March 1999.
20. M. Zorzi and R. Rao. Perspective on the Impact of Error Statistics on Protocols for Wireless Networks. *IEEE Personal Communications Magazine*, 6:34–39, October 1999.
21. "—" The Network Simulator - NS-2. Technical report, Web Page: http://www.isi.edu/nsnam/ns/. Version 2.27, January 2004.

Rate Allocation and Buffer Management for Proportional Service Differentiation in Location-Aided Ad Hoc Networks

Sivapathalingham Sivavakeesar and George Pavlou

Centre for Communication Systems Research,
University of Surrey,
Guildford, Surrey GU2 7XH
{S.Sivavakeesar, G.Pavlou}@surrey.ac.uk

Abstract. Given that applications and services for evolving mobile ad hoc networks (MANETs) have diverse quality of service requirements in a similar fashion to fixed networks, this paper proposes a proportional service differentiation (PSD) model. This model is highly scalable and simple to adopt in MANETs because it does not require explicit admission control or maintenance of state information in any intermediate node. It relies instead on localized scheduling and buffer management to achieve a desired global objective. Motivated by this aspect of the PSD model, we propose to combine it with a location-based forwarding strategy as a way to facilitate cross-layer optimization. This association is performed with a view to improve end-to-end service differentiation, although no other explicit mechanisms are used to achieve end-to-end guarantees. This model takes also into consideration the time-varying nature of available bandwidth in MANETs, and tries to calculate it dynamically. Simulation results confirm the per-hop performance improvement.

1 Introduction

The emergence of diversified multimedia applications requires that mobile ad hoc networks (MANETs) are augmented with service differentiation mechanisms, so that certain applications/services and users can possibly benefit from better quality of service (QoS) [2][3][5]. Service differentiation enables categorization of traffic into a set of classes to which network nodes provide priority-based treatment. Although service differentiation can be absolute or relative in nature [16], relative differentiation is preferred in ad hoc networks given that random node mobility, bandwidth and energy constrained operation and the unpredictable behavior of radio channels require a cost-effective solution [5][6]. In addition, absolute differentiation requires sophisticated admission control and resource reservation mechanisms, which are difficult to achieve in highly dynamic ad hoc networks. A relative differentiation mechanism that supports a small number of service classes is simpler in terms of implementation, deployment and manageability [7]. However, relative service differentiation (RSD) can only provide weak guarantees, which do not

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 97–106, 2005.

always address the requirements of applications [8]. In the case of RSD, there is no absolute guarantee that high-priority classes will perform better than lower-priority ones, and this varies depending on the load of each class. In order to tackle this problem, recent research studies have aimed to strengthen the service assurance provided by RSD without incurring much additional complexity. The proportional service differentiation (PSD) model is the result of such studies, and defines a service model with no admission control or an explicit resource reservation mechanism [7][8]. It supports a certain number of service classes relatively ordered in terms of loss rate or average queuing delay and the ratios of those QoS metrics between successive priority classes remain roughly constant, irrespective of network load [7], [8], [9], [10], [11], [12]. The PSD model exhibits controllability and predictability, which make it different from RSD [7].

Service differentiation at MAC-level in terms of prioritized access to the common wireless medium using the DCF-based operation of IEEE 802.11 a/b has been proposed in the literature [3], [4]. Although our work does not address such MAC-level differentiation, it considers the fact that the wireless channel in MANETs is a shared-access medium in which the available bandwidth varies with the number of hosts contending for access [15]. Hence, our work adopts an approach of determining the effective bandwidth dynamically, and this is important for arriving at a service rate allocation among the different service classes. At the network level, we consider only per-hop service differentiation with the understanding that the adoption of such localized behavior at each node will result in end-to-end service differentiation. The latter is facilitated by our forwarder-node selection strategy that identifies and routes packets along non-overloaded mobile nodes (MNs). With the adoption of effective bandwidth calculation mechanism and our forwarder-node selection algorithm, the proposed model attempts to alleviate two major challenges in mobile ad hoc networks; i) fluctuating bandwidth at each node, and ii) topology changes due to mobility.

The rest of this paper is organized as follows. After providing the description of the problem statement in the next subsection, previous work on MANET quality of service and proportional service differentiation is reviewed in section 2. Section 3 presents our model and section 4 evaluates it through simulations. Finally, section 5 concludes the paper and points to future work.

1.1 Problem Statement

The proposed model ensures proportional service differentiation over multiple QoS metrics (packet loss rate and queuing delay) among classes at a mobile ad hoc node (i.e. single-hop only). Hence, the problem is to develop scheduling (service rate allocation) and buffering management policies that each node can use to service multiple competing classes in order to satisfy the QoS and system constraints such as maximum available buffer size and time-varying link capacity at each node. The service rate allocation can thus be viewed as an optimization problem performed in a distributed fashion and subject to the above QoS and system constraints.

2 Previous Work

2.1 Service Differentiation in MANETs

Service differentiation in MANETs has originally focused on MAC design, especially tailored to the IEEE 802.11 DCF-based operation [2], [3]. Service differentiation is achieved by setting different values for the lower and upper bounds of contention windows for different service classes. There is however, no explicit guarantee of the level of service differentiation. Due to this reason, although such MAC-level differentiation is beneficial, strong per-hop class differentiation is possible through the adoption of PSD model as described in this paper. More recently, network-level service differentiation was addressed in MANETs, with a stateless QoS framework that uses rate control for best-effort traffic and source-based admission control for real-time traffic [4]. This model is called SWAN (service differentiation in stateless wireless ad hoc networks) and any source can admit its own flow based on sending probing-requests towards a destination. Although this approach claims to be stateless, intermediate nodes may be required to remember whether the flows that traverse them are new or old in order to regulate traffic [4]. In addition, source-based admission control using probing-packets is unrealistic in a dynamic environment like MANETs, as conditions and network topology tend to change fairly frequently. Bandwidth calculations do not take best-effort traffic into consideration, and hence may lead to a false estimation of the available bandwidth.

Very recently, relative bandwidth service differentiation was proposed in [5]. The service profile for a traffic flow is defined as a relative target rate, which is a fraction of the effective link capacity of nodes. This flow-based (as opposed to class-based) approach is ambiguous and unrealistic as service profiles are arbitrarily assigned to flows. It does not show how a target rate for a particular flow is arrived at in any node and passed to other nodes along a specific route. This approach has an important drawback, which becomes clear mainly in shorter timescales, unless these target rates are adjusted dynamically based on performance measurements. The reason for this behavior is that the service quality in each class depends on the short-term relationship between the allocated services to a class and the arriving load in that class [7]. As a result, a higher class can often provide worse QoS than lower classes, invalidating the main premise of relative service differentiation.

Another work on proportional differentiation considered mainly delay differentiation in a WLAN environment [6]. However, our approach attempts for the first time to support both proportional loss and delay differentiation in a dynamic environment such as an ad hoc network.

2.2 Related Work on Proportional Service Differentiation

The proportional service differentiation model was first introduced as a per-hop-behavior (PHB) in the context of wireline differentiated services networks [7]. Two key modules are required to realize PSD: packet scheduling and buffer management. The common approach is to use scheduling algorithms for delay differentiation and use buffer management for loss differentiation. Two scheduling

approaches, the proportional queue control mechanism (PQCM) and backlog-proportional rate (BPR) dynamically adjust class service rate allocations to meet QoS requirements [9]. A waiting-time priority (WTP) scheduler operates on dynamic time-dependent priorities, while there exist a number of WTP variations in the literature. Buffer management functionality can be split into the backlog controller and the dropper. The backlog controller specifies when packets need to be dropped, while the dropper actually drops them. Random early detection (RED) is a typical example of a backlog controller, whereas drop-tail is a widely used dropping mechanism [9].

3 The Proposed System Model

3.1 Model Description

The proportional delay and loss differentiation model is implemented with the use of a packet-forwarding engine. This consists of buffer and scheduling units [9]. The packet buffer is logically organized into Q queues, one for each class (where Q is total number of classes of service). These Q queues share the physical link bandwidth and the buffer space. Scheduling is assumed to be work-conserving, and the scheduler dynamically allocates bandwidth to each of the Q service classes in order to achieve delay differentiation. When the packet buffer is full, the buffer management unit will select and drop certain number of packets from the tail of a particular class in order to meet the proportional loss rate constrains. A first-come-first-served policy is used to transmit traffic from the same class. Although our scheduler inherits the idea of the time-dependent priority concept from WTP, it has been sufficiently augmented with predictions – and hence, our scheduler is called predicted delay proportional (PDP) scheduler. The PDP scheduler is based on a fluid traffic model.

Given that there exist Q service classes which are ordered, so that class-i is better than class-j for $i \neq j$, $1 \leq i < Q$ and $i < j \leq Q$. Assume that $\bar{d}_i(t,t+\tau)$ is the average queuing delay of packets of class-i in the interval $(t, t+\tau)$, where $\tau > 0$ is the monitoring timescale, then for all classes i and j, the following needs to be satisfied per-hop independently of available bandwidth and class loads:

$$\frac{\bar{d}_i(t,t+\tau)}{\bar{d}_j(t,t+\tau)} = \frac{\zeta_i}{\zeta_j} \qquad (1) \qquad\qquad \frac{\bar{l}_i(t,t+\tau)}{\bar{l}_j(t,t+\tau)} = \frac{\sigma_i}{\sigma_j} \qquad (2)$$

ζ_i are delay differentiation parameters (DDPs), being ordered as $\zeta_1 < \zeta_2 < \ldots < \zeta_Q$. In case of loss rate differentiation, for the fraction of packets of a specific class that were backlogged at time t or arrived during the interval $(t, t+\tau)$ and were dropped in this same time interval, the above proportional loss rate differentiation per single hop as given by equation (2) should hold. In this case, σ_i are the loss differentiation parameters (LDPs), being ordered as $\sigma_1 < \sigma_2 < \ldots < \sigma_Q$. In order to satisfy equations (1) and (2), instead of considering only the delay of a top packet of a queue, the PDP scheduler tries to predict the average delay of all the packets in the queue. In this process, the following are assumed during the monitoring interval [9]:

i. The service rate of a specific class is unchanged until the packet is dequeued.
ii. No packet is dropped.
iii. The packet loss rate of a specific class is unchanged.

After predicting the delays of all packets in a specific class queue, the mean delay of that queue is determined. If the predicted delay of each class follows the required proportional delay differentiation with respect to other service classes, the service rate and the loss rate associated with each class are not altered. On the other hand, if the predicted delays of one or many classes do not satisfy a constraint, then either the service rate or the loss rate for each backlogged class need to be changed as explained below. If there are no buffer overflows, the predictions for delay violations are made only once for every Y packet arrivals. The selection of a proper value for Y represents a tradeoff between the runtime complexity and performance improvement with respect to satisfying the constraints. On the other hand, when there is a buffer overflow, packets need to be dropped while still maintaining the constraints. In our work, we consider per-hop proportional loss rate and queuing delay constraints among classes. In case any of these proportional constraints lead to an infeasible system, some constraints need to be relaxed in a specific precedence order until the system becomes feasible. For this purpose, system constraints have priority over proportional constraints [9]. Since the service rate allocation is viewed as an optimization problem, the objective function aims at i) minimizing the amount of traffic to be dropped, and ii) maintaining the current service rate allocation. The first objective ensures that traffic is dropped only if there is no alternative way to satisfy the constraints, while the second is to minimize fluctuations in the service rate allocation to each class. As mentioned in the problem statement, this work considers the maximum buffer size and bandwidth available at each node as system constraints as follows: $\sum_i B_i(t) \le B_M$ and $\sum_i \mu_i(t) = C_M$, where $B_i(t)$ backlog of class-i at a time instance t, B_M is the buffer size of any node M, $\mu_i(t)$ is the service rate allocated to class-i at a time instance t, and $C_M(t)$ is the time-varying link capacity available at any node M. Since the maximum link capacity available to a node vary with time in MANETs, we adopt a method to determine it dynamically as will be explained in section 3.2. Since our model makes delay predictions while keeping the loss rate of a class-i ($1 \le i \le Q$) constant in the monitoring timescale, the optimization problem can be formulated as follows:

Find new service rates $\mu_i'(t)$ and $\mu_j'(t)$ for backlogged service classes i and j, where $i \neq j$, $1 \le i < Q$ and $i < j \le Q$, such that at least the three conditions as given by equations (1) and those imposed by system constraints are satisfied.

The proportional loss rate dropper is a simple dropper sharing the idea of WTP scheduler and has two objectives: i) try to minimize the number of packets being dropped and ii) when there needs to be a packet drop, pick a packet from a certain class in order to keep the loss rate proportional, while satisfying the other constraints. The concept of a weighted loss rate is used for comparison purposes in order to make the packet dropping decision. Whenever a packet tries to join an already full buffer, the packet dropper is triggered. Instead of just dropping the incoming packet, the packet dropper makes a decision as to which priority packet it should drop in order to keep the

loss rate proportional. For this purpose, the loss weight parameters are used here to calculate the weighted loss rate of each class. The values of these parameters are chosen such that $w_1' \times l_1 = w_2' \times l_2 = \cdots = w_Q' \times l_Q$ is satisfied, where $w_i' \times l_i\ (t)$ is the weighted loss rate for each class-i ($1 \leq i \leq Q$). The tail packet of a class with the lowest weighted loss rate is then dropped to keep the loss rate differentiation proportional.

3.2 Link Capacity Estimation

Since we assume the DCF-based operation of IEEE 802.11 as the underlying MAC, the link capacity available to any node needs to be determined dynamically [13]. This is because the available bandwidth varies depending on several factors, namely node-mobility, network topology, power constraints, and contention from other neighboring nodes. In the estimation process we assume that each node employs RTS and CTS frame transmission at the MAC-level in order to minimize the hidden and exposed terminal problems [15]. If time instances t_1 and t_5 and frame size (fs) of the data of Fig. 1 are known, then any node M can predict the bandwidth available using equation (3) [13]:

$$\text{Predicted Bandwidth Availability } (C_M(t)) = \frac{FrameSize}{t_5 - t_1} = \frac{fs}{\Delta t} \tag{3}$$

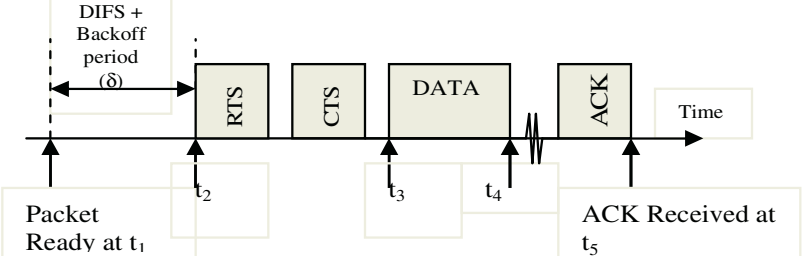


Fig. 1. IEEE 802.11 unicast packet transmission sequence

This link layer measurement mechanism captures the effect of contention on available bandwidth. If contention is high, t5 – t1 would increase and bandwidth available would decrease. This mechanism also captures the effect of fading and interference, since if RTS or CTS get lost, they need to be re-transmitted. This increases the Δt of equation (3), and hence would result in lower bandwidth. It should be noted that the available bandwidth is measured using only successful link layer transmissions. In addition, this estimation process uses average throughput of past packets to estimate the current bandwidth, and it has been proved feasible and robust [13].

3.3 Traffic Regulation

Although our present work does not consider any mechanism for explicit end-to-end absolute service guarantees, our model tries to use localized per-hop and per-node

information to improve end-to-end performance. For this purpose, it works in conjunction with a location-based forwarding mechanism [1]. With the proper forwarder-node selection, the per-hop information is efficiently utilized to minimize congestion, and hence enables traffic regulation.

Sophisticated admission control and traffic policing mechanisms are normally used to regulate traffic, and such mechanisms normally necessitate maintenance of state information [7]. Admission control and policing lead to better performance in the case of fixed IP networks, where routes taken by packets are not volatile. On the other hand, they may not bring in a tangible improvement in the case of dynamic MANETs, as routes taken by packets of the same flow may vary heavily with time. Traffic can be regulated in such situations in a proactive manner by selecting proper non-overloaded forwarding nodes, which is possible in a mesh-like network.

$$\Omega_{MI} = \frac{LET_{MI} * BW_{Available\_I}}{d_{MI} + d_{IB}} \tag{4}$$

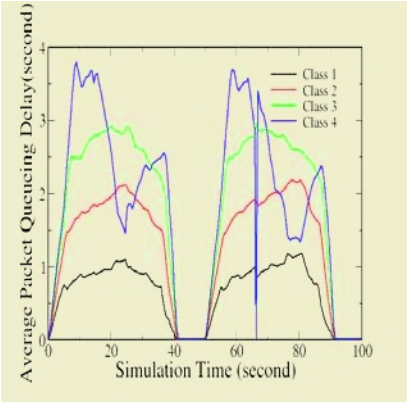
Let N(M) be the neighbor set of node M, and M currently have a packet to be forwarded, d_{MI} be the distance from node M to any of its one-hop neighbors I ($I \in N(M)$), d_{IB} be the distance from any node I ($I \in N(M)$) to the packet's destination B, LET_{MI} be link expiration time of M with respect to I ($I \in N(M)$), and $BW_{Available}$ be the bandwidth available at node I ($I \in N(M)$). The criterion used in our forwarder selection algorithm is given by equation (4). The selection strategy considers the currently available bandwidth to a neighbor, link expiration time (LET) and relative locations of the node-pair under consideration [1]. Any neighbor I of node M that has the highest value for Ω_{IM} of equation (4) can be chosen by our algorithm as a forwarding-node. This way of proper forwarder-node selection is essential in MANETs due to the following two reasons, i) node mobility may lead to a situation where the selected forwarder will soon move away from the sender so that the packet transmission will fail, and ii) the selected forwarder is so busy that the forwarded packets face long delays or get ultimately dropped. Hence, our forwarder selection strategy involves relative mobility prediction and dynamic bandwidth estimation as described below in order to minimize these undesirable aspects. If motion parameters of two neighbor nodes (e.g. velocity, radio propagation range) are known, there is a way to determine the duration of time these two nodes will remain connected [14]. The predicted time is the link expiration time (LET) between two nodes, and this is used in equation (4).

The bandwidth available to a node can be estimated using equation (4) only if t_1 and t_5 of Fig. 1 are known. In our strategy any node should be able to calculate the bandwidth available to any one of its neighbors by listening to the transmission initiated by the latter. However, it is difficult for any node to determine the exact time (t_1) at which a packet becomes ready for transmission in its neighbor. On the other hand, any node can be aware of time instances at which control frames associated to a particular data frame are initiated by one-hop neighbors by listening to the medium promiscuously. This MAC-level listening is facilitated in the DCF-based operation of IEEE 802.11, as virtual carrier sensing with the use of network allocation vector

(NAV) is necessary for the correct DCF-based operation [13], [15]. If the time instance at which the RTS frame is initiated is known, then we can decide a value for t_1 empirically based on node-density and recent traffic characteristics. The frame size of the neighbor node can be statistically determined by taking a time average of the last k-number of packets it generated. Any node can determine this by either analyzing the packets received from its neighbor or by listening to the latter's transmissions promiscuously. Also, under certain circumstances, it may be difficult for a listening node to determine the exact time at which the neighbor receives an ACK. Hence, in such cases we have to determine a value for t_5 empirically by knowing the time at which the data frame transmission ends (t_4). The time instances t_1 and t_5 can be determined by the following two equations: $t_1 = t_2 - DIFS - \delta_1$ and $t_5 = t_4 + SIFS + t_{ACK} + \delta_2$, where SIFS is short inter-frame space, DIFS is distributed inter-frame space, t_{ACK} is the transmission time for ACK frame, δ_1 is added to take care of an extra time involved to access the channel due to the binary exponential backoff mechanism of the DCF and δ_2 is added to take care of an extra time involved if data needs to be retransmitted due to collisions or channel errors [13], [15]. In order to find values for $\delta 1$ and $\delta 2$, we need to first analyze the binary exponential backoff mechanism of the DCF-based operation of the IEEE 802.11 [15]. For simplicity it is assumed that $\delta_1 = \delta_2$, and further we assume that each takes a value of an average backoff window in a saturated network condition. By determining a suitable value for the mean backoff window of an interested neighbor node, we can determine its bandwidth using equation (3) [13], [15].

4 Evaluation Through Simulation

The GloMoSim simulation package [17] was used to evaluate how well proportional service differentiation is achieved in a per-hop manner using our model. Four service classes 1, 2, 3 and 4 were considered for this purpose, where class-1 had the highest priority. Proportional factors for packet loss rate and delay were chosen as 1:2:3:4 in our simulations. The routing protocol used was a location-based one, which had been augmented with our forwarder-node selection algorithm. Our implemented routing mechanism considered simple greedy forwarding only. Traffic was generated using random CBR connections having a payload size of 512 bytes. These CBR connections were randomly generated so that at any moment the total number of source-destination pairs was kept constant – and each session lasted for a time-period that was uniformly distributed between 40 and 50 seconds. Each class contributed equally to the total traffic in the network (i.e. 25%). We considered two QoS-constraints, proportional loss rate and proportional delay. Metrics such as the average delay and loss performance were measured using a sliding window size of 0.5 second. A terrain of 600X 600 m^2 was considered with 80 nodes moving at a maximum speed of 10 ms^{-1} and initiating 10 sessions at any instant. Fig. 3 (a) and Fig. 3 (b) depict per-hop absolute loss rates and absolute average queuing delays of all classes averaged over time intervals of length 0.5 s respectively, when the traffic is generated as a sinusoidal function of simulation time. As it can be seen from these two cases, our model maintains a consistent ratio between classes irrespective of the network load conditions.



(a) Average Queuing Delay (b) Average Loss Rate

Fig. 3. Absolute Per-hop Queuing Delays and Loss Rates as function of simulation time

5 Conclusions and Future Work

In this paper we presented a novel scheduling and buffer management mechanism to realize per-hop proportional loss and delay differentiation in ad hoc networks. In this context, we considered the time-varying nature of wireless link bandwidth available to any node in an ad hoc network and adopted an approach to estimate it. The proposed proportional service differentiation model works in conjunction with location-based forwarding mechanisms. Although the motivation behind this interaction is to facilitate end-to-end service guarantees, in this paper we did not include any explicit mechanism to achieve this objective. In the future, we will concentrate on how to incorporate end-to-end absolute service guarantees in our model through interactions between the PSD unit and the location-based forwarding engine. We will then evaluate how well our enhanced PSD model will guarantee both proportional and absolute service guarantees in an end-to-end manner.

References

1. S. Sivavakeesar, and G. Pavlou : Cluster-based Location Services for Scalable Ad hoc Network Routing. Int. Conf. on Mobile and Wireless Communications Networks (MWCN' 2004), Paris, France, Oct. 2004, 433 – 448
2. M.Barry, A.T.Campbell, and A.Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks", *Proc. 20th Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2001), vol. 1, Apr. 2001, pp. 582 – 590.
3. I.Aad, and C.Castelluccia, "Differentiation Mechanisms for IEEE 802.11", *Proc. 20th Annual Joint Conference of the IEEE Computer and Com. Societies* (INFOCOM 2001), vol. 1, Apr. 2001, pp. 209 – 218.

4. G-S.Ahn, A.T.Campbell, A.Veres and L-H.Sun: SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks. Proc. IEEE INFOCOM'2002, June 2002
5. K.C.Chua, H.Xiao, and K.G.Seah : Relative Service Differentiation for Mobile Ad hoc Networks. *Proc*. IEEE Wireless Communications and Networking Conference (WCNC 2003), vol. 2, Mar. 2003, 1379 – 1384
6. K-C.Wang, P.Ramanathan, "End-to-End Delay Assurance in Multihop Wireless Local Area Networks", *Proc. Global Telecommunications Conference (GLOBECOM 2003)*, vol. 5, Dec. 2003, pp. 2962 – 2966.
7. C.Dovrolis, and P.Ramanathan : A Case for Relative Differentiated Services and the Proportional Differentiation Model. IEEE Network, vol. 13, no. 5, Sep. 1999, 26 – 34
8. C.Dovrolis, and P.Ramanathan : Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping. Proc. 8th Int'l Workshop on Quality of Service (IWQoS 2000), Jun. 2000, 53 – 61
9. N.Christin, and J. Liebeherr : A QoS Architecture for Quantitative Service Differentiation. IEEE Communication Magazine, vol. 41, no. 6, Jun. 2003, 38 – 45
10. A.Striegel, and G.Manimaran : Packet Scheduling with Delay and Loss Differentiation. Elsevier Computer Communications, vol. 25, issue 1, Jan 2002, 21 – 31
11. M.Hamdaoui, and P.Ramanathan: A Dynamic Priority Assignment Technique for Streams with (m, k)-Firm Deadlines. IEEE Transactions on Computers, vol. 44, no. 12, Dec. 1995, 1443 – 1451
12. W.Lindsay, and P.Ramanathan : DBP-M: A Technique for Meeting End-to-End (m, k)-Firm Guarantee Requirements in Point-to-Point Networks. Proc. IEEE Conf. on Local Computers, 1997, 294 – 303
13. S.H.Shah, K.Chen, and K.Nahrstedt : Dynamic Bandwidth Management for Single-hop Ad Hoc Wireless Networks. ACM/Kluwer Mobile Networks and Applications (MONET) Journal, Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks, vol. 10, no. 1, 2005
14. W.Su, S-J.Lee, and M.Gerla : Mobility Prediction and Routing in Ad Hoc Wireless Networks. Int. Journal of Network Management, vol. 11, no. 1, John Wiley & Sons, Jan.- Feb. 2001, 3 – 30
15. O.Tickoo, and B.Sikdar : Queuing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks. Proc. IEEE INFOCOM'2004, vol. 2, Mar. 2004, 1404 – 1413
16. S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, W.Weiss : An Architecture for Differentiated Services. IETF RFC 2475, Dec. 1998
17. X. Zengu, R.Bagrodia, and M.Gerla : GloMoSim: A Library for Parallel Simulations of Large-scale Wireless Networks. Proc. of the 12th Workshop on Parallel and Distributed Simulations, May 1998

Multiservice Communications over TDMA/TDD Wireless LANs*

Francisco M. Delicado, Pedro Cuenca, and Luis Orozco-Barbosa

Department of Computer Engineering,
University of Castilla la Mancha,
Campus Universitario s/n, 02071 Albacete, Spain
{franman, pcuenca, lorozco}@info-ab.uclm.es

Abstract. In recent years, several wireless LAN technologies making use of TDMA/TDD MAC have been designed. In this type of environment, there is the need for a central controller responsible for allocating the bandwidth among all the active mobile terminals. In order to properly carry out this task, the use of simple but effective signaling and bandwidth allocation mechanisms is a must. In this way, the mobile terminals can let the controller know their needs in terms of the bandwidth to be allocated. In turn, the central controller has to properly allocate the available bandwidth among all the competing mobiles taking into account their QoS requirements. In this work, we undertake the design and performance evaluation of the signaling protocols and bandwidth allocation mechanisms. We have paid particular attention to studying the amount of overhead introduced by these mechanisms: an important feature when designing control mechanisms for wireless environments. We validate the effectiveness of our proposed schemes when supporting a multi-service environment comprising four types of services: video, voice, best-effort and background.

Keywords: TDMA/TDD, QoS, Resource Request, Bandwidth Allocation.

1 Introduction

Nowadays wireless local networks represent an alternative to wired LANs. Current wireless LANs operate at transmission rates able to support all types of applications: data, voice, video, etc. It is widely recognized that one of the main advantages of WLANs is their great flexibility: the wire is done away with allowing users to freely connect to the network. Standardization efforts have resulted in the definition of wireless LAN standards one of the main aims of which is to guarantee the interoperability among equipment developed by different

* This work was supported by the Ministry of Science and Technology of Spain under the CICYT project TIC2003-08154-C06-02 and the Council of Science and Technology of Castilla-La Mancha under project PBC-03-001.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 107–116, 2005.

vendors. To date, two of the most important wireless LAN standards are: the IEEE 802.11 standard and the HIPERLAN/2 standard developed by the ETSI (European Telecommunications Standards Institute). Furthermore, the development of wireless communications is now spanning into the area of metropolitan area networks (MANs) where standards, such as, IEEE 802.16, are under study.

The ETSI within the framework of its project BRAN (Broadband Radio Access Networks) has developed various standards for wireless LAN and MAN. One of these is the HIPERLAN/2 standard, which operates in the band of 5 GHz with transmission rates from 6 up to 54 Mbit/s. HIPERLAN/2 supports both operating modes; infrastructure and ad-hoc modes. When operating under the infrastructure mode, the standard distinguishes between two types of devices: the Access Point (AP) and the Mobile Terminal (MT). The AP is responsible for providing connectivity with the core network as well as for adapting the user requirements by taking into account the characteristics of the core network and the services offered by HIPERLAN/2. On the other hand the AP takes care of the distribution of the resources and the coordination of all the MTs located within the cell.

The fact that access is controlled by a single device, facilitates the design and deployment of mechanisms capable of satisfying the QoS requirements of various applications. Based on the MAC protocol defined by HIPERLAN/2, it is possible to build up QoS mechanisms capable of providing the guarantees needed by various applications. HIPERLAN/2 also defines the structure and sequence of the control messages between the MTs and the AP. However, the HIPERLAN/2 does not define the specifics regarding the timing and numerical values of the system parameters, such as the bandwidth to be reserved for a specific type of connection. Neither is it the objective of the standard to describe the specifics of the algorithm to request and grant bandwidth. Therefore, HIPERLAN/2 has intrinsic characteristics allowing it to support several classes of traffic with different QoS requirements.

One of the first issues to solve when developing a structured set of resource allocation mechanisms is how to make the application requirements available to the AP. In this paper, we show that making use of a set of resource request mechanisms designed taking into account the requirements and characteristics of the applications can indeed improve the performance of the network.

Another major issue to be addressed is the definition of the resource granting mechanisms. In TDMA/TDD wireless LANs, such as HIPERLAN/2 and IEEE 802.16 , the AP has to inform the MTs on the bandwidth assigned to each active connection. This information is transmitted to the MT by including it into the frame. It follows that as the number of active connections increases, the frame overhead will increase accordingly. In order to make proper use of the network bandwidth, it is important to limit the amount of overhead. In this work, we will analyze various bandwidth allocation schemes. This refers to the impact of such mechanisms on the structure of the frame, and in particular on the amount of overhead introduced into it.

The main objectives of this work can be simply stated as follows. First, we define a taxonomy of the various service classes to be supported. This taxonomy

will allow us to define the resource request mechanisms according to the needs of each service class. Second, we show that by properly matching the proposed resource request mechanisms to the various types of applications under consideration, our schemes should fulfill the application requirements and enhance the overall system performance. Third, we should show that the overhead introduced into the frame structure can be significantly reduced by making use of a bandwidth allocation mechanism that takes into account the traffic characteristics of the applications.

The article is organized as follows. Section 2 provides a short overview of the HIPERLAN/2 standard. The QoS framework presented in this work is described in Section 3. The results of our performance evaluation study are given in Section 4. Finally, Section 5 concludes the paper.

2 HIPERLAN/2 MAC Protocol

The HIPERLAN/2 MAC protocol [1] is based on a dynamic TDMA/TDD scheme with centralized control, using frames of 2 *ms* as logical transmission unit. Given that the allocation of the frame resources to each MT is made by the AP, the requirements of the application resources have to be known by these entities, which are responsible for allocating the available resources according to user needs. This end, each MT has to request for the required resources from the AP by issuing a *Resource Request (RR)* message, while the AP informs the MT of the positive outcome by using a *Resource Grant (RG)* message.

The HIPERLAN/2 frame is divided into four phases, each phase being composed by a group of transport channels. A transport channel is a logical entity and its classification depends on the type of data that it conveys.

The phases of a frame are:

1. *Broadcast phase*: this phase is used for the communications taking place on the downlink. It contains the configuration parameters of the frame, the resource grant (RG) messages for each active connection in the frame, and the information regarding the number of collisions having occurred in the previous frame.
2. *Downlink phase & Uplink phase*: these phases are formed by a group of PDU trains, which are formed by a preamble and a variable number of Short transport CHannels (SCHs) and Long transport CHannels (LCHs) dedicated to each one of those connections with resources granted in the frame. The LCH channels transport user data and the SCH channels convey error control or resource request messages.
3. *Random Access (RA) phase*: consists of a number of Random CHannels (RCH), which can be used for transmission of resource request messages. A contention process based on a Slotted-ALOHA scheme is used to access the RCH channels.

It is important to note that not all of the transport channels are the same size and that this one depends on the channel type [1].

3 A QoS Framework

3.1 Service Classes and Resource Request Mechanisms

Regarding the underlying HIPERLAN/2 mechanisms, we focus on a two level hierarchy: connection establishment procedures and signaling primitives. Regarding the former, we enable the provision of contracted and non-contracted services. These two types of service relate to the provisioning of the bandwidth required to convey the signaling primitives. In particular, when making use of contracted services, the MTs are assigned a number of SCHs (at least one) for signaling and/or a number of LCHs for data transmission, while in the case of a non-contracted services, the MTs are either polled by the AP or have to go through a contention mechanism to place their resource requests. Our main objective is therefore to propose a general framework for provisioning the HIPERLAN/2 standards of a comprehensive set of QoS mechanisms. It is worth mentioning that, to the authors knowledge, past work in this area has been limited to the definition of resource request mechanisms to support time-constrained services [2], [3]. However, no attempts have been made to define an overall framework integrating all service classes.

Based in the type of resource request mechanisms used by a connection, we can defined four types of them:

- *Type 1*: In this case, the MT operates under a contracted service policy: a certain number of SCH and LCH channels are assigned per frame or every given number of frames to the MT.
- *Type 2*:Under this second type, the resource request mechanism is initiated by the AP through a *polling* mechanism. The AP polls the MT at the beginning of the connection allowing the MT to request the number of SCHs and/or LCHs that it needs in the following frame. The AP may require more than one frame to allocate the LCH channels requested by the MT depending on the network load. As soon as the AP finishes granting the total number of LCH channels requested, the AP polls the MT once again. In order to avoid excessive delays, the AP initializes a timer as soon it polls the MT. When the timer expires, the AP polls once again the MT. This mechanism attempts to compensate for any extra delay incurred during the resource granting process.
- *Type 3*: Under this connection class, the MT operates under a non-contracted service policy. The MT has then to request its resources by sending a message using an RCH channel. The access to this channel is done using a contention process. Once having finished the allocation of the LCH channels required by the MT, the AP, similarly to the Type 2 mechanism, allocates an SCH to the MT. In other words, as long as the MT remains active, an SCH is assigned to it. Otherwise, the MT will have to go through the contention process to send an RR message after an idle period.
- *Type 4*: The main difference between this type when compared to Type 3 comes by the fact that regardless of the activity of the connection, the MT has to go through a contention process, via a RCH channel, to place its

resource request. In particular, different to the previous type, Type 3, once the AP has finished fulfilling the MT requirement, the MT has to go once again through a competitive process to place its request, conveyed via an RCH channel.

3.2 Connection Types vs. Applications

Each one of the connections previously described have been defined bearing in mind that HIPERLAN/2 will have to provide support to various types of applications. In this way, Type 1 is an excellent candidate for CBR applications requiring a fixed capacity to fulfill their QoS requirements.

In turn, Type 2 connections are well adapted for VBR applications, such as video streaming and videoconference, among others. The use of the polling mechanism guarantees that the MTs will be able to periodically gain access to the channel to place their requests.

The Type 3 connections have been designed to accommodate a best-effort type of service. The use of a contention-based process responds to the fact that this type of service does not require any guarantees in terms of delays or jitter. However, by allowing the MT to receive an SCH channel as soon as it has finished to exhausting its previous resource booking, the MT is able to place its next resource request quickly. Finally, Type 4 connections offer the lowest access level. This service has been designed to provide support to traffic that does not require any service guarantee, such as background traffic, while limiting the use of resources for signaling purposes at the lowest level, i.e., minimum overhead.

3.3 Bandwidth Allocation Schemes

One of the main roles of the AP is to define the actual allocation of the channels that compose the frame. In general terms, two groups of channels can be distinguished, data (LCH) and control (SCH) channels. As we shall see, the amount of overhead introduced into the frame will heavily depend on the way the channels are assigned to the various connections. In particular, the overhead can be reduced by contiguously placing the channels associated to a given connection. In order to analyze this important issue, we consider the use of the following three bandwidth allocation schemes:

- FIFO (*First-In-First-Out*): each resource request is served following a FIFO discipline. Given that the there is an explicit classification based on the resource request mechanisms, all the requests are stored in a single queue upon their arrival.
- RR (*Round Robin*): under this policy, a queue is assigned to each connection. The queues are served following a round-robin discipline and are allowed to make use of only one (LCH or SCH) channel per visit. A queue will be visited only once again after all the other queues have been visited. It is important to note that a work conserving strategy is used, i.e., whenever a queue being visited is empty, the next non-empty queue can make use of the available bandwidth. The requests for control channels (SCHs) are assigned a higher priority over the requests for data channels (LCHs).

– MORR (*Minimum Overhead Round Robin*): this scheme is similar in operation to the RR scheme. The main difference lies on the fact that whenever a queue is visited, all the requests present in it are served up to available bandwidth. The main aim of this scheme is to limit the amount of overhead to be introduced in the frame by contiguously allocating the channels pertaining to a given connection.

It should be clear that these bandwidth allocation schemes come to supplement the resource request mechanisms. While the role of the resource request mechanisms is the classification of the various applications, the role of the allocation mechanisms is the distribution of the channels among the requesting MTs based on this classification.

4 Performance Evaluation

In our study we use one HIPERLAN/2 cell operating in centralized mode, which has been implemented in OPNET 10.0 [4]. We assume that the connections have already been established, i.e., the only control messages being sent over the channel are those used by the resource requests mechanisms previously described. In the composition of the frame we use short preambles, guard times of $2\,\mu s$, three RCH channels in the RA phase and the physical mode for the SCH and LCH channels are QPSK3/4 (18 Mbit/s) and 16QAM9/16 (27 Mbit/s), respectively.

Throughout our study, we have considered four main traffic types: video, voice, best-effort and background. The video traffic has been characterized by MPEG-4 [5] video traffic traces. Each video application begins its transmission within a random period given by the expression $t = uniform(0, \frac{12}{f})$ being f the frame rate. In this way, the peak periods of the source rates are randomly distributed along a GOP (Group of Picture) period. The transmission of a video frame is uniformly distributed along the interval of duration of a frame ($\frac{1}{f}$). We use the MPEG-4 sequence *funny* encoded on CIF format at 25 frames/sec.

We assume the use of constant bit-rate voice sources encoded at a rate of 16 Kbit/s according to the G.728 standard [6]. Similarly to the video applications, the voice sources are randomly activated within the first 24 ms of the simulation. The best-effort traffic is generated using the traffic model for Web surfing applications described in [7]. The background traffic generated by each source is a combination of ftp, e-mail and Napster according to the model described in [8]. The traffic sources of these two latter traffic types are initiated at the beginning of the simulation run.

In order to limit the delay experienced by the video and voice applications, an essential condition to guarantee the QoS required by both applications, the maximum time that a unit of video and voice, referred to from now on as packet may remain in the transmission buffer has been set to 100 ms and 10 ms, respectively. These time limits are on-line with the values specified by the standards and in literature [9]. A packet exceeding this upper bound is dropped.

In order to carry out this study, we have considered a scenario where a third of the MTs are running voice/video applications. Another third of MTs generate best-effort traffic and finally all other MTs generate background traffic.

Given that one of the main objectives of the study is to evaluate the performance and effectiveness of the proposed resource request mechanisms, we have carried out two sets of simulations corresponding to two different scenarios. Under the first scenario, namely Scenario 1, all applications have to go through a contention-based process when attempting to transmit each and every resource request packet. Under the second scenario, Scenario 2, each of the applications makes use of a different type of mechanism. The following has been used: voice services make use of the Type 1 mechanism with an LCH channel reserved every 12 frames (this corresponds to a guaranteed data rate of 16 Kbit/s). Video services make use of the Type 2 mechanism with a timer period of 40 ms; the value of this parameter has been derived based on the results obtained in [10]. The best-effort and background traffic make use of the Type 3 and Type 4 mechanisms, respectively.

The second objective of the work is to evaluate the performance of the bandwidth allocation schemes. In particular, we have been interested in studying the frame occupancy in terms of the overhead introduced to properly identify the channels allocated to each MT. In this second part of our study, we will be making use of all the three bandwidth allocation mechanisms introduced in Section 3.3.3.

In our study, we have been interested in assessing the performance in terms of the following metrics: total normalized throughput, overhead, the cumulative distribution functions for the end-to-end delay and jitter, and the packet loss rate. Each point in our plots is an average over twenty five simulation runs, and the error bars indicate the 90% confidence interval.

Figure 1 represents the normalized (carried) throughput as a function of the offered load for both scenarios and all three bandwidth allocation mechanisms. As seen from the figure, as the load increases, the performance of Scenario 1 badly degrades. This situation can be simply explained as follows. Since the MTs have to go through a contention mechanism to place their requests, as the load increases the number of collisions in the RA phase increases dramatically.

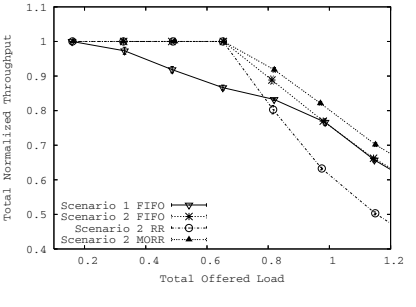


Fig. 1. Traffic Granted

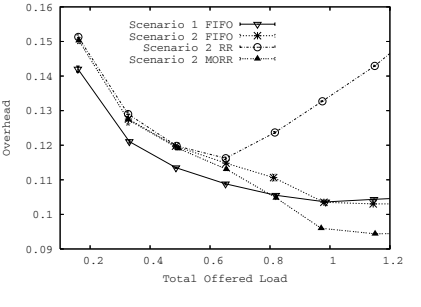
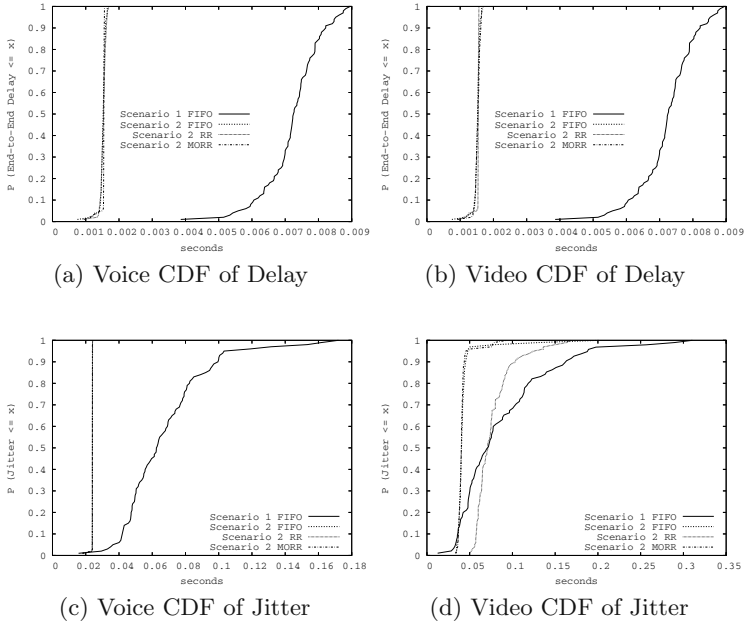


Fig. 2. Overhead

Furthermore, the fact that RR bandwidth allocation scheme exhibits the worst results under heavy load conditions is due to the need to dedicate more channels for control purposes. This problem is partially solved by making use of the MORR at the expense of penalizing the multiplexing gain.

Figure 2 depicts the overhead as a function of the offered load for all the three bandwidth allocation mechanisms under study. As seen in the figure, the overhead decreases as the load is increased for all three mechanisms and for loads up to 50%. However, as the load increases, the overhead introduced by the RR starts to increase steadily. This is due to the fact that by allocating the bandwidth to a larger number of MTs, the number of channels dedicated to conveying control information increases. Under heavy load conditions, this behavior penalizes the system performance by limiting the available bandwidth for actual data transfer.

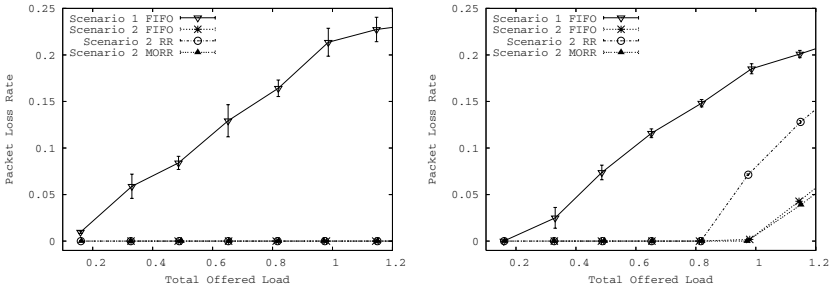
For the case of the FIFO mechanism, the overhead introduced in the frame is lower under Scenario 1 than in Scenario 2. This difference is due to the mechanism used to place the requests and the policy used to serve the requests. Remember that under Scenario 1, the MTs make use of a contention-based process to place their requests. As the load increases, the MTs spend more time attempting to place their requests. As the number of channels requested is being updated during this period of time, a larger number of channels will be requested. Furthermore,



(a) Voice CDF of Delay (b) Video CDF of Delay

(c) Voice CDF of Jitter (d) Video CDF of Jitter

Fig. 3. CDF for the End-to-End Delay & the Jitter for Voice & Video Connections (Offered Load $\cong 0.98$)



(a) Voice PLR (b) Video PLR

Fig. 4. PLR for Voice and Video Connections

since the requests are served following a FIFO policy, the overhead decreases as the number of actual channels used to convey user data is increased.

However, in the case of Scenario 1 making use of a FIFO discipline, the amount of overhead is initially lower than for Scenario 2 and a MORR discipline. This is due to the delay encountered when placing the requests as previously explained. This latter scenario rises as the load increases an converges with Scenario 1 when the network becomes saturated.

Figure 3 shows the CDF of the end-to-end delay and the jitter for a system operating at full load ($\approxeq 98\%$). Figures 3.(a) and (c) shows that voice communications are unaffected since the networks guarantee them the required capacity (Scenario 2). In the case of the video traffic, our results show that the MORR mechanism guarantees an end-to-end delay of less than 50 *ms* to all packets. In the case of the RR mechanism, the video packets can experience up to the maximum allowable end-to-end delay, i.e., 100 *ms*. For the jitter, Figure 3.(d) shows that 95% of the inter-arrival times between video frames are 40 *ms* when MORR or FIFO are used in Scenario 2. This corresponds to the sampling rate of 25 frames/*s*, i.e., a frame every 40 *ms*. In other words, 95% of the video frames arrive to their destination on an isochronous manner. This is an excellent result that indicates clearly the effectiveness of the proposed mechanism.

Figure 4 shows the packet loss rates for voice and video connections. These losses correspond to the packets dropped as soon as they exceed the maximum allowable queuing delay. In the case of voice connections, Figure 4.(a) shows that the losses are completely avoided by statically allocating an LCH every 12 frames and independently of the bandwidth allocation scheme being used. In the case of video connections, the bandwidth allocation scheme plays a major role on their performance. Figure 4.(b) shows that for the case when the RR scheme is used, the PLR steadily increases beyond 70% of the network capacity. Once again, this can be explained by the overhead introduced by this scheme that attempts to multiplex a larger number of connections than the other two bandwidth allocation schemes, namely FIFO and MORR. The use of these two last schemes limits the PLR to less than 1% even when the network operates under very heavy load conditions ($\approxeq 1$).

5 Conclusions

In this work, we have proposed a complete set of control mechanisms and evaluated their performance in terms of various metrics of interest. Our main aim has been to construct a structured set of mechanisms aiming to provide the QoS guarantees required by time constrained applications when coexisting with other services (applications) in a TDMA/TDD wireless network.

We have come to the conclusion that the use of resource request mechanisms adapted to the requirements of various types proves effective as an initial step towards the provisioning of QoS guarantees. We have also evaluated various bandwidth allocation mechanisms, showing that it is possible to make use of a simple scheme to reduce the amount of overhead to be introduced into the frame.

References

1. *Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part1: Basic Data Transport Functions*, ETSI Std. TS 101 761-1, 2000.
2. L. Lenzini and E. Mingozzi, "Performance evaluation of capacity request and allocation mechanisms for HiperLAN2 wireless LANs," *Computer Networks*, vol. 37, pp. 5–15, 2001.
3. E. Mingozzi, "QoS support by the HiperLAN/2 MAC protocol: A performance evaluation," *Cluster Computing*, vol. 5, pp. 145–155, 2002.
4. O. T. Inc., *OPNET Modeler 10.0*, ©1987-2003 OPNET Technologies, Inc., http://www.opnet.com.
5. *Infomation technology- Generic coding of audio-visual objects- Part 2: Visual*, ISO Std. ISO/IEC 14 486-2 PDAM1, 1999.
6. *Coding of speech at 16 kbit/s using low-delay code excited linear prediction*, Std. ITU-T Recommendation G.728, September 1992.
7. G. Colombo, L. Lenzini, E. Mingozzi, B. Cornaglia, and R. Santaniello, "Performance evaluation of PRADOS: a scheduling algorithm for traffic integration in a wireless ATM networks," in *Proc. ACM MOBICOM'99*, Seattle, WA, August 1999, pp. 143–150.
8. A. Klemm, C. Lindemann, and M. Lohmann, "Traffic modeling and characterization for UMTS networks," in *Proc. of IEEE GLOBECOM'01, Internet Performance Symposium*, San Antonio, TX, November 2001.
9. A. Karam and F. Tobagi, "On the traffic and service classes in the internet," in *Proc. of IEEE GLOBECOM'00*, San Francisco, CA, USA, 2000.
10. F. Delicado, P. Cuenca, and L. Orozco-Barbosa, "A QoS-aware resource request mechanism for delay sensitive services over TDMA/TDD wireless networks," in *Proc. of ICETE 2004*, vol. 3, August 2004, pp. 402–408.

Interference-Based Routing in Multi-hop Wireless Infrastructures[1]

Geert Heijenk and Fei Liu

University of Twente, P.O. Box 217,
7500 AE Enschede, The Netherlands
{geert.heijenk, f.liu}@utwente.nl

Abstract. In this paper, multi-hop wireless infrastructures are identified as a way to increase user data rates and/or capacity of wireless systems by means of a high base station density without high base station interconnection costs. For such a system, a new routing algorithm, named Balanced Interference Routing Algorithm (BIRA), is proposed. One of the main features of this new routing algorithm is to take the interference between wirelessly transmitting nodes into account. In BIRA a link cost is calculated considering the interference level of a node and a fixed cost for each link. Based on this link cost, the Dijkstra algorithm is used to compute routes. From the performance analysis, we see that BIRA outperforms other algorithms in terms of obtained data rates for a given available spectrum. BIRA helps to reduce the interference in the network and to achieve higher throughput.

1 Introduction

In order to increase the capacity of cellular communication systems, future generation systems may need to locate base stations much closer together compared to 2nd and 3rd generation cellular systems. Doing so may result in increased frequency reuse and increased data rates and/or capacity. Such a scenario will lead to a very high density of base stations, which will not be all connected to a wired infrastructure, for cost reasons. This will lead to a situation, where data from wireless terminals, such as cellular phones, portable or wearable computers, and sensors, is not transferred to and from the wired infrastructure in a single hop. Since a significant number of base stations do not have a wired connection, they merely serve as a relay, resulting in a situation where most of the data is transferred to and from the wired infrastructure in multiple hops. This leads to a network with (1) base stations connected to the wired infrastructure; (2) stationary relaying base stations without a connection to the wired infrastructure; and (3) terminals (mobile devices generating and absorbing data).

One of the problems to be solved for such a multi-hop wireless infrastructure is the routing of traffic between terminals and base stations connected to the wired

[1] This work is part of the Freeband AWARENESS project (http://awareness.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 117 – 127, 2005.

infrastructure. Compared to the routing problem in ad-hoc networks, the problem is simplified, since nodes are stationary, and most traffic flows either begin or end in a limited set of nodes, i.e., those connected to the fixed infrastructure. On the other hand, since a significant amount of traffic is aggregated, the optimality of routing in terms of demand on the radio spectrum is much more important. The paths used should be such that the throughput of the wireless infrastructure is maximized, given a certain available spectrum and base station receiver sensitivity. We do this by introducing a new cost metric, based on the interference generated to transfer data over a specific link, and by using Dijkstra's shortest path algorithm to determine routes. Our analysis reveals that this cross-layer optimization leads to significantly increased data rates in the same radio spectrum, compared to traditional routing algorithms.

Routing in wired networks is relatively well understood. In wireless access networks, such as 3G cellular networks and wireless LANs, the wireless part of the network is confined to the last hop. As a result, routing comes down to finding the closest access point and routing in the wired part of the network, although the problem is complicated by the dynamics caused by end node mobility. The routing problem in multi-hop wireless networks is much more a research challenge. Considerable research has been done in the area of protocols for routing in ad-hoc networks, which is a very difficult problem [1]. This has lead to proposals that do not scale very well, and are not optimal in terms of spectrum utilization. Nodes in multi-hop wireless networks do not have to be always mobile. The above-mentioned scenario of a multi-hop wireless infrastructure is such a network with stationary wireless nodes.

This paper is organized as follows. Section 2 sketches in which context, and under what assumptions our routing algorithm has been developed. Section 3 describes existing interference-based routing algorithms, and introduces the proposed Balanced Interference Routing Algorithm (BIRA). The performance of BIRA is modelled and evaluated in Section 4. In Section 5, conclusions and future work are given.

2 Context and Assumptions

Our work focuses on the routing problem in a multi-hop wireless infrastructure. The network considered consists of a large number of stationary relaying base stations, and a limited number of special base stations with connection to the fixed network. Each base station serves a number of mobile terminals, with a total offered load of d Mbits per second per base station. All of the offered traffic in a base station is destined for the fixed network. The traffic is relayed via zero or more other base stations to the fixed network. The same amount of carried traffic is to be transported from the fixed network to each of the base stations, i.e. d Mbits per second per base station.

It is assumed that the base stations use CDMA as the access technique for their interconnection network. Perfect power control is assumed for this CDMA network, so that all transmitters use just the transmission power level that is needed to let the receiver decode the signal with the proper quality. That is, the received signal meets the signal over interference ratio requirements (SIR_{target}) of the receiver hardware. It is also assumed that all transmitters have the same maximum transmission power, so

that, as interference levels increase, some transmitters are not able to increase their transmission power any further, which will cause the link between this transmitter-receiver pair to fail.

With respect to the propagation environment, we assume that the signal is only deteriorated by path loss. So the signal strength is assumed to only depend on the transmission power and the distance. It is assumed that the signal strength is decreasing with the distance to the power α, with α between 3 and 4, as is known from experience in many CDMA networks [2]. It is finally assumed that the signal strength decay between each pair of base stations is known. This information can be obtained from feedback on previous transmissions (e.g., from power control information). Alternatively, this information can be derived from the distance between base stations. This can also be learned from previous transmissions (e.g., derived from the timing advance), or derived from positioning information (e.g., using a GPS receiver). In the remainder of this paper, it is assumed that the distance between base stations is known, although this does not preclude the former scenario.

3 Interference-Based Routing

3.1 Existing Routing Algorithms

In recent years, there has been a lot of work on wireless mesh networks and multi-hop routing algorithms (e.g., 802.11s [8], TORA[9], AODV[10]). Here, we are mostly interested in related interference-based routing algorithms. Two such algorithms are Least Interference Routing (LIR) [3] and Minimum Interference Routing Algorithm (MIRA) [4]. In the latter algorithm, the term "interference" does not really refer to the meaning of interference received from the physical layer. Indeed, it focuses more on a better distribution of the network "load". However, in wireless networks the interference at the physical layer is one of the main issues to be solved. The first algorithm will be discussed in some more detail below.

LIR computes a minimum-cost route metric. The cost of the links here takes the possible interference into account. The interference metric is created in each node. The interference generated by a node is considered to be the number of neighbours that can receive a transmission from that node. Therefore, the interference information can be calculated locally. Based on this interference metric, routes are calculated using Dijkstra's algorithm. LIR helps to lower the probability of interference to neighbours efficiently. Since the paths are calculated only based on the number of neighbours who can overhear a transmission, LIR is a simple algorithm to implement.

In CDMA type of networks, it is not sufficient that the interference a transmitter generated only takes the number of neighbours who can listen to a transmission into account. A transmitter can send signals to several receivers at the same time by using different codes with certain transmission power. In this case, the interference a node receives is also related to the distance to the transmitter. With the same transmission power, the closer nodes receive higher interference. Besides, the link cost is not only based on the interference. There are other factors influencing the computing of the

routes, such as the number of hops, fixed link cost, error rate, reliability etc. Based on the reasons mentioned above, we are going to propose a better interference-based routing algorithm for CDMA type of networks.

3.2 The Need for Interference-Based Routing

It is known that in CDMA systems, decrease of interference translates directly into increase of capacity. If interference at the receiver is decreased, less energy per bit is needed to correctly decode the signal. As a result, the transmitter can transmit with less power, which translates in again decreased interference at the other receivers, or transmission at a higher bit rate. This is why power control is so important in CDMA systems, as it reduces the transmission power of transmitters to adjust the level needed to correctly decode the signal at the receiver.

When faced with the problem introduced in Section 2, one straightforward solution might seem to be to let all base stations transmit their signal to the one connected to the fixed network in a single hop. However, since potentially large distances have to be bridged, this leads to high transmission powers, which in turn interfere with other transmissions. Breaking up a large transmission link into stages, leads to a transmission power (measured at the transmitters) that increases linearly with distance. For transmission in a single stage, it is a known phenomena that the signal strength decays with the distance to the power α, with α between 3 and 4. This suggests that it is beneficial to break up a transmission path, and to use multi-hop transmissions instead.

As we see the impact of interference on network capacity, it seems also important to decrease the interference caused by transmissions as much as possible. In a mesh type of network, this can be done by using transmitting nodes that are geographically far away from other (receiving) nodes in the network, because of the decay of the signal strength with distance. So, by taking the interference level caused by transmission over a certain link into account, we can balance and minimize the transmission power in the entire network, and increase the network capacity. This will result in a higher offered load per base station, supported by the wireless multi-hop infrastructure.

3.3 Algorithm Definition

Balanced Interference Routing Algorithm (BIRA) is an algorithm which we propose for interconnection of base stations to the wired infrastructure via multiple wireless hops. It takes the interference a transmitter generates to the other nodes and a fixed transmission cost of each link into account. The idea of this algorithm is to try to generate a new cost function for all the links and compute the routing according to this cost based on the Dijkstra Algorithm. The resulting routes constitute a spanning tree to/from the gateway node. To simplify the problem, in this paper, we assume only a single gateway node, connected to the external infrastructure for each network area.

Let C_{ij} denote the cost for the link between i and j. We define the link cost:

$$C_{ij} = \beta A_{ij} + (1-\beta) IL_{ij} \tag{1}$$

This link cost function constitutes of two parts, weighted by the weight factor β. A_{ij} stands for a fixed cost for the link between i and j. Further, IL_{ij} is the interference level of the link between node i and j, generated to the other nodes in the network.

For determining the interference level, let us consider an arbitrary node r. It will receive interference from the transmission from i to j, when i is sending packets to j, with a power of $I_{ir,j} = P_{ij}/(D_{ir})^{\alpha}$, where $I_{ir,j}$ denotes the interference r receives from the signal i sent to j, and D_{ij} represents the distance between station i and j. The transmission power P_{ij} of the transmission from node i to node j declines with the distance between i and receiver with power of α, where α is again the propagation path loss decay exponent. For a certain reference received power P_{ref} at the receiver j, the transmission power of node i for its transmission to j should be

$$P_{ij} = D_{ij}^{\alpha} P_{ref} \tag{2}$$

So the interference at node r will be

$$I_{ir,j} = \frac{D_{ij}^{\alpha}}{D_{ir}^{\alpha}} P_{ref} \tag{3}$$

So the related interference level, relative to the same reference received power P_{ref} at node r will be

$$IL_{ir,j} = \frac{I_{ir,j}}{P_{ref}} = \left(\frac{D_{ij}}{D_{ir}} \right)^{\alpha} \tag{4}$$

In order to obtain the overall interference level of the link from node i to node j, we add the interference generated by the transmission from i to j and the interference generated by the transmission from j to i, and sum over all potentially interfered nodes r:

$$IL_{ij} = \sum_{r \neq i, r \neq j} \left(\left(\frac{D_{ij}}{D_{ir}} \right)^{\alpha} + \left(\frac{D_{ji}}{D_{jr}} \right)^{\alpha} \right) \tag{5}$$

Using the link costs defined above as input to the Dijkstra algorithm results in routes that tend to cause least interference to other nodes transmissions. Because power control used in CDMA systems decreases transmission power so that the required signal to interference ratio SIR_{target} is just met at the receiver, decrease of interference results in decrease of required transmission power per bit. As a result, the use of BIRA defined above will translate into increased capacity of the network.

From our experiments we have observed that using just the interference level as a link cost metric will in some situation cause very long chains of nodes to be constructed as routes. This is not necessarily good, as in such a chain, each base station has to transmit both its own traffic load and the load received from the previous base station to the next base station, and the traffic flows can take a large "detour". Experiments have shown that introducing a fixed link cost weighted by β avoids this problem, and increases the network capacity.

Typically, a node cannot transmit and receive at the same time in the same frequency band, as the interference generated by a node's own transmission would be too strong for the node to correctly receive the incoming signal. Therefore, we

propose to divide the transmission capacity in two (either in frequency or in time), where a specific node always uses one frequency (or time slot) for transmission, and the other for reception. For that purpose, all nodes are divided into two groups, based on the distance (in number of hops) to the gateway node. During the transmission of the nodes with even number of nodes to the gateway node, the nodes with odd number of nodes to the gateway node will receive and vice versa. Note that this mechanism of dividing the transmission capacity relies on the use of static routing. It is not a feasible solution for networks with dynamic routing.

The main algorithm run by all nodes will thus be as follows:

i. Determine link costs using Eq. (1).
ii. Run Dijkstra algorithm to find spanning tree from/to gateway node.
iii. Determine transmission frequency / time slot, based on number of hops to gateway node (even / odd).

4 Performance Modeling

In order to evaluate the performance of BIRA, we have developed a model of a multi-hop wireless infrastructure. This model is presented in Section 4.1. Using this model, we analyze the performance of BIRA, first for a basic network topology, in Section 4.2, and then for large number of randomly generated network topologies in Section 4.3. In the analysis, we study the impact of our tuning parameter β, and compare BIRA with three other routing algorithms. These are the Least Interference Routing (LIR) algorithm, described in Section 3.1, and two rather straightforward algorithms: Minimum Distance Routing (MDR), and Minimum Hops Routing (MHR). MDR uses the geographic distance between two nodes as link cost, whereas MHR gives each link an identical link cost. Note that MHR is equivalent to BIRA with $\beta = 1$. After using the Dijkstra algorithm to calculate least cost paths, MDR will give routes with minimum geographic distance, whereas MHR will give routes with a minimum number of hops. In the remainder of this section, some of the results are shown. More extensive analysis result can be found in [7].

For the BIRA algorithm, throughout this performance evaluation, we assume that the fixed link cost equals 1 ($A = 1$), and the propagation path loss decay exponent equals 3 ($\alpha = 3$). Furthermore, for all four algorithms, a maximum transmission range is assumed, i.e., links between nodes only exist when the distance between the nodes is less than the transmission range.

4.1 Model

The model describes a set of n nodes i ($1 \leq i \leq n$), where node n represents a gateway base station, connected to the fixed network, and all the other nodes represent stationary relaying base stations. We assume that each of the nodes inserts an amount of traffic into the network, destined to the gateway node n with a data rate of d Mbits/s. Further, the gateway node inserts for each other node in the network a traffic flow with a date rate of d Mbits/s into the network. The resulting transmission bit rate of a

node i to a specific other node j (b_{ij}) is a multiple of d Mbits/s, depending on the number of flows that are aggregated to and from the gateway node in that specific node. We implemented the aforementioned routing algorithms, to find the routes used for the traffic flows and the transmission bit rates of each of the nodes.

We assume that the inserted traffic is transferred wirelessly between the nodes, using CDMA technology. For determining the transmission power P_{ij} required for the transmission from node i to node j, we use a formula derived from (5):

$$P_{ij} = SIR_{t\arg et} \times \frac{\mu_i b_{ij}}{w} \times D_{ij}^{\alpha} \times \left(PN + \sum_{r \neq i} \frac{P_r \times CI_{rj}}{D_{ij}^{\alpha}} \right) \tag{6}$$

Here, μ_i is the activity level of node i, and w is the chip rate used in the system. The resulting $(\mu_i b_{ij})/w$ is the reciprocal of the processing gain of the transmission. SIR_{target} is the required signal to interference ratio at the receiver, whereas PN is the background noise. In all the experiments of this paper, we assume that μ_i is 1, w is 3.84 Mchips/s, SIR_{target} is the linear equivalent of 5dB, and PN is calculated assuming a background noise of -174 dBm/Hz with 5 MHz bandwidth. The last term in the equation denotes the received interference at node j. CI_{rj} is 1 if node r is transmitting in the same frequency / time slot where node j is receiving, otherwise it is 0. Finally, P_i is the sum of the transmission powers of all transmission done by node i, i.e.

$$P_i = \sum_j P_{ij} \tag{7}$$

Note that these equations assumes the propagation channel only to exhibit path loss, perfect power control, and complete orthogonality of multiple transmissions by the same node. By solving this system of equations iteratively, we can obtain the transmission power for each node. We ran a fixed number (10000) of iterations, after which the change of transmission powers between subsequent iterations was negligible. Further, we varied the offered data rate per node d, to find the highest value for d, for which the transmission power of each node is below a predefined maximum value. Thus, we compared different routing algorithms with respect to the offered data per node a network can support. The model has been implemented in Matlab 6.5.

4.2 Basic Analysis

Based on the model we introduced above, we start the comparison using a basic topology. Suppose in a 1000×1000 m^2 network area, 10 base stations are distributed evenly in a straight line with the same space 100 meters between neighbours. Station A, B, C,…, H, and I are stationary relaying base stations, while J is connected to the fixed infrastructure by a wired connection. All the packets are sent to or received from the fixed network via Station J. In this network, we can evaluate the potential gain of transmitting flows in multiple hops, instead of a single hop. If we apply MHR without communication range to compute the routes, all nodes will take a single hop to the destination. Intuitively, this will generate the most interference to the network. In this sense, MHR is not a suitable algorithm for the network. We are going to compare MHR with BIRA by observing the maximum data rate with different communication range. A larger maximum data rate from BIRA than from MHR is expected.

4.2.1 Methodology

In MHR, the cost of all the links is same. This implies that always the direct links are used, if the link is available. Through the model above, we can obtain the maximum data rate d in each node. We will compare the value of d between different algorithms, and see which algorithm can obtain the maximum data rate/capacity.

4.2.2 Results

Fig 1 shows two different routes obtained by MHR and BIRA (with weight parameter $\beta = 0.5$) with unlimited communication range. Fig 1-a shows that with MHR, all the relaying nodes reach the destination (J) within the least number of hops (a single hop in here). However, with BIRA, nodes are trying to reach the closest node in order to minimize the interference as show in Fig 1-b. Meanwhile, the fixed link cost A also tries to balance the length of branches, and avoid the long branches as we mentioned in subsection 3.3., so that the first node chooses the path via the third node, instead of the second one.

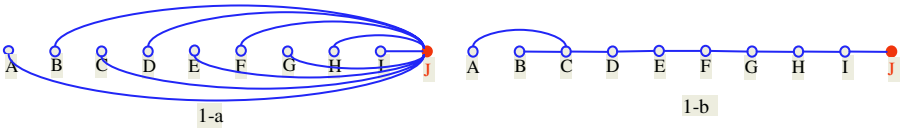


Fig. 1. Routing for MHR algorithm (1-a) and BIRA (1-b) with unlimited communication range

Because the transmission power increases with the distance to the power of 3, the nodes in Fig 1-a need more transmission power to transmit the signals, which implies more interference in the network. Intuitively, the routes in Fig 1-b can decrease the interference and obtain larger data rate. The modelling results proved our assumption.

Table 1. Maximum data rates for different communication range

Communication Range (meters)	100	200	400	600	800
D_{MHR} (Mbit/s)	0.105	0.027	0.008	0.005	0.002
D_{BIRA} (Mbit/s)	0.105	0.105	0.105	0.105	0.105

Table 1 shows the maximum data rates the network we describe can obtain by using algorithm MHR and BIRA with different communication range. D_{MHR} and D_{BIRA} represent the maximum data rate MHR and BIRA can obtain respectively. It can be observed that bridging a distance in multiple hops really leads to an increased maximum data rate. MHR can be forced to do so, by limiting the communication range, whereas BIRA automatically chooses this optimal routing. It must be noted that this is an extreme case, in which the rationale behind the BIRA algorithm is exploited very well.

4.3 Extensive Analysis

In order to analyse a more realistic situation, in this subsection, we modelled a network with 30 nodes randomly distributed in an area of 1000m×1000m, where the last generated node is the gateway node connected to the fixed network. Moreover, we assume that all the nodes have 600 meters communication range. We will compare BIRA with different values of the weight parameter β to optimize the value of β. Furthermore, we will compare BIRA with several other routing algorithms, i.e., MDR, MHR, and LIR.

4.3.1 Methodology

In order to compare these algorithms, we define

$$\delta = \frac{d_1 - d_2}{d_2} = \frac{d_1}{d_2} - 1 \tag{8}$$

where d_1 and d_2 denote the maximum data rates for two different routing algorithms. When δ is equal to 0, $d_1 = d_2$, which is one of the important features of δ; when $\delta < 0$, $d_1 < d_2$; when $\delta > 0$, $d_1 > d_2$.

We generated several topologies. For each topology, we obtained a δ for a specific pair of routing algorithms. We further define the function

$$f(\delta = i) = \frac{\text{the number of } \delta\text{, where } \delta \le i}{\text{the total number of } \delta} \tag{9}$$

Especially, f(0) represents the fraction of cases where d_1 is worse than or equal to d_2.

4.3.2 Results

Two experiments are done to evaluate the performance of BIRA. The first experiment compares BIRA with different values of β in order to obtain the optimal value. With this optimal value, BIRA is compared with MHR, MDR, and LIR in the second experiment.

In Experiment 1, we generate 100 topologies to compare BIRA with $\beta = \beta_1 = 0.4$ and $\beta = \beta_2 = 0$, 0.2, 0.5, 0.6, 0.8, and 1, respectively, so that d_1 stands for the data rate of $\beta = 0.4$, while d_2 represents the data rates of the others. By comparing the value of f(0) and 90% confidence interval of δ, we conclude that the optimal value of β lies between 0.2 and 0.6, whereas $\beta = 0.4$ seems to be a good choice.

We take $\beta = 0.4$ and compare BIRA with MHR, MDR, and LIR by analyzing the results of 100 topologies in Experiment 2. We define the data rate of BIRA as d_1 and the others as d_2. When comparing BIRA with MDR, we found the average value of δ is 1.5, and $f(0)$ equals 0.13, i.e., the date rate obtained with BIRA is on average 2.5 times the data rate obtained with MDR, and in 87% of the topologies, BIRA had better performance. When comparing BIRA with MHR, we found the average value of δ is 1.8, and $f(0)$ equals 0.35, i.e., BIRA obtains on average 2.8 times the data rate of MHR, and in 65% of the topologies BIRA achieves better performance than MHR. When comparing BIRA with LIR, we found the average value of δ is 1.4, and *f(0)* equals 0.22, i.e., BIRA obtains on average 2.4 times the data rate of LIR, and in 78% of the topologies BIRA obtains larger data rates than LIR. The results are further illustrated in Fig 3, where the function f is plotted for the three comparisons.

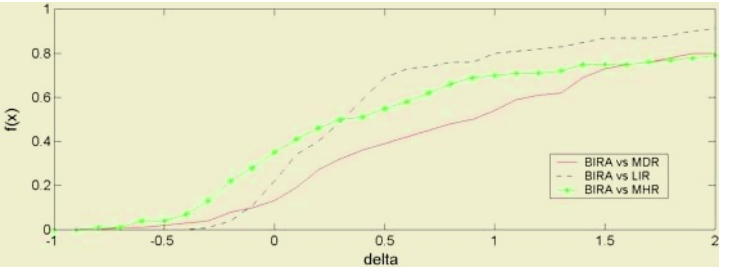


Fig. 2. Comparison of BIRA with alternative routing algorithms

5 Conclusions and Future Work

A new routing algorithm for multi-hop wireless infrastructures, the BIRA algorithm, was proposed. BIRA is promising for the interconnection of high density base stations, where some of the base stations do not have a direct connection to the fixed network. Such a base station will send/receive packets to the wired infrastructures via multiple hops. BIRA is based on the Dijkstra algorithm, where the link cost is a weighted combination of fixed link cost and interference level. Modelling and performance analysis shows that using a combination of fixed link cost and interference level yields better performance than using one of the two as a cost function. More specifically, using a weight factor, β, of 0.4 is definitely better than using β below 0.2, or above 0.6, whereas, it is probably better than other values within this interval. BIRA with the value of $\beta = 0.4$ outperforms the MHR, MHR, and LIR algorithm in terms of obtained data rates for a given available spectrum. It helps to reduce the interference in the network and to achieve higher throughput.

Further work includes the extension of BIRA to situations where the node locations are unknown, so that interference levels have to be derived from physical layer procedures, e.g., from power control information. Also, the extension of BIRA, to allow for rerouting to cope with link or node unavailability needs to be investigated. Finally, analysis of topologies with multiple gateway nodes has to be performed. The application of interference-based routing to multi-hop wireless systems with random access techniques such as the one used in IEEE 802.11 Wireless LAN is also a promising field of research.

References

1. Xiaoyan Hong, Kaixin Xu, M. Gerla, "Scalable routing protocols for mobile ad hoc networks", *IEEE Network, Vol. 16, No. 4,* July / August 2002.
2. T.S. Rappaport, “Wireless Communication: Principles & Practice”, *Prentice Hall PTR*, ISBN 0 133 75536 3, New Jersey, 1996.
3. Jochen H. Schiller, “Mobile Communications”, *Addison Wesley Longman*, Inc., ISBN 0 201 39836, 2000.

4. Murali S. Kodialam, T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering", in *Proceedings of INFOCOM (2)*, pages 884-893, 2000.
5. H. Holma, A. Toskala (ed.), "WCDMA for UMTS – Radio access For Third Generation Mobile Communications", *John Wiley & Sons*, ISBN 0 471 48687 6, England 2001.
6. Raj Jain, "The Art of Computer Systems Performance Analysis", *John Willey & Sons*, Inc., ISBN 0 471 50336 3, 1991.
7. Fei Liu, "Routing in Multi hop Wireless Infrastructures", Master thesis*, University of Twente,* 2004.
8. Jim Hauser, Dennis Baker, W. Steven Conner, "Draft PAR for IEEE 802.11 ESS Mesh", http://www.ieee802.org/11/PARs/11-04-0054-02-0mes-par-ieee-802-11-ess-mesh.doc, 2003.
9. Charles E. Perkins and Elizabeth M. Royer, "Ad hoc On-Demand Distance Vector Routing" *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

A Probabilistic Transmission Slot Selection Scheme for MC-CDMA Systems Using QoS History and Delay Bound

Jibum Kim, Kyungho Sohn, Chungha Koh, and Youngyong Kim

Department of Electrical and Electronic Engineering,
Yonsei University, Shinchon-Dong, Seodaemoon-Ku,
Seoul 120-749, Korea
{voodoo96, heroson7, ski327, y2k}@yonsei.ac.kr

Abstract. In this paper, we propose an efficient transmission slot selection scheme for Band Division Multi-Carrier-CDMA (BD-MC-CDMA) systems under the constraints of packet loss and delay bound for each individual session. By utilizing channel dynamics together with the delay deadline and loss history, one can determine whether to transmit or not on each time slot, based on the prediction of future channel variations. In this way, one can enhance the chance for transmitting packets with the best channel quality. To validate the efficiency of the proposed algorithm, we model each sub-band as a discrete time Markov Chain using a finite state Markov channel (FSMC) and derive the criteria required for transmission decision. Simulation results show that our proposed scheme can satisfy quality of service (QoS) requirements for real-time traffic with a minimum use of power, while increasing throughput of non-real-time traffic with the power saved from real-time traffic.

1 Introduction

Recently, Band Division Multi-Carrier-CDMA (BD-MC-CDMA) [1], which is one of the variations of MC-CDMA, has been proposed for high-speed wireless communications. In BD-MC-CDMA systems, the transmitter selects the frequency bands which are relatively under good condition according to feedback information from the receivers, thereby decreases required transmission power for each receiver according to its QoS requirements.

Several scheduling algorithms for Orthogonal Frequency Division Multiplexing (OFDM) systems have been proposed in recent years [2], [3]. A practical dynamic resource allocation scheme for OFDM systems has been investigated in [2]. However, it simply extended proportional fairness (PF) scheduling algorithm for CDMA/HDR (High Data Rate) to multi-carrier systems. A cross-layer adaptive resource allocation algorithm for packet-based OFDM systems has been studied in [3]. However, it concentrated on non-real-time traffic without considering QoS of real-time traffic.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 128–137, 2005.

Resource allocation problems in BD-MC-CDMA systems are investigated in Mori and Kobayashi's paper [4]. Mori's scheme selects a frequency band according to the SIR estimation with the pilot signal at mobile station and selects the most efficient transmission time slot using the SIR threshold. Although Mori's scheme provides some interesting and innovative concepts, it incurs the following issues to handle. First, it does not consider the heterogeneous QoS requirements of diverse traffic types in terms of packet loss ratio and delay bound. Second, system resources can be wasted because it transmits the packet regardless of the current channel condition when confronting the risk of a time-out. In other aspects, it is difficult to decide an appropriate threshold value, which is critical for the overall performance of proposed algorithm presented in [4].

In this paper, we propose a novel transmission slot selection algorithm for BD-MC-CDMA systems, which accommodates diverse QoS requirements while minimizing total power consumption. We utilize the opportunity for channel improvement while considering the margin for an individual packet's delay deadline and the individual session's packet loss ratio target.

The remainder of this paper is organized as follows. In Section 2, the system model considered in this paper is given and the architecture of the proposed algorithm is described. In Section 3, we present a description of the novel packet scheduler developed for BD-MC-CDMA systems. The performance of the proposed scheme is investigated in Section 4. Finally, Section 5 gives the conclusions.

2 System Model

2.1 BD-MC-CDMA Systems

A BD-MC-CDMA transmitter spreads the original data stream over multiple sub-carriers using a given spreading code in the frequency domain. Each sub-band consists of the same number of sub-carriers and the number of sub-carriers corresponds to the length of the spreading code. The base station selects the sub-band with the best channel quality for each user, adopts the appropriate ranking mechanism and transmits data to the mobile station via the selected sub-bands.

2.2 Scheduler Architecture

Figure 1 shows the architecture of the proposed algorithm. The proposed scheduler is composed of three sections. In priority determination section, the priority of the packet is determined. The optimal packet transmission time is determined during the probabilistic transmission slot selection section. In resource allocation section, we allocate the packet to the sub-band with the minimum power requirements. In our scheme, scheduling is performed on a frame by frame basis.

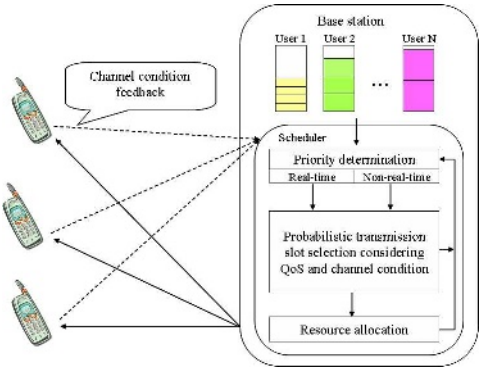


Fig. 1. Architecture of the proposed scheduler

3 Proposed Algorithm

3.1 Priority Determination

In order to provide guaranteed QoS for real-time traffic, a higher priority is always given to real-time traffic by decoupling real-time traffic and non-real-time traffic. The Delay-earliest-due-date (Delay-EDD) [5] and virtual clock [6] are used to determine the packet transmission order for real-time traffic and non-real-time traffic, respectively. It is well known that Delay-EDD scheme can decouple the bandwidth and delay requirements [7]. We choose virtual clock policy for simplicity while providing fairness of non-real-time traffic.

3.2 Probabilistic Transmission Slot Selection

3.2.1 FSMC Model

We assume a slow varying flat Rayleigh fading channel in each sub-band. A slowly varying flat Rayleigh fading channel can be represented as a FSMC model [8]. We assume that the Rayleigh fading channel is slow enough that the channel gain remains unchanged for the duration of a frame. Furthermore, we also assume that each sub-band suffers from independent fading and the fading fluctuation follows an exponential distribution with an average of 1.0.

3.2.2 Evaluation of Channel Quality Improvement

To best utilize channel fluctuation, it is necessary for each user to wait for better channel quality, as long as delay deadline is not expired. Then, the remaining question is how to evaluate the probability that a given channel will ever improve before the delay deadline occurs. Suppose that the current sub-band, which is selected for a certain user to transmit, is in state m as shown in Fig. 2. Then, one can derive the probability that this sub-band will be in a better channel condition more than once before a time-out occurs for the specific packet using the concept of *taboo probability*. A taboo probability, defined on Markov Chains,

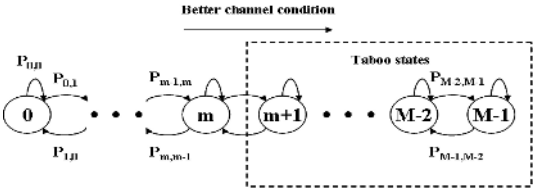


Fig. 2. Illustration of the probability $P^l_{m,better}$

is the probability of going from one state to another without visiting a particular set of states, known as *taboo states*.

Let $mP^l_{m,n}$ be the l-th step transition probability from state state m to n, conditioned on never visiting state m during transitions. In this paper, l-th step can be defined as l-th frame occurring after current frame. Then, one can derive the probability that this sub-band will never be in a better channel condition within the l-th frame, $(m+1)P^l_{m,n}$, by denoting states $m+1$,..., $M-1$ as taboo states (see Fig. 2).

Then, $(m+1)P^l_{m,n}$, $n = 0$,..., m, is given as:

$$(m+1)\,P^l_{m,n} = \begin{cases} (m+1)\,P^{l-1}_{m,0}P_{0,0} + (m+1)\,P^{l-1}_{m,1}P_{1,0}, & n=0 \\ (m+1)\,P^{l-1}_{m,n}P_{n,n} + (m+1)\,P^{l-1}_{m,n-1}P_{n-1,n} & \\ \quad + (m+1)\,P^{l-1}_{m,n+1}P_{n+1,n}, & 1 \le n \le m-1 \\ (m+1)\,P^{l-1}_{m,m}P_{m,m} + (m+1)\,P^{l-1}_{m,m-1}P_{m-1,m}, & n=m \end{cases}$$

By subtracting $(m+1)P^l_{m,n}$ from 1, the probability that a specific sub-band in state m will be in a better channel condition more than once within the l-th frame, $P^l_{m,better}$, can be represented as:

$$P^l_{m,better} = 1 - \sum_{n=0}^{m} (m+1)\,P^l_{m,n}$$

3.2.3 Probabilistic Transmission Slot Selection Considering Both QoS and Channel Condition

After the initial priority determination, we decide whether to transmit the packet or not during the next frame by using a probabilistic transmission slot selection scheme. In this application, the time-out value represents the number of remaining frames from the current frame before a packet time-out occurs. A time-out value of '1' represents the fact that the next frame is the last chance for transmission. We determine optimal packet transmission time considering three factors: delay bound, packet loss ratio history of an individual session and the probability that the next selected sub-band will be in a better channel condition more than one time before a time-out occurs.

Denote $P^l_{m,better,k}$ as the probability that the k-th sub-band in state m will be in a better channel condition more than once within the l-th frame, P_i as the probability that the next selected sub-band will be in a better channel condition

more than once before a time-out occurs, PLR_i as the average packet loss ratio until the previous frame for user i. The expression $TPLR_i$ indicates that the target packet loss ratio of user i, $(Average\ SIR)_i$ represents an average value of SIR in all sub-bands for user i and $SIR_{i,k}$ denotes the SIR value of the k-th sub-band for user i.

$P^l_{m,better,k}$ can be derived in the same way as $P^l_{m,better}$ presented in Section 3.2.2. Then, if the time-out value is l, P_i can be calculated as follows:

$$P_i = \frac{1}{K} \sum_{k=1}^{K} P^l_{m,better,k}$$

where K is total number of sub-bands. Through the priority determination method presented in Section 3.1, suppose that the j^*-th packet of user i^* has the highest priority. If it has a time-out value larger than one, we decide whether to transmit the packet or not during next frame by using the following criteria:

$$MAX\{1 - P_{i^*}, PLR_{i^*}/(TPLR)_{i^*}\} \geq Uniform\ random\ variable$$

$1 - P_{i^*}$ represents the probability that user i^* will never experience a better channel condition before time-out. If $1 - P_{i^*}$ is adequately large, it means that the user i^* will experience little chance for better channel quality than with the current situation existing before delay deadline. If PLR_{i^*} gets closer to $TPLR_{i^*}$, the transmission probability should be increased in order to control the packet loss ratio experienced under target value. To adopt the dominant factor between $1 - P_{i^*}$ and $PLR_{i^*}/TPLR_{i^*}$, we use the max operator between them.

On the other hand, if the j^*-th packet of user i^* with the highest priority has a time-out value equal to one, (i.e., last chance to transmit before deadline), we use a single threshold value. If PLR_{i^*} is greater than or equal to $TPLR_{i^*}$:

$$PLR_{i^*} \geq TPLR_{i^*}$$

Then, the packet should be given a chance to transmit in the next frame in order to meet the anticipated target packet loss ratio. However, if PLR_{i^*} is smaller than $TPLR_{i^*}$, we compare the SIR value of the sub-band k^* of the packet with $(Average\ SIR)_{i^*}$. If SIR_{i^*,k^*} is better than the $(Average\ SIR)_{i^*}$, then

$$SIR_{i^*,k^*} \geq (Average\ SIR)_{i^*}$$

and the average packet loss ratio until previous frame of user i^* satisfies the following equation, then

$$PLR_{i^*}/TPLR_{i^*} \geq Uniform\ random\ variable$$

packet j^* is allocated to the sub-band k^* in the next frame. Otherwise, the scheduler should drop the packet because the packet loss ratio of user i^* has some margin available to the target packet loss ratio and it would be beneficial not to transmit when the channel quality is bad.

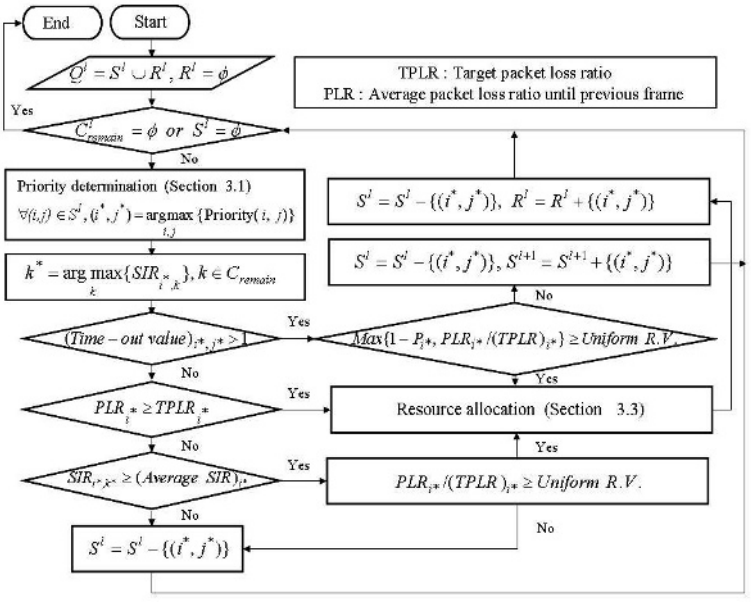


Fig. 3. Flow chart of the proposed algorithm

3.3 Resource Allocation

If transmission for next frame is decided by the probabilistic transmission slot selection portion, we allocate each packet to the sub-band with the best channel quality, as long as the sub-band is not depleted of available code resources. Consequently, the base station can decrease the transmit power in such selected sub-band. Once the optimal sub-channel is determined, the minimum power level that satisfies the BER requirement of the traffic is assigned. In this way, we iterate on number of packets for resource allocation.

3.4 Aggregated Flow of the Proposed Algorithm

A flow chart providing the detailed process of the proposed algorithm is illustrated in Fig. 3. Assume that the j^*-th packet of user i^* has the highest priority and the k^*-th sub-band in C^l_{remain} is the best channel quality for that packet. Define $Q^l = S^l \cup R^l$, where S^l is a set of the packets waiting to be scheduled in the queue in l-th frame and R^l is a set of already scheduled packets in l-th frame. We define C^l_{remain} to be a set of the sub-bands that has remaining codes in l-th frame.

4 Performance Evaluation

4.1 Simulation Model and Parameters

Three different traffic types are considered. Table 1 summarizes the numerical values used for the simulation model.

Table 1. Heterogeneous traffic types used in the simulation

Traffic types	Data rate	BER requirement	Delay bound	Packet loss ratio bound
Real-time traffic				
Voice	16.5 kbps	10^{-3}	30 ms	0.01
CBR video	160 kbps	10^{-5}	60 ms	0.01
Non-real-time traffic				
Data	250 kbps	10^{-9}	2 s	0.001

Table 2. System parameters used for BD-MC-CDMA systems

Frame length	System bandwidth	Number of sub-carriers	Number of sub-bands	Spreading factor
10 ms	2.5 Mhz	64	8	8

- *Voice Traffic*: The voice traffic is modeled as a two state Markov Chain with talkspurts and silence gaps. The average durations of talkspurts and silence gaps are 1.00 and 1.35 s, respectively. The voice packet size is set to 384 bits.
- *CBR Video Traffic*: This model is a constant bit stream with the rate equal to 160 kb/s. The CBR video packet size is set to 1600 bits.
- *Data Traffic*: This model is used to simulate a non-real-time traffic. The bit rate is equal to 250 kb/s and the holding time of each data session is assumed to be exponentially distributed with a mean equal to 20 s. The data packet size is set to 2500 bits.

We consider a downlink BD-MC-CDMA system in single cell environments, thereby, no inter-cell interference is assumed. A slowly varying flat Rayleigh fading channel in each sub-band is assumed. We use an eight-state FSMC model with the maximum Doppler frequency of 5 Hz and choose an equal probability method for dividing channel gains as described in [8]. Total transmission power of a base station is assumed to be limited. We assume no error in the channel information feedback. The system parameters of BD-MC-CDMA systems, considered in this paper, include frame length, system bandwidth, number of sub-carriers, number of sub-bands and spreading factor as shown in Table 2.

4.2 Comparison with Existing Algorithms

To evaluate the performance of our proposed algorithm, we compare some very recent algorithms in similar context. The details of each algorithm other than the proposed algorithm are given as follows:

- *Mori's Scheme*: This scheme is explained in [4]. In Mori's scheme, no priority determination method is offered. Therefore, we employ the basic FCFS service discipline without decoupling real-time traffic and non-real-time traffic.

– *Wang's Priority*: For this scheme, we adopt the priority function proposed in [9]. Other than the priority section, we use our own probabilistic transmission slot selection method and resource allocation scheme. We compared this scheme, especially with respect to the priority determination section. In [9], higher priority is not always given to real-time traffic over non-real-time traffic.
– *No PLR History*: This scheme uses our priority determination and resource allocation methods, but does not use the probabilistic transmission slot selection scheme. Therefore, the comparison will highlight the effects of our proposed probabilistic transmission slot selection section.

4.3 Simulation Results and Discussion

The average packet loss ratio of voice traffic is shown in Fig. 4. These results show that our proposed scheme can guarantee the desired packet loss ratio even with a very high system load. Mori's scheme does not satisfy target packet loss ratio because it employs a basic FCFS service discipline and thereby fails to decouple real-time traffic and non-real-time traffic.

The advantages of the priority determination section are shown in Fig. 5. Compared with the proposed scheme, Wang's priority does not show good QoS provision when real-time traffic load becomes high, due to the insufficient decoupling of real-time and non-real-time traffic, resulting in a considerably high packet loss ratio for CBR video traffic.

In Fig. 6, we present a throughput comparison of data traffic. Wang's priority scheme has the best performance in throughput. However, Wang's priority scheme and Mori's scheme sacrifice the target packet loss ratio of CBR video traffic as shown in Fig. 5. This figure also demonstrates the performance improvements achieved by the proposed algorithm when compared to the No PLR history scheme in data throughput. When the system load is 0.7, a gain of more than 20% in throughput can be achieved using the proposed scheme compared to the No PLR history scheme.

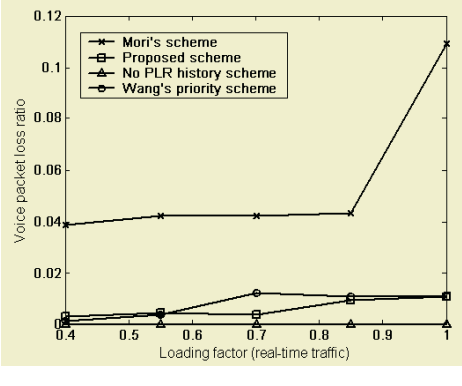


Fig. 4. The average packet loss ratio of voice traffic

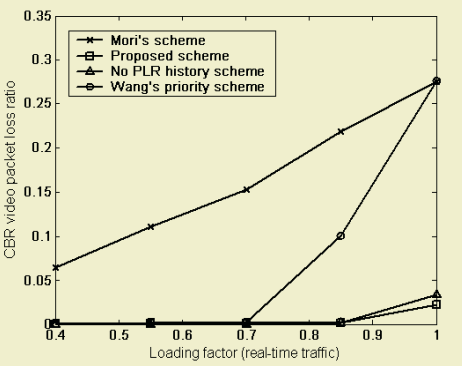


Fig. 5. The average packet loss ratio of CBR video traffic

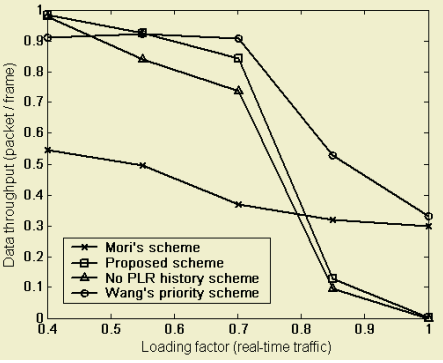


Fig. 6. Throughput of data traffic

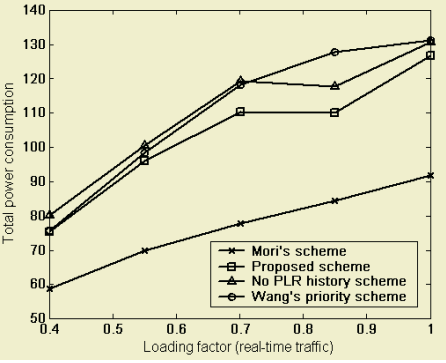


Fig. 7. Total power consumption

In Fig. 7, the total power consumption of all traffic is represented. Our proposed scheme consumes the second smallest power levels, next to Mori's scheme. A unit of power represents the amount of transmission power needed to satisfy the BER requirement of one voice packet with a channel gain of 1.0. When the system load is 0.7, about 10% of the power can be saved, when compared to the No PLR history scheme. This is mainly due to an efficient transmission selection strategy adopted in the proposed algorithm.

5 Conclusion

In this paper, we proposed a novel transmission slot selection algorithm for BD-MC-CDMA systems. By considering delay bound, packet loss ratio and channel diversity altogether, we show that one can find an efficient slot selection criteria, resulting in better QoS provisioning while minimizing total power consumption. Simulation results show that the delay guarantee for real-time traffic can be achieved by the decoupling of traffic types and packet loss ratio can be made under control by the utilizing packet loss ratio history of each session. On the other hand, power minimization can be acquired by efficiently selecting better channel within delay bound. Among all schemes compared in this paper, only the proposed method achieves all the required goals. Our proposed algorithm can be extended to other multi-carrier systems such as OFDM systems.

Acknowledgement

– This work was supported by the Samsung Advanced Institute of Technology (SAIT).
– This work was supported by the Ministry of Information and Communication (MIC) under the Information Technology Research Center (ITRC).

References

1. D. Takeda and H. Atarashi, "Orthogonal multicode OFDM/CDMA system using partial bandwidth transmission," IEICE Trans. Commun., vol. E81-B, no.11, pp. 2183-2190, Nov. 1998.
2. L. Xiao, A. Wang, S. Zhou and Y. Yao, "A dynamic resource scheduling algorithm for OFDM system," IEEE Proc. APCC '03, vol. 2, pp.444-447, Sept. 2003.
3. Y. Zhang and K. Ben. Letaief, "Adaptive resource allocation and scheduling for multiuser packet-based OFDM networks," IEEE Proc. ICC '04, vol. 5, pp.2949-2953, Jun. 2004.
4. K. Mori and H. Kobayashi, "Frequency band and time slot selection scheme for downlink packet communication in cellular band division MC-CDM systems," IEICE Trans. Commun., vol. E87-B, no.5, pp. 1114-1122, May. 2004.
5. D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," IEEE J. Select. Areas Commun., vol. SAC-8, PP. 368-379, Apr. 1990.
6. L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," Proc. ACM SIGCOMM'90, pp.19-29, Sep. 1990.
7. S. Keshav, An Engineering Approach to Computer Networking. Reading, MA: Addison-Wesley, 1997, ch. 9.
8. H. S. Wang and N. Moayeri, "Finite-state Markov channel - a useful model for radio communication channels," IEEE Trans. on Veh. Technol., vol.44, No. 1, pp. 163-171, Feb. 1995.
9. X. Wang, "Wide-band TD-CDMA with minimum-power allocation and rate and BER-scheduling for wireless multimedia networks," IEEE/ACM Trans. Networking, vol. 12, pp. 103-116, Feb. 2004.

Evaluation of QoS Provisioning Capabilities of IEEE 802.11E Wireless LANs

Frank Roijers[1], Hans van den Berg[1,2], Xiang Fan[1,2], and Maria Fleuren[1]

TNO Telecom, Delft, The Netherlands,
Univerisity of Twente,
Department of Design and Analysis of Commmuncation Systems,
Enschede, The Netherlands

Abstract. Several studies in literature have investigated the performance of the proposed IEEE 802.11E standard for QOS differentiation in WLAN, but most of them are limited both with respect to the range of the parameter settings and the considered traffic scenarios. The aim of the present study is to systematically investigate (by simulations) the impact of each of the QOS differentiation parameters, under more realistic traffic conditions. In particular, we investigate flow-level performance characteristics (e.g., file transfer times) in the situation that the number of active stations varies dynamically in time.

1 Introduction

A major drawback of existing versions of the IEEE 802.11 WLAN standards, notably the widely used IEEE 802.11B version, is that they are not capable of providing any service guarantees. The most widely deployed IEEE 802.11B MAC protocol, the so-called Distributed Coordination Function (DCF), is a random access scheme based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Current research and standardization efforts are aiming at enhancements of the DCF MAC protocol enabling the support of multi-media applications with stringent QOS requirements. In particular, the Enhanced Distributed Coordination Function (EDCF) of the IEEE 802.11E standard [8], which is currently being finalized, provides several parameters enabling QOS differentiation among the traffic originating from applications with different service characteristics. Existing studies on the QOS provisioning capabilities of IEEE 802.11E are limited both with respect to the range of the parameter settings and the assumptions about the traffic generated by the WLAN stations/users. Therefore, the aim of the present study is to systematically investigate (by extensive simulations) the impact of the QOS differentiation parameters, under more realistic traffic conditions.

Related Literature

For the 802.11B version several analytical models have been developed in order to study the system's saturation throughput as a function of the number of (persistently) active stations. The most well-known model is the one developed by

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 138–148, 2005.

Bianchi [2]. It is based on a Markov chain describing the behavior of a single station attempting to send its packets. Foh and Zuckerman [4] and Litjens et al. [11] investigate the flow-level performance of 802.11B WLAN when the number of active stations varies in time, e.g. due to the random initiation and completion of file transfers. In particular, building on Bianchi's work [2], they obtain accurate approximations for the mean flow transfer delay.

Most performance studies of the QoS-enabled 802.11E WLAN are based on simulation. Relatively few papers present an analytical approach. E.g., [16] propose extensions of the Markov chain analysis presented in [2] for the 802.11B version, in order to capture the impact of variation of the AIFS parameter (one of the EDCF parameters) on the saturation throughputs. Both analytical models yield accurate results. The simulation studies usually consider more general scenarios (sometimes also capturing the impact of higher layer protocols like TCP), but a systematic study of the impact of each of the EDCF parameters on WLAN performance is lacking. In particular, [3, 6, 10, 12] compare the 802.11B version with the 802.11E version, but only for the default parameter settings; other papers (e.g., [1, 5, 9, 15]) consider a broader range of parameter values but only for some of the QoS differentiation parameters. Most of the studies mentioned above assume a fixed number of persistently active stations. In some cases (see e.g., [1, 3, 5, 6]) the impact of adding one or two additional (persistently active) stations is studied by plotting the throughputs as function of the time. However, flow-level performance studies, which take into account that the number of active stations varies dynamically in time, are not available.

Contribution

Our first contribution is a systematic evaluation of the IEEE 802.11E QoS differentiation parameters (EDCF parameters) CW_{min},CW_{max}, AIFS and the TXOP_{limit} in the situation with persistently active stations. The impact of the parameter setting is studied by simulations of a WLAN scenario with two different service classes. The main performance metrics that are investigated are the throughputs per service class and their ratio.

Our second contribution is a thorough investigation of the EDCF's capabilities to provide QoS guarantees in a (more realistic) scenario with a dynamically varying number of active stations. We consider three different service classes: voice, video and (TCP controled) data traffic. The main performance metrics, studied by simulation, are packet loss, packet delay (particularly important for voice) and data flow (file) transfer time.

Outline

The remaining of this paper is organized as follows. In Section 2 the main principles of the 802.11E MAC protocol and its mechanisms to provide QoS differentiation are explained. In Section 3.2 the simulation scenarios are described both for the packet level study (assuming persistently active stations) and the flow level study. In Section 4 the results of the simulation studies are presented and discussed. Section 5 concludes this paper.

2 IEEE 802.11E QoS Enhanced Wireless LAN

In this section we briefly explain the IEEE 802.11B Distributed Coordination Function (DCF) and its enhancements as specified in IEEE 802.11E ([8]) in order to support QoS differentiation. We concentrate here on the so-called BASIC mode of the DCF which we considered in our performance study.

2.1 IEEE 802.11B Distributed Coordination Function

In BASIC access scheme, when a station wants to transmit a data packet, it first senses the medium to determine whether or not the channel is already in use by another station (*physical carrier sensing*). If the channel is sensed idle for a contiguous period of time called DIFS (Distributed InterFrame Space), the considered station transmits its packet. In case the channel is sensed busy, the station must wait until it becomes idle again and subsequently remains idle for a DIFS period, after which it has to wait another randomly sampled number of time slots before it is permitted to transmit its data packet. This *backoff* period is sampled from a discrete uniform distribution on $\{0, \cdots, \mathrm{CW}_r - 1\}$, with CW_r the contention after r failed packet transfer attempts (CW_0 is the initial contention window size). The backoff counter is decremented from its initially sampled value until the packet is transferred when the counter reaches zero, unless it is temporarily 'frozen' in case the channel is sensed busy before the backoff counter reaches zero. In the latter case the station continues decrementing its backoff counter once the medium is sensed idle for at least a DIFS period.

If the destination station successfully captures the transmitted data packet, it responds by sending an ACK (ACKnowledgment) after a SIFS (Short InterFrame Space) time period. If the source station fails to receive the ACK within a predefined time-out period, the contention window size is doubled unless it has reached its maximum window size CW_{max}, upon which the data packet transfer is reattempted. The total number of transmission attempts is limited to r_{max}. Once the data packet is successfully transferred, the contention window size is reset to CW_0 and the entire procedure is repeated for subsequent data packets.

A station has a finite size InterFace Queue (IFQ) where IP packets, which arrive from higher OSI-layers, have to wait for their turn to content for the medium. IP packets that find the IFQ full upon arrival will be dropped.

2.2 IEEE 802.11E Enhanced Distributed Coordination Function

An 802.11E station (QSTA) deploys multiple Traffic Categories (TCs); traffic is mapped into a particular TC according to its service requirements. Each TC contents, independently of the other TCs, for the medium using the CSMA/CA mechanism decribed in Section 2.1 according to its own set of EDCF parameters values. These EDCF parameters are CW_{min}, CW_{max}, AIFS and the TXOP_{limit}.

The parameters CW_{min} and CW_{max} have the same functionality as in the DCF. The parameter AIFS (Arbitrary InterFrame Space) differentiates the time that each TC has to wait before it is allowed to start contending after the medium

has become free. An AIFS is at least a DIFS period possibly extended by a discrete number of time slots. The TXOP_{limit} (Transmittion OPportunity limit) is the duration of time that a TC may send after it has won the contention, so it may send multiple packets within a TXOP_{limit}.

The backoff counters of the TCs of a particular station can reach zero at the same moment, a so-called *virtual collision*. The highest priority TC may actually put its packets on the medium, the lower priority TCs react as if they experienced a collision, so they have to double their contention window CW_r and start a new contention for the medium, however the parameter r counting the number of attempts is not increased.

3 Description of the Simulation Scenarios

3.1 System Model

We consider a single Basic Service Set (BSS) with stations contending for a shared radio access medium with a channel rate of $r_{\text{WLAN}} = \{1, 11\}$ Mbit/s. The physical layer preamble is always transmitted at 1 Mbit/s and the rate of the MAC layer preambles is $\{1, 2\}$ Mbit/s. All stations are assumed to have comparable radio conditions so that a uniform channel rate can be assumed. Only the BASIC-access mode is considered in the simulations.

The simulations are performed using the Network Simulator NS-2 [13] extended by the EDCF implementation of the TKN Group of the Technical University of Berlin [14]. This implementation contains the differentiation parameters explained in the previous section. Packet capture, which is the possibility that a packet with a strong signal may survive a collision, is turned of in this study.

3.2 Traffic Scenarios

EDCF performance is studied for two main traffic scenarios. In Scenario 1, the impact of the EDCF QoS differentiation parameters is studied assuming persistently active traffic sources (stations). In Scenario 2, the QoS differentiation capabilities of EDCF in the case of non-persistent traffic sources (i.e., a dynamically varying number of active stations) are investigated.

Scenario 1: Persistent Traffic Sources
In Scenario 1 the number of active stations remains fixed during a single simulation. Each station generates traffic in the upstream direction and is assumed to always have traffic available for transmission. The traffic consists of 1500 Byte IP/UDP packets and all data and headers are transmitted at 1 Mbit/s.

We consider two TCs with different 802.11E parameter settings, a high priority class TC_0 and a lower priority class TC_1. In each scenario only the 802.11E parameter under investigation is varied, the other parameters are set according to their 802.11B equivalents. The investigated parameter settings are (802.11B values are denoted in boldface): $\text{CW}_{min} = \{7, 15, \mathbf{31}, 63\}$, $\text{CW}_{max} = \{31, 63, 127, \mathbf{1023}\}$, $\text{AIFS} = \{\mathbf{0}, 1, 2, 5\}$ and $\text{TXOP}_{limit} = \{\mathbf{0}, 0.03, 0.06, 0.1\}$ sec.

Two series of experiments are performed: (i) experiments where the total number of active stations is increased in subsequent simulations, while the mix of active TC_0 and TC_1 stations remains equal (50%-50%), and (ii) experiments where the number of active stations for one of the TCs is increased in subsequent simulations. In both cases the saturation throughput per station of each Traffic Categorie is determined and compared.

Scenario 2: Dynamic User Scenario (Non-persistent Traffic Sources) In Scenario 2, the number of active stations varies dynamically during a simulation due to e.g., the initiation and completion of speech calls or web page downloads. Three different Traffic Categories are considered corresponding with Voice over IP (VoIP, an interactive service), Video-on-Demand (VoD, a streaming service) and Web Browsing (an elastic data service). For each of these services we will consider below the main characteristics and modeling assumptions made in our simulations; specific modeling assumptions are summarized in Table 1.

VoIP is a real-time, interactive service and requires the end-to-end delay to be less than 150 ms. Besides the delay also the packet loss is constrained; it should be less than a few percent. In the simulations new VoIP calls are initiated according to a Poisson process and a VoIP-call is modeled by two UDP CBR streams (80 Kbit/s each).

Table 1. Service Classes

VOICE OVER IP		VIDEO-ON-DEMAND		WEB-BROWSING	
FLOW LEVEL PARAMETERS					
Transport prot.	UDP	Transp. prot.	UDP	Transp. prot.	TCP
Downstream	CBR	Downstream	CBR	Downstream	TCP data
Upstream	CBR	Upstream	-	Upstream	TCP ACKs
Arrival process	Pois. Proc.	Arr. process	ON-OFF	Arr. process	Poiss. Proc.
ON-time distr.	exp.	ON-time distr.	exp.	file size distr.	exp.
avg ON-time	180 sec	avg ON-time	300 sec	avg file size	15 KBytes
Arrival rate	1/60	OFF-time distr	exp.	Arrival rate	4
		ON-OFF ratio	1 : 4		
PACKET LEVEL PARAMETERS					
IP packet size	200 Bytes	IP packet size	1500 Bytes	IP packet size	1500 Bytes
bit rate	80 Kbit/s	bit rate	480 Kbit/s		

VoD traffic is sent at a fixed rate from a video server to a user. The most important QoS-constrained for streaming video is packet loss as video-codecs are very sensitive to loss. Packet delays are less important. In the simulations a VoD traffic stream is modeled by a UDP CBR packet stream (480 Kbit/s). The VoD calls are generated by a fixed number of users; the time between the completion of a VoD call and the initiation of a new call by a particular user is exponentially distributed.

Web-Browsing is controlled by TCP. The most important QoS metric for this application type is the web page download time or, closely related, the through-

put during a web page download. In the simulations web page downloads are initiated according to a Poisson process. Web pages are retrieved from a web server that is connected to the AP by a fixed link with a certain capacity and transmission delay. The capacity is chosen such that it is not a bottleneck and no packets will be lost.

In Scenario 2 the WLAN operates at 11 Mbit/s. Starting with a certain mix of offered traffic (determined by the default parameter settings shown in Table 1) we study the effects of increasing the traffic load due to one service class, while the traffic load due to the other service classes remains unchanged. The performance metrics of interest are packet loss, packet delay and delay jitter and the mean web page download time; because of lack of space we will omit in this paper the performance results for the VoD and WB service classes. In the simulations, the total number of users simultaneously present in the system is at most 50 as new users are blocked when already 50 users are present.

4 Numerical Results

4.1 Scenario 1: Persistent Traffic Sources

This section presents the results of the Persistent Traffic Sources scenario described in Section 3.2. Due to space limitations within this paper, we will present here only the conclusions of this study.

Table 2. System and traffic model parameter settings, based on the DSSS PHY layer

WLAN PHYSICAL		WLAN MAC		FIXED NETWORK	
data rate	11 Mbits/s	MAC overhead	224 bits	delay	10 ms
basic rate	2 Mbits/s	r_{max}	3	capacity	100 Mbit/s
SIFS	10 μs	IFQ length	50 packets	TCP/UDP	
DIFS	50 μs	max # STAs	50	TCP header	20 Bytes
EIFS	304 μs			TCP receiver W_{max}	20 packets
PHY header	48 μs			UDP header	20 packets
PLCP header	144 μs			IP	
Time slot	20 μs			IP header	20 Bytes

The simulations results show the impact of the four QoS differentiation parameters. Each parameter can provide differentiation for a Traffic Categorie and each parameter has its own qualities and a system load where it performs best. Distinguishing three criteria listed below, we can summarize the results of our study as given in Table 3, distinguishing between the effects for a high and a lower number of stations present.

- Differentiating capability: ability to give preference to one Traffic Category over the other.
- Fairness: ability to share the capacity among the TCs as intended, according to an a-priori defined ratio independent of the number of active users.
- Efficiency: ability to achieve a high aggregate throughput.

Table 3. Qualitative assessment of the EDCF differentiating parameters

	Differentiation		Fairness		Capacity	
Number of stations	low	high	low	high	low	high
CW_{min}	++	+	−	++	0	−
CW_{max}	0	++	+	−−	0	−−
AIFS	+	++	0	−	0	++
$TXOP_{limit}$	+	+	++	++	+	++

The results of our study indicate how the QoS parameters can be applied to meet certain QoS requirements. The $TXOP_{limit}$ has perfect capabilities (cf. Table 3) w.r.t. to all above-mentioned criteria. However, a drawback of setting a large value of $TXOP_{limit}$ is the increase in delay and delay jitter. CW_{min} differentiates well, but for high loads the system capacity decreases. AIFS and CW_{max} both differentiate very well and become even more effective in situations with high load; noted that CW_{max} is only used in situations with many retransmissions.

Thus, the 'optimal' choice of the QoS differentiation parameters settings depend on the specific objectives of the operator; e.g. VoIP protection (because of its high delay sensitivity) or some guaranteed througput for Web Browsers of protection of low priority of an operator.

4.2 Scenario 2: Dynamic User Scenario (Non-persistent Traffic)

This section presents the results of the flow level simulations decribed in Section 3.2 and Table 1. The load of 0.17 corresponds to traffic settings of Table 1, which corresponds to an average of 3 VoIP, 2 VoD and 12 WB users. The load is varied by varying the arrival rate of only VoIP. Note that although a 'net' load of 0.17 seems to be light traffic, in fact it is already heavy traffic and the 'gross' load is close to 1. VoIP users have a high gross load caused by inefficient channel usage due to their small packet size. A high number of users also results in a decrease of the channel capacity, so the gross load per user increases.

Performance of 802.11B. First the results of the dynamic scenario over an IEEE 802.11B DCF are presented, so all service classes have the same priority. The left graph of Figure 1 shows that the downstream direction performance metrics are worse than for the upstream direction for all loads. Already for load 0.15 all three downstream performance metrics are above the QoS targets. The downstream direction performs worse as the majority of all traffic is sent downstream via the AP to the stations. The AP becomes the bottleneck, queueing occurs at its IFQ resulting in larger delays and possibly into packet losses.

The right graph shows the transfer times of Web Browsers for the same scenario. An increase of the load results in a higher number of VoIP users in the system and WB users' TCP will adapt to the lower remaining available capacity.

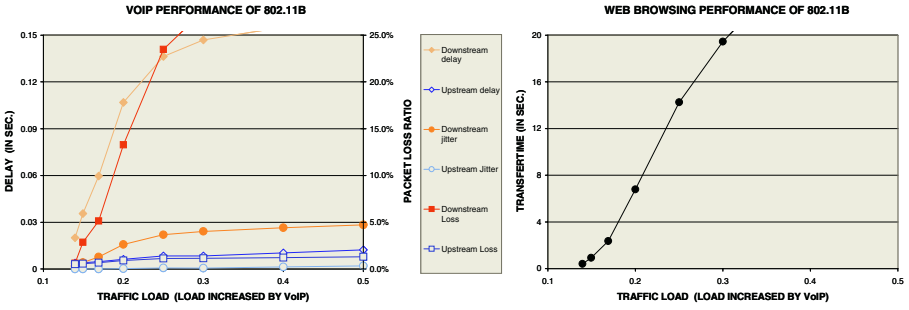


Fig. 1. VoIP and WB performance of 802.11B , load increased by VoIP traffic

Performance of 802.11E (Without Use of $TXOP_{limit}$). IEEE 802.11E EDCF differentiates the service classes by mapping them into different Traffic Categories. VoIP, VoD and WB are mapped into the highest, second highest and lowest priority class respectively. The values of the differentiation parameters per TC,

Table 4. Parameter settings for EDCF TCs. Left: without $TXOP_{limit}$. Right: with $TXOP_{limit}$

	TC_1	TC_2	TC_3
TRAFFIC	VOIP	VOD	WB
CW_{min}	7	15	31
CW_{max}	63	255	1023
AIFS	2	3	4
$TXOP_{limit}$	0	0	0

	TC_0	TC_1	TC_2	TC_3
TRAFFIC	VOIP DOWN	VOIP UP	VOD	WB
CW_{min}	7	7	15	31
CW_{max}	63	63	255	1023
AIFS	2	2	3	4
$TXOP_{limit}$	0.06 sec	0.03 sec	0	0

whose differentiating capabilities are mainly determined by relative differences in their values (e.g. see Section 4.1), are set according to Table 4 (left). To provide high priority for VoIP all parameters have low values. Low CW_{min} provides fast contention and fast retransmissions after a collision in order to fulfill the delay constraints, AIFS also provide fast contention and CW_{max} provide fast retransmissions. The EDCF parameters for WB are chosen equally to 802.11B (except for AIFS) and the VoD parameters are set in between the other Traffic Categories.

Figure 2 illustrates that the downstream packet loss improved tremendously compared to 802.11B (note that the scale of the packet loss axis has changed) and for low loads all the performance metrics are within the requirements. For higher loads ($\rho > 0.27$) the downstream packet loss is above 1%, so still the performance of the downstream direction has to be improved.

The right graph shows that although WB traffic has the lowest priority, compared to the DCF it performs slightly better. The performance improvement is caused by a higher aggregate throughput as the EDCF parameter values of the higher priority classes are smaller than the normal 802.11B DCF parameters.

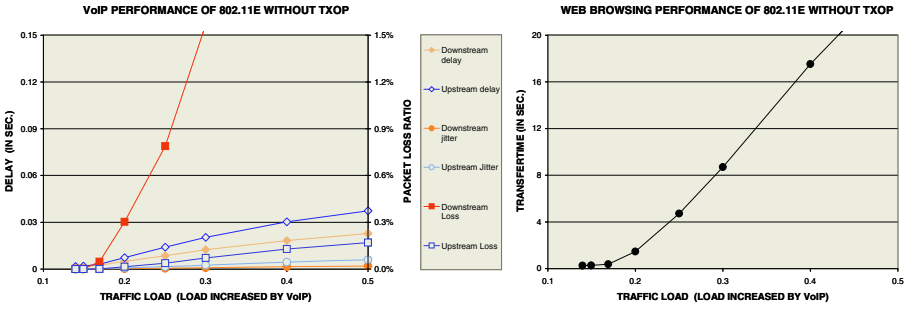


Fig. 2. VoIP and WB performance of 802.11E

Performance of 802.11E (with Large $TXOP_{limit}$ for Downstream VoIP). The downstream direction is the bottleneck as most of the traffic is sent via the AP and the channel efficiency is low due to the small size of the VoIP packets. To improve the performance of the AP an extra TC is added for downstream VoIP (see Table 4 (right)), so the AP is now allowed to send multiple VoIP packets after its VoIP Traffic Category has won a contention.

The left graph of Figure 3 illustrates that the downstream packet loss remains the bottneck as the performance is only slightly improved. The small improvement is caused as the AP wins the same amount of contentions, and if it wins, it is allowed to sent multiple VoIP packets, so its buffer is emptied faster. The delay and delay jitter in the upstream direction perform worse than in figure 2 due to the TXOP_{limit}, but they are still within the requirements. The performance of WB (right graph) is also similar to the previous scenario. Further improvement of VoIP performance metric packet loss can be obtained by enlarging the TXOP_{limit} (even for all TCs), however this will introduce extra delay and jitter. for all TCs.

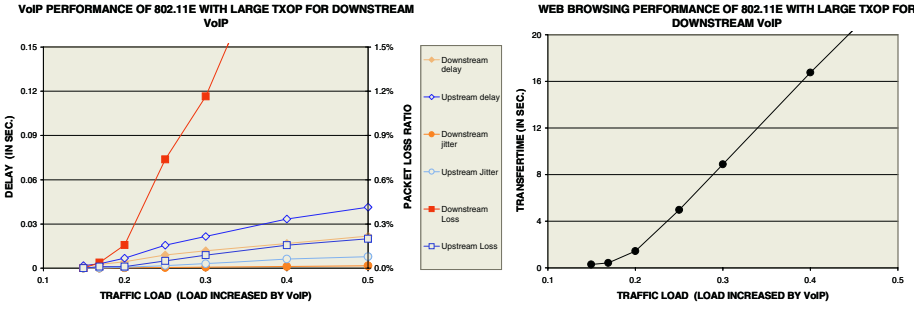


Fig. 3. VoIP and WB performance of 802.11E with large TXOP_{limit} for downstream VoIP

Finally, other simulations (not presented due to lack of space) show that the performance of VoIP, as the highest priority TC, is hardly influenced if the load is increased by WB traffic.

5 Concluding Remarks

In this paper we have studied the EDCF mechanism for QoS provisioning in WLAN. First, extensive simulations of scenarios with persistent users illustrate the QoS differentiating capabilities of the EDCF parameters. The impact of the EDCF parameters is not ambiguous and depends on the system characteristics, e.g. the number and types of users. The results are summarized in Table 3.

Second, we haved studied the impact of the EDCF parameters in a scenario with three service types and dynamic arrivals and departures of users. It is shown that the plain 802.11B is not capable of fulfilling service requirements of interactive services. 802.11E improves the performance, however a major drawback is that the Access Point becomes the bottleneck in the downstream direction. The performance can be improved by implementing extra TCs especially for the AP with preferential treatment.

The EDCF is only capable of providing service differentiation, and not of delivering absolute QoS guarantees. The best approach to attempt to give absolute guarantees is to deploy Call Admission Control (CAC). The results of the present study can be used to determine CAC boundaries on the number of users per type that may be present in the system.

Acknowledgements

This work has been partially funded by the Dutch Ministry of Economics Affairs under the program "Technologische Samenwerking ICT-doorbraakprojecten", project TSIT1025 BEYOND 3G.

References

1. I. Aad and C. Castelluccia, Priorities in WLANs, Computer Networks, vol. 41, no. 4, pp 505-526, 2003.
2. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Comm., vol. 18, no. 3, pp. 535-547, 2000.
3. S. Choi, J. del Prado, S. Shankar N, S. Mangold, IEEE 802.11E Contention-Based Channel Access (EDCF), ICC 2003, Anchorage AL, USA, 2003.
4. C.H. Foh and M. Zukerman, Performance analysis of the IEEE 802.11 MAC protocol, Proceedings of European Wireless '02, Florence, Italy, 2002.
5. P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, X. Cheng, Using IEEE 802.11E MAC for QoS over Wireless, Proceedings of the IPCCC 2003, Phoenix, Arizona, 2003.
6. D. He, Ch. Shen, Simulation Study of IEEE 802.11E EDCF, IST Project Moby Dick.
7. IEEE P802.11B/D7.0, Supplement: higher speed physical layer extension in the 2.4 GHz band, 1999.
8. IEEE P802.11E/D3.0, Draft supplement: enhancements for Quality Of Service , 2002.
9. J. Kim and C. Kim, Performance analysis and evaluation of IEEE 802.11E EDCF, Wireless Communications and Mobile Computing 2004, 4:55-74.
10. A. Lindgren, A. Almquist, O. Schelen, Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LAN, Proceedings of IEEE LCN, pp. 348–351, 2001.

11. R. Litjens, F. Roijers, J. L. van den Berg, R. J. Boucherie and M. J. Fleuren, Performance analysis of wireless LANs: An integrated packet/flow level approach, Proceedings of ITC 18, Berlin, Germany, 2003.
12. S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz and L. Stibor, IEEE 802.11E Wireless LAN for Quality of Service, European Wireless 2002, Florence, Italy, 2002.
13. Network Simulator 2: http://www.isi.edu/nsnam/ns.
14. http://www.tkn.tu-berlin.de/research/802.11e_ns2/.
15. T. Raimondi and M. Davis, Design Rules for a Class-based Differentiatied Service QOS Scheme in IEEE 802.11E Wireless LANs, MSWiM'04, Venice, Italy, 2004.
16. J. Zhao, Z. Guo, Q. Zhang and W. Zhu, Performance study of MAC for service differentiation in IEEE 802.11, Proc. of IEEE GLOBECOM 2002, Taipei, Taiwan, 2002.

Content-Aware Packet-Level Interleaving Method for Video Transmission over Wireless Networks

Jeong-Yong Choi and Jitae Shin

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

Abstract. In the wireless network environment, the effect of transmission errors and losses on the video quality varies depending on the intensity of the burst and which parts of the video stream are lost. Among the existing transmission error control techniques, FEC and ARQ are good solutions for combating transmission errors, but they require redundant data. Although interleaving has no error correcting capability, it can improve subjective video quality without wasting additional bandwidth, because it allows the spreading of successive errors. In this paper, we propose a content-aware packet-level interleaving method, which uses a quantitative index to indicate the degree of content-importance of the video content, so that the effect of burst packet losses is distributed intelligently. The proposed scheme improves the overall video quality in comparison with content-blind interleaving methods.

Keywords: interleaving, content-aware, wireless network, video transmission.

1 Introduction

Two or one way streaming over unreliable and error-prone wireless channel is one of the major challenges for wireless video applications. There are many research efforts in wireless communication area to combat transmission error/loss over wireless channel such as channel coding, modulation, interleaving, etc. Besides these bit-level error control methods, there is also a need for application-aware techniques in order to provide multi-class service for diverse multimedia traffics, e.g., the four classes in Universal Mobile Telecommunications System (UMTS), viz. the conversational, streaming, interactive and background traffics. In addition, transmission error/loss control (TEC) needs to be both application content-aware and wireless channel-aware, therefore necessitating a cross-layer approach, which combines application quality of service (QoS) requests or content priorities and channel condition information.

Classic bit-level TEC in wireless communications can compensate for the time-varying channel effect, but still lacks the ability to accomplish the efficient

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 149–158, 2005.

transmission of diverse multimedia applications over packet-switched networks. On the other hand, packet-level TEC would have the advantage of satisfying the need for application QoS requests and content-aware treatment. There are three main packet-level TEC methods: (1) *packet-level forward error correction (P-FEC)*; (2) *automatic repeat request (ARQ)*; and (3) *packet-level interleaving with packetization (P-interleaving)* [1].

There are trade-offs among these control techniques. FEC requires additional bits, but at the same time it can correct corrupted data without retransmission[2]-[5]. ARQ is inadequate for real-time video streaming because of delay-constraints, but it is more effective than FEC under relatively good channel conditions and loose delay requirements [4]-[7]. P-interleaving can spread burst errors and has no overhead, but causes additional delays associated with packet permutation. A low-delay interleaving method [7] was proposed using a video encoder buffer as part of interleaving memory, and Y. J. Liang et al.[8] determined the optimal interleaver minimizing the expected total distortion of the decoded video, subject to a delay constraint.

There have been few studies of interleaving which take video content into consideration, S. K. Chin et al.[9] proposed a content-aware interleaving method which considers the priority of the base layer and enhancement layer in the layered codec. They attempt to change packet-sending orders by randomizing and interleaving the base layer packets with several enhancement layer packets. However this technique provides only a quite coarse degree of content-aware interleaving, because it operates on only layer-level not packet-level.

In this paper, we propose a more general and finer content-aware P-interleaving method, which regulates burst error effects by spreading out the video packets in accordance with each packet's pre-calculated priority.

In section 2, we review general interleaving methods and describe the problem posed by the P-interleaver in video transmission. In section 3, we describe our own interleaving method, which is designed to solve the above mentioned problems. Lastly, we present the experimental results and further discussions in section 4.

2 General Packet-Level Interleaving

In the burst error-prone wireless network environment, consecutive packet losses happen more frequently, and this causes more serious degradation in the video quality than losses that are spread uniformly, for a similar average packet loss rate. P-interleaving is a method which is widely used to spread spatio-temporally contiguous packet losses. Y. J. Liang et al. [8] proposed an interleaver which minimizes the total distortion, given knowledge of the channel burst loss characteristics and the delay constraint. According to [8], given the channel loss characteristics and the delay constraint $\delta_{\max}$ determine the optimal interleaver $(n_{\text{opt}}, d_{\text{opt}})$, such that the total distortion of the decoded video sequence $D\left[I\left(n, d, K_{\text{orig}}\right)\right]$ is minimized, i.e.,

$$(n_{\text{opt}}, d_{\text{opt}}) = \arg\ \min_{n,d:(n-1)\times(d-1)\leq\delta_{\max}} D\left[I\left(n, d, K_{\text{orig}}\right)\right] \tag{1}$$

where $I(\cdot)$ denotes the functional representation of the interleaver (n, d) indicating the interleaving data size, and K_{orig} denotes the indices of the lost packets before interleaving. Let us consider two methods of the (n, d) interleaver in video transmission, namely temporal interleaving and spatial interleaving.

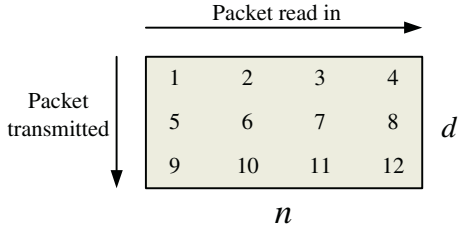


Fig. 1. An example of general (n, d) interleaver $(n = 4, d = 3)$

First, before going into depth, let us consider video sequence, $\boldsymbol{S}$, in Eq.(2)-(4).

$$\boldsymbol{S} = [\boldsymbol{s_0}; \boldsymbol{s_1}; \boldsymbol{s_2}; \ldots; \boldsymbol{s_m}; \boldsymbol{s_{m+1}}; \ldots] \tag{2}$$

where $\boldsymbol{S}$ is a matrix that represents the sequence of video packets, and is a vector that represents the sequence of video packets of the m-th picture, which is expressed as,

$$\boldsymbol{s_m} = [s_{m,0}, s_{m,1}, s_{m,2}, \ldots, s_{m,n}, s_{m,n+1}, \ldots] \tag{3}$$

where $s_{m,n}$ represents the n-th packet of the m-th picture of the video sequence.

Let us assume that burst errors impact on the video sequence $\boldsymbol{S}$ and the temporally interleaved video sequence, $\boldsymbol{S}_T$

$$\boldsymbol{S}_T = I_T\left(n_{\text{pic}}, d_{\text{pic}}, \boldsymbol{S}\right) \tag{4}$$

where $I_T\left(n_{\text{pic}}, d_{\text{pic}}, \boldsymbol{S}\right)$ is the functional representation of the $(n_{\text{pic}}, d_{\text{pic}})$ temporal interleaver on video sequence, $\boldsymbol{S}$.

As shown in Fig. 2, the temporal interleaving process scatters the impact of burst errors, which would have affected two or more consecutive pictures,

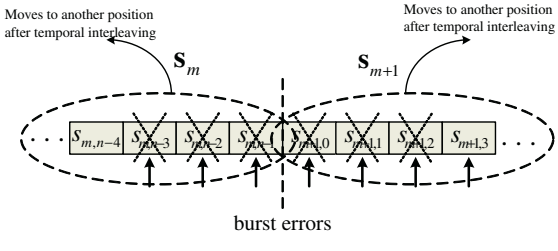


Fig. 2. Temporal interleaving

on non-consecutive pictures by changing the sequence of the pictures. Next, let us consider the corrupted video packets in the picture shown in Fig. 2. Although temporal interleaving scatters burst errors on non-consecutive pictures rather than consecutive pictures, some burst errors still remain in the pictures, as shown in Fig. 3, which result in serious quality degradation. For this reason, additional interleaving of the individual packets making up each picture is required to spread out the burst errors, and this is referred to as spatial interleaving. Although spatial interleaving spreads out burst errors, the number of packet errors within a picture, e.g., s_{m+1}, remains the same.

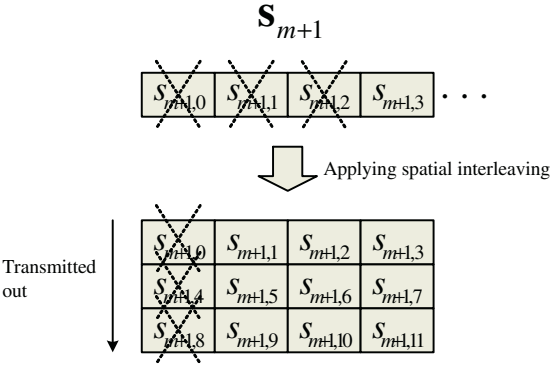


Fig. 3. Spatial interleaving within a picture after applying temporal interleaving

It is known that for the same amount of data lost from the video stream in a given communication channel, the effect on the end-to-end video quality varies considerably depending on the position of the data in the stream [1]. Thus, in the next section, we propose a content-aware packet-level interleaving technique that spreads the priority of the content, thereby resulting in improved protection against burst errors.

3 Proposed Content-Aware Packet-Level Interleaving

3.1 Quantitative Packet-Priority Metric with Video Content-Awareness

For a motion-compensated video coder such as ITU-T H.263 [10], the macroblock (MB)-based corruption can be modeled while taking into consideration the effects of error concealment, the temporal dependency controlled by coding modes and motion vectors, and prediction loop filtering. By assuming that the loss impact of each MB is independent, the impact of one MB loss can be expressed as the sum of the initial error and the propagation error. Also, assuming that the encoder is familiar with the error concealment method at the decoder, the initial error for the MB can be estimated by differentiating to a general error

propagation behavior, where the initial error which propagates to subsequent frames is governed by the effects of the temporal dependency and prediction loop filtering. Under such a scenario and by making use of the results of our previous work [11], we can estimate the total impact of an MB loss in terms of its error energy by

$$\sigma_{MB}^2 = \sigma_u^2 + \sum_{m=1}^{M} \sum_{j=1}^{N} w_{n,j}^2 (m,j)\, \sigma_v^2 (m,j) \tag{5}$$

where σ_u^2 is the initial error and its value depends on the underlying error concealment scheme, $w_{n,j}(m,j)$ and $\sigma_v(m,j)$ stand for the temporal dependency weight and the propagating error impact on the j-th MB(among N MBs) of the m-th frame(among M subsequent frames), respectively. In addition,

$$\sigma_v^2 (m,j) = \frac{\sigma_u^2}{1+\gamma_{m,j}} \tag{6}$$

where $\gamma_{m,j}$ is a parameter called the decaying factor that is governed by the strength of the prediction loop filtering and the frequency characteristic of the initial error. In order to transmit the video stream, it is efficient to packetize based on the synchronization code, where a start code can be inserted into the start of every GOB or slice data in H.263. Thus, we extend the above-mentioned MB-level corruption model, in order for it to be interleaved on GOB level.

Once σ_{MB}^2 is estimated, the GOB-level corruption effect, σ_{GOB}^2, as a packetization unit, can be estimated by averaging σ_{MB}^2 over the number of MBs in the GOB, N_{MB}. The estimated GOB-level relative priority index (RPI)-values (σ_{GOB}^2) are used as a parameter representing the effect on the end-to-end video quality. This RPI is adopted in our content-aware packet-level interleaver that is explained in Section 3.2.

3.2 The Algorithm of Content-Aware Packet-Level Interleaving

In this section, lets assume that only P-interleaving is available because of channel bandwidth constraints and that the maximum allowed transmission delay, $\delta_{\max}$, has been determined. In this case, temporal interleaving is necessary to limit the effect of burst errors on consecutive pictures. Besides temporal interleaving, we also need additional transmission procedures that minimize the degradation of the video quality. The spatial packet interleaving technique mentioned in Section 2, which interleaves the frames within each picture, can provide a solution for burst errors within a picture. However, content-blind interleaving can cause abrupt quality degradation when the most important portion of the packets contained in a burst pattern is lost. In order to solve this problem, we propose a content-aware interleaving technique designed to scatter burst errors intelligently with content-awareness.

Optimal Content-Aware Packet-Level Interleaving. Basically, the packetized video sequence, $\boldsymbol{S}$, is transmitted in the order $\boldsymbol{s_0}, \boldsymbol{s_1}, \boldsymbol{s_2}, \ldots$. The proposed content-aware P-interleaving method consists of 3 steps, i.e., (1) *temporal interleaving*, (2) *spatial interleaving and packetization with RPI*, and (3) *packet transposition*.

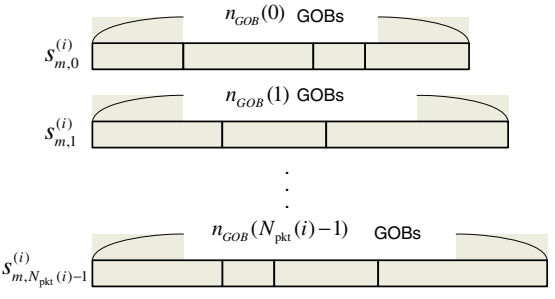


Fig. 4. Example of i-th selection set, $\boldsymbol{s_m^{(i)}}$, with $N_{\text{pkt}}^{(i)}$ packets of arbitrary m-th picture

First, we apply the temporal interleaving technique explained in Section 2, in order to spread the burst errors which would affect two or more consecutive pictures, so that a new picture sequence $\boldsymbol{S}_T$ is obtained. Next, all of the temporally interleaved pictures are spatially interleaved and packetized based on the RPI-value of the GOBs. Denoting N_S the number of possible packetization sets of $\boldsymbol{s_m} = \left\{\boldsymbol{s_m^{(0)}}, \boldsymbol{s_m^{(1)}}, \boldsymbol{s_m^{(2)}}, \cdots, \boldsymbol{s_m^{(N_S-1)}}\right\}$, the optimal content-aware interleaving and packetization is to find the selection set i_{opt}, so that $\boldsymbol{s_m^{(i_{\text{opt}})}}$ minimizes the quality-variation of packet sequence, $\left(\Delta_{\text{pkt}}^{(i)}\right)^2$, as described in Eqs.(7) and (8).

$$\boldsymbol{s_m^{(i_{\text{opt}})}} = \arg\min_{\boldsymbol{s_m^{(i)}}} \left(\Delta_{\text{pkt}}^{(i)}\right)^2 \tag{7}$$

$$\left(\Delta_{\text{pkt}}^{(i)}\right)^2 = \frac{1}{N_{\text{pkt}}^{(i)} - 1} \sum_{j=0}^{N_{\text{pkt}}^{(i)}-1} \sum_{k=0, k\neq j}^{N_{\text{pkt}}^{(i)}-1} \left[\sigma_{\text{pkt}}^2(j) - \sigma_{\text{pkt}}^2(k)\right]^2 \tag{8}$$

where $N_{\text{pkt}}^{(i)}$ is the number of packets in $\boldsymbol{s_m^{(i)}}$, $\sigma_{\text{pkt}}^2(j)$ is the averaged RPI-value of the j-th packet of $\boldsymbol{s_m^{(i)}}$. The values of $\left(\Delta_{\text{pkt}}^{(i)}\right)^2$ are evaluated for N_S number of $\boldsymbol{s_m^{(i)}}$'s in $\boldsymbol{s_m}$ and the optimal solution can be obtained from Eq.(7).

Lastly, once the spatio-temporally interleaved and packetized sequence of video packets, $\boldsymbol{S}_{T,S} = I_S(\boldsymbol{S}_T)$, has been generated, where $I_S(\cdot)$ is the functional representation of the spatial interleaving, all of the packets in $\boldsymbol{S}_{T,S}$ should be transmitted in a manner that minimizes the expected degradation of the video quality. To accomlish this, we propose an additional transmission process, namely, packet transposition. To explain packet transposition, we need to go into

detail about the packet transmission procedure. In the general packet transmission procedure, the spatio-temporally interleaved and packetized sequence, $\boldsymbol{S}_{T,S}$, is transmitted in the order $\boldsymbol{s'_0}, \boldsymbol{s'_1}, \boldsymbol{s'_2}, \ldots, \boldsymbol{s'_m}, \ldots$, where $\boldsymbol{s'_m}$ denotes the m-th picture after temporal interleaving in step 1. When transmitting in this order, the packets in $\boldsymbol{s'_{m+1}}$ should be transmitted only after $\boldsymbol{s'_m}$ has been completely transmitted, in order to limit the impact of burst errors on the picture sequence. To further reduce the impact of the error burst, we reschedule packet sequence $\boldsymbol{S}_{T,S}$ by applying a transpose operation to $\boldsymbol{S}^{(i)}_{\boldsymbol{T,S}}$ in order to find the optimal solution, i.e., the selection set i, $\left(\boldsymbol{S}^{(i)}_{\boldsymbol{T,S}}\right)^T$, which minimizes the effect of the burst error on the end-to-end video quality, by applying the rule contained in Eq.(8) to $\boldsymbol{S}_{T,S}$, thereby producing the packet sequence described in Eq.(9),

$$\left(\boldsymbol{S}^{(i)}_{\boldsymbol{T,S}}\right)^T = \left[s^{(i)}_{0,0}{}', s^{(i)}_{1,0}{}', s^{(i)}_{2,0}{}', \ldots, s^{(i)}_{N_P,0}{}', s^{(i)}_{0,1}{}', s^{(i)}_{1,1}{}', \ldots, s^{(i)}_{N_P,1}{}', s^{(i)}_{2,0}{}', \ldots\right] \quad (9)$$

where $(\cdot)^T$ represents the transpose operation on the matrix, N_P is the number of picture in the interleaving interval and $s'_{m,n}$ is the n-th packet of the m-th picture. Since the above technique requires full scanning of the whole sequence of packets in an interleaving interval to obtain the optimal solution, which is too computationally expensive, we propose a more practical solution in the next subsection.

Practical Content-Aware Packet-Level Interleaving. In this section, we propose a practical content-aware packet-level interleaving algorithm, designed to reduce the computational complexity of the optimal content-aware packet-level interleaving algorithm described in the previous subsection.

In order to simplify step 2 of the 3 steps described in the previous subsection, i.e., the spatial interleaving and packetization step, we consider the packetization unit as a GOB per packet. Of course, this is not the best method of minimizing the variation of the averaged RPI in a packet, however we compensate for this

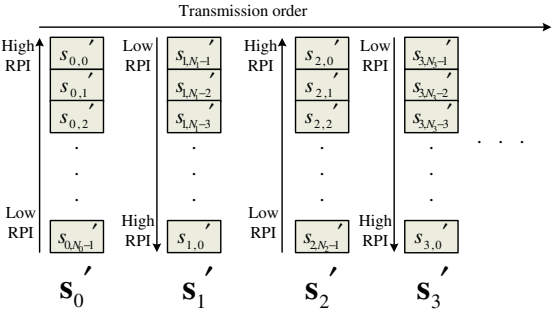


Fig. 5. *Packet transposition process for the spatio-temporally interleaved packet sequence. (*$\boldsymbol{S}_{T,S}$: N_0, N_1, and N_2 are the number of packets for the 0-th, 1st and 2nd picture, respectively.)

shortcoming by means of the packet transposition step. The next step is packet transposition. To implement this step, those packets in a picture with an even-numbered picture number are sorted in descending order of their RPI-value, σ^2_{pkt}, and those packets with an odd-numbered pictures are sorted in ascending order of their RPI-value, and this process is continued for all of the pictures in the interleaving interval. Then the packet sequence of $\boldsymbol{S}_{T,S}$ is transposed to $(\boldsymbol{S}_{T,S})^T$, which is the practical solution of the content-aware packet-level interleaving method.

4 Experimental Results and Discussion

In this section, we demonstrate the performance of the proposed content-aware interleaving scheme, by performing the simulations under the burst error environments (with 4, 6, 8 and 10 consecutive packet errors). A "Foreman" sequence (CIF, 352×288) was encoded using H.263 with the encoding parameters, shown in Table 1, and then transmitted through the burst error-prone channels using four different methods, i.e., (1) *no interleaving (neither temporal nor spatial)*, (2) *temporal interleaving only*, (3) *content-blind spatio-temporal interleaving*, and (4) *content-aware spatio-temporal interleaving.*

Table 1. Video encoder parameters used in the experiment

Parameter settings	
Video encoder	H.263
Sequence name	Foreman
Image format	CIF (352×288)
Number of encoded pictures	300
Encoding method	1 I-picture followed by 299 P-pictures with VBR
Channel setting	
Channel errors	4, 6, 8 and 10 burst packet errors every 15 pictures.

Figs. 6 and 7 show the experimental results for each of these four methods. The experimental result in Fig. 6 shows that the content-blind methods, i.e., (1), (2) and (3), are influenced by the burst errors, so that the PSNR distribution shows abrupt degradations. In contrast, the proposed content-aware method is less susceptible to the burst errors, with the result that the PSNR curve descends slowly and maintains a certain degree of video quality in spite of the severe burst errors.

The three content-blind methods do not consider the variable effect of the burst errors on the quality degradation, by making use of an index of content-awareness such as RPI. Therefore, as shown in Figs. 7(a) and 7(b), the variation and average PSNR performances of these schemes are worse than those of the proposed one. In contrast, the content-aware scheme spreads the packet error

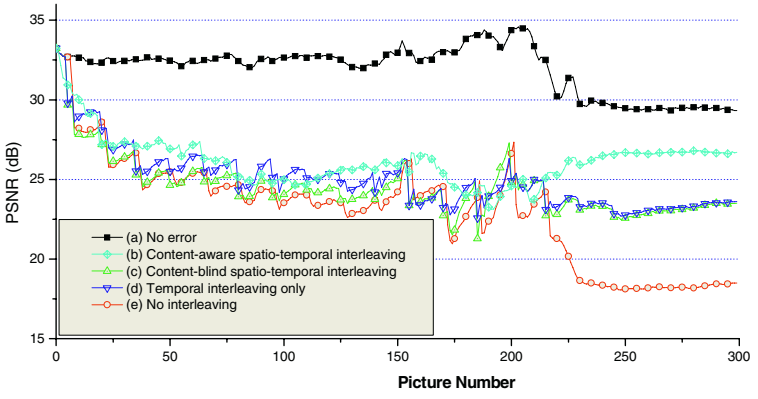
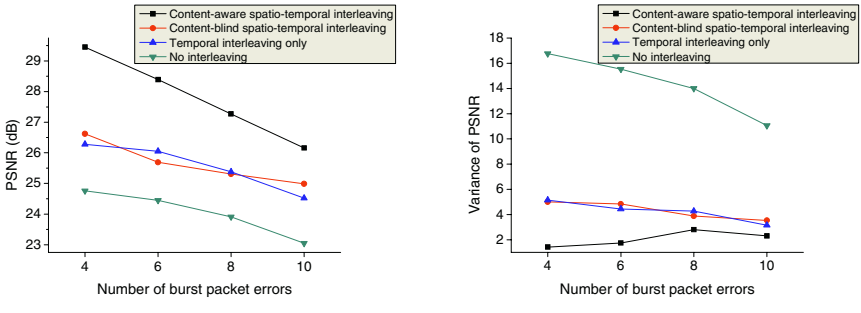


Fig. 6. *PSNR distribution of Y-component of transmitted sequence ((b)-(e): applied 10 burst packet errors with period of 15 pictures)*: (a) No error, (b) Content-aware spatio-temporal interleaving, (c) Content-blind spatio-temporal interleaving, (d) Temporal interleaving only, (e) No interleaving



(a) Average PSNR (b) Variance of PSNR

Fig. 7. The perfermance comparison of the different interleaving methods

bursts by re-ordering the packets spatially, according to the importance of the content of each packet with respect to the picture, and keeps the resultant quality degradation resulting from the burst errors as uniform as possible, through packet transposition. The experimental results show that the proposed scheme provides robust protection against burst errors.

From the experimental results, it can be inferred that the proposed content-aware P-interleaving method shows good performance from the viewpoint of the objective and subjective video quality by reducing the variance of the PSNR. These results provide some insight into why the simple interleaving method is insufficient to improve the overall video quality, in that although it spreads the quality degradation, the average PSNR remains at a similar level.

References

1. B. Girod and N. Färber, "Wireless Video," in A. Reibman, M.-T. Sun (eds.), Compressed Video over Networks, Marcel Dekker, 2000
2. T. Nguyen, and A. Zakhor, "Distributed video streaming with forward error correction," 12th International Packet Video Workshop (PV 2002), Apr. 2002
3. W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, Mar. 2001, pp. 373-386
4. D. Wu, Y. T. Hou, and Y. Q. Zhang, "Transporting real-time video over the Internet : Challenges and approaches," Proceedings of the IEEE, Vol. 88, No.12, Dec. 2000, pp. 1855-1975
5. D. Wu, Y. T. Hou, W. Zhu, Y. Q. 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, Mar. 2001, pp. 282-300
6. N. Guo, and S. D. Morgera, "Frequency-Hopped ARQ for wireless network data services," IEEE Journal on Selected Areas in Communications, Vol. 12, No. 8, Oct. 1994, pp. 1324-1337
7. S. Aramvith, C. W. Lin, S. Roy, and M. T. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 6, Jun. 2002, pp. 558-565
8. Y. J. Liang, J. G. Apostolopoulos and B. Girod, "Model-based delay-distortion optimization for video streaming using packet interleaving," Proc. Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 2002. Invited Paper
9. Suk Kim Chin and Braun, R, "Improving video quality using packet interleaving, randomisation and redundancy," Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on , 14-16 Nov. 2001, pp. 405-413
10. ITU-T, "Recommendation H.263, video coding for low bit rate communication," Feb. 1998
11. J. Shin, J. G. Kim, J. W. Kim and C.-C.J. Kuo, "Dynamic QoS mapping control for streaming video in relative service differentiation networks," European Transactions on Telecommunications, Vol. 12, No. 3, May-June 2001, pp. 217-230

A Performance Model for Admission Control in IEEE 802.16

Eunhyun Kwon[1], Jaiyong Lee[1], Kyunghun Jung[2], and Shihoon Ryu[3]

[1] Dept. of Elec. Engineering, Yonsei University, Republic of Korea
{ehkwon, jyl}@nasla.yonsei.ac.kr
[2] Samsung Electronics Co., LTD., Republic of Korea
kyunghun.jung@samsung.com
[3] Network R&D Center, SK Telecom, Republic of Korea
uchoon@sktelecom.com

Abstract. For systems based on connection-oriented services, such as IEEE 802.16, call admission control (CAC) strategy is essential to provide a desired level of quality of service (QoS). Although many handoff-prioritized CAC schemes, which assume a fixed channel capacity, have been introduced, this assumption is not always valid for IEEE 802.16 that uses adaptive modulation and coding (AMC). With AMC, the modulation type of a user's connection can be changed dynamically and the ongoing connection might fail due to the change of modulation. In this paper, we approach the AMC-induced CAC problem by focusing on the guaranteed connection. Three kinds of calls, new, handoff, and modulation-changed calls, are considered. We propose a modified guard channel CAC scheme that allows the modulation-changed and handoff calls to use the guard channel. Then we analyze a Markov model for the CAC scheme with long-term AMC in mind. According to the simulation results, the proposed approach reduces the call dropping probability for modulation-changed calls, which suggests the threshold of guard channels in IEEE 802.16 can be determined based on the proposed approach.

1 Introduction

IEEE 802.16 has a connection-oriented Medium Access Control (MAC) protocol [1]. In the connection-oriented systems, the CAC mechanism deals with the arrival of a new call. CAC determines whether the system accepts a new connection or not. Before the decision, CAC should confirm that the new call does not degrade the QoS of current connections and the system can provide the QoS requirements of the new call.

Recently many CAC strategies for mobile networks have been studied [2]-[6]. Due to the user mobility, ongoing calls of current cell might be handed over to another cell. However, the receiving cell might have insufficient resources due to the network overload or hostile channel conditions. Therefore if the arrival rate of new or handoff calls exceeds the capacity of a cell, it may start dropping calls or refuse handoff attempts. Since call dropping is generally considered more

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 159–168, 2005.

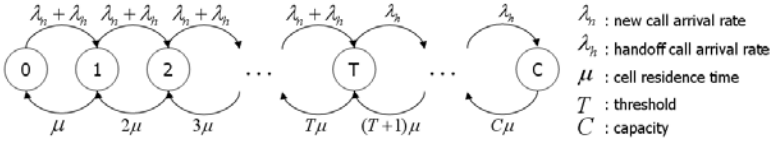


Fig. 1. State transition diagram for handoff-prioritized service

annoying than call blocking, many CAC mechanisms put a higher priority to handoff calls than new calls. The guard channel CAC strategy is one of those schemes that provide handoff-prioritized services for the mobile networks [5], [6]. For PCS networks, various schemes assuming a fixed amount of bandwidth were proposed. In these schemes, some of the bandwidth is exclusively reserved for handoff calls, which enables handoff calls to take precedence over new calls.

Fig. 1 shows a generalized state transition diagram for handoff-prioritized services. Traditional guard channel schemes consider a single cell with a fixed channel capacity (C), and give a higher priority to handoff calls by reserving a portion of the capacity for handoff calls. A new call can be admitted only when the occupied channels are less than a threshold (T). On the other hand, a handoff call is rejected only when all C channels are used up. Although available bandwidth is limited, a handoff call is guaranteed to get enough bandwidth for its objective. In this paper, we analyze the effects of AMC while a CAC process is running. AMC has been proved an effective technique against time-varying channel conditions and was adopted into the physical layer of several standards such as IEEE 802.16. In the next section, we describe the system model for an AMC-based CAC scheme. Analysis and simulation results are presented in Section 3.

2 System Model

The IEEE 802.16 working group on BWA(Broadband Wireless Access) develops the standards and recommended practices for broadband wireless metropolitan area networks [1]. AMC and orthogonal frequency division multiple access (OFDMA) have been adopted in the IEEE 802.16 system.

2.1 Adaptive Modulation in IEEE 802.16

The objective of AMC is to maximize the data rate by adjusting some transmission parameters to available channel information while maintaining a predetermined packet error rate. The transmission parameters can be modulation scheme, channel coding rate etc [7]. When AMC is used in a single cell, the throughput is related to the distance between two nodes. Since multi-path and user mobility inherent in mobile networks lead to fading and the change of distance, the modulation method needs to be changed to maintain the necessary packet error rate for the service.

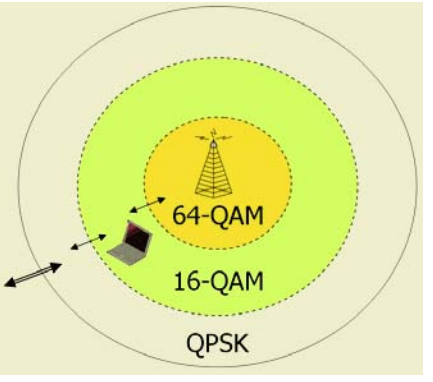


Fig. 2. Cell organization with adaptive modulation

If every connection is assumed to maintain the same bandwidth requirement, a connection using a high-order modulation, e.g, 64QAM requires less resource than those with lower-order modulations. Even if there is neither new call request nor handoff call request, change of modulation scheme may degrade QoS or even lose the connection. Therefore, a simple model such as the one in Fig. 1 fails to reflect the transmission situations faithfully, which requires the incorporation of modulation type into the CAC process. Although AMC schemes are typically designed exclusively for the physical layer, their influences will reach higher layers [8]. Our design and analysis will cover such a multi-layered aspects of AMC embedded in a CAC system.

The rates of handoff and modulation change are related to the user's velocity and the cell size he/she belongs to. To find the modulation change rate, we have to determine the cell organization like Fig. 2. A comparison of three modulation schemes is presented at table 1 and its parameters are specified in [9].

Table 1. A comparison of different transmission modes

Scheme	Spectral efficiency (b/s/Hz)	Relative coverage (%)	Relative link margin (dB)
QPSK	1.5	100	0
16-QAM	3	49	-9
64-QAM	4.5	23	-17

2.2 Markov Model for a CAC System Using AMC

In the wireless AMC systems, each subscriber is assigned a different transmission mode that can be changed dynamically due to the user mobility. We consider the effects of long-term modulation changes, which have not received enough research interest in the literature of classical guard-channel CAC schemes.

To model a CAC system consisting of a single cell, analysis is carried out under the following assumptions. We consider two kinds of modulations and only

guaranteed services which need the same amount of bandwidth. Three types of call requests, handoff call, new call, and modulation changed call, are modeled as independent Poisson processes. Channel holding time is assumed to be exponentially distributed. ($\mu_1 = \mu_{d1} + \mu_{h1}$, $\mu_2 = \mu_{d2} + \mu_{h2}$)

- C : The total number of mini slots as a cell capacity
- s_1, s_2 : The number of mini slots which is needed to a connection with modulation type 1 and type2, respectively
- λ_{n1}, λ_{n2} : Arrival rate of new calls with modulation type 1 and type 2, respectively
- λ_{h1}, λ_{h2} : Arrival rate of handoff calls with modulation type 1 and type 2, respectively
- $1/\mu_{d1}$, $1/\mu_{d2}$: Average cell residence time in a cell until disconnection with modulation type 1 and type 2, respectively
- $1/\mu_{h1}$, $1/\mu_{h2}$: Average cell residence time in a cell until handoff with modulation type 1 and type 2, respectively
- $1/\mu_{m1}$: Average channel holding time of a call in a cell until changing modulation from type 1 to type 2
- $1/\mu_{m2}$: Average channel holding time of a call in a cell until changing modulation from type 2 to type 1

In our guard-channel scheme, a modulation changed call takes precedence over a new call as well as a handoff call. Let g represent the proportion of the guard channel to the total bandwidth. We consider a bufferless system and a new call can only be admitted when current bandwidth usage is less than $C(1-g)$. The system reserves $g \cdot C$ of available channels as guard channels. When the resource is not enough during the process of handoff or modulation change, the call is dropped rather than degrading QoS.

In Fig. 3, shown is a state transition diagram for the number of calls in the AMC system. Each state can be represented by k_1 ,k_2, the number of calls for each modulation type. Then the state space can be denoted as $E = \{(k_1, k_2)\,|0 \leq k_1, 0 \leq k_2, 0 \leq k_1 \cdot s_1 + k_2 \cdot s_2 \leq C\}$ where k_1 represents the number of users using modulation type 1, and k_2 for modulation type 2.

From Fig. 3, we obtain the following transition rates. Let $q(k_1, k_2 : k_1', k_2')$ be the probability of translation from state (k_1, k_2) to state (k_1', k_2'). The amount of free bandwidth for a new call and a handoff call in state (k_1, k_2) can be calculated as

$$f_n(k_1, k_2) = C(1-g) - (k_1 s_1 + k_2 s_2) \tag{1}$$

$$f_h(k_1, k_2) = C - (k_1 s_1 + k_2 s_2) \tag{2}$$

Then we have

$q(k_1, k_2 : k_1 + 1, k_2) = \lambda_{n1} \cdot 1_{f_n(k_1+1,k_2)\geq 0} + \lambda_{h1} \cdot 1_{f_h(k_1+1,k_2)\geq 0}$
$q(k_1, k_2 : k_1, k_2 + 1) = \lambda_{n2} \cdot 1_{f_n(k_1,k_2+1)\geq 0} + \lambda_{h2} \cdot 1_{f_h(k_1,k_2+1)\geq 0}$
$q(k_1, k_2 : k_1 - 1, k_2) = \{k_1\mu_1 + k_1\mu_{m1} \cdot 1_{f_h(k_1-1,k_2+1)<0}\} \cdot 1_{k_1-1\geq 0}$
$q(k_1, k_2 : k_1, k_2 - 1) = \{k_2\mu_2 + k_2\mu_{m2} \cdot 1_{f_h(k_1+1,k_2-1)<0}\} \cdot 1_{k_2-1\geq 0}$
$q(k_1, k_2 : k_1 + 1, k_2 - 1) = k_2\mu_{m2} \cdot 1_{k_2-1\geq 0} \cdot 1_{f_h(k_1+1,k_2-1)\geq 0}$
$q(k_1, k_2 : k_1 - 1, k_2 + 1) = k_1\mu_{m1} \cdot 1_{k_1-1\geq 0} \cdot 1_{f_h(k_1-1,k_2+1)\geq 0}$

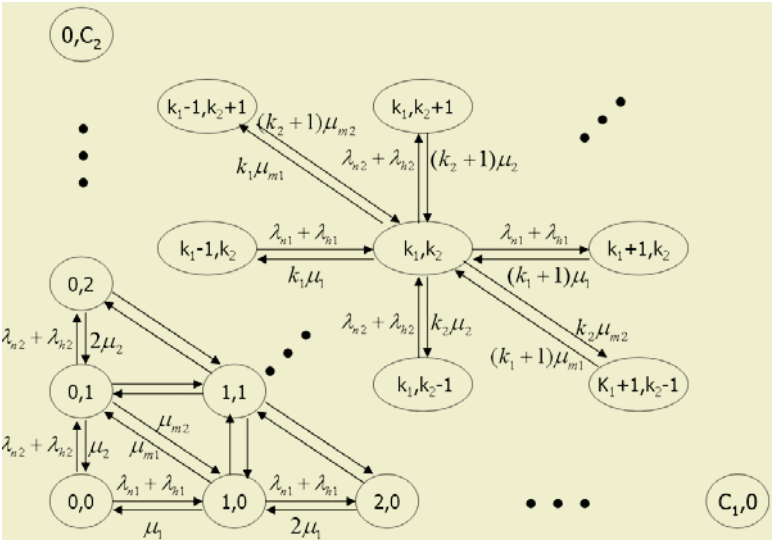


Fig. 3. Markov chain representation for the AMC system

where (k_1, k_2) is a feasible state in E, and $1_{true} = 1$ when the expression is true, otherwise $1_{false} = 0$. The $q(k_1, k_2 : k_1 - 1, k_2)$ and $q(k_1, k_2 : k_1, k_2 - 1)$ have the dropping transition due to the failure of modulation change.

Let $\pi(k_1, k_2)$ be the steady state probability for state (k_1, k_2). Applying the above transition rate, the balance equation for state (k_1, k_2) can be obtained by equating the flux out of a state to that into a state.

$$
\begin{aligned}
- \; & [\lambda_{n1} \cdot 1_{f_n(k_1+1,k_2)\geq 0} + \lambda_{h1} \cdot 1_{f_h(k_1+1,k_2)\geq 0} \\
& + \lambda_{n2} \cdot 1_{f_n(k_1,k_2+1)\geq 0} + \lambda_{h2} \cdot 1_{f_h(k_1,k_2+1)\geq 0} \\
& + \{k_1\mu_1 + k_1\mu_{m1} \cdot 1_{f_h(k_1-1,k_2+1)<0}\} \cdot 1_{k_1-1\geq 0} \\
& + \{k_2\mu_2 + k_2\mu_{m2} \cdot 1_{f_h(k_1+1,k_2-1)<0}\} \cdot 1_{k_2-1\geq 0} \\
& + k_2\mu_{m2} \cdot 1_{k_2-1\geq 0, f_h(k_1+1,k_2-1)\geq 0} \\
& + k_1\mu_{m1} \cdot 1_{k_1-1\geq 0, f_h(k_1-1,k_2+1)\geq 0}] \cdot \pi(k_1, k_2) \\
+ \; & \{\lambda_{n1} \cdot 1_{f_n(k_1,k_2)\geq 0} + \lambda_{h1}\} \cdot 1_{k_1-1\geq 0} \cdot \pi(k_1 - 1, k_2) \\
+ \; & \{\lambda_{n2} \cdot 1_{f_n(k_1,k_2)\geq 0} + \lambda_{h2}\} \cdot 1_{k_2-1\geq 0} \cdot \pi(k_1, k_2 - 1) \\
+ \; & \{(k_1+1)\mu_1 + (k_1+1)\mu_{m1} \cdot 1_{f_h(k_1,k_2+1)<0}\} \cdot 1_{f_h(k_1+1,k_2)\geq 0} \cdot \pi(k_1 + 1, k_2) \\
+ \; & \{(k_2+1)\mu_2 + (k_2+1)\mu_{m2} \cdot 1_{f_h(k_1+1,k_2)<0}\} \cdot 1_{f_h(k_1,k_2+1)\geq 0} \cdot \pi(k_1, k_2 + 1) \\
+ \; & (k_2+1)\mu_{m2} \cdot 1_{k_1-1\geq 0, f_h(k_1-1,k_2+1)\geq 0} \cdot \pi(k_1 - 1, k_2 + 1) \\
+ \; & (k_1+1)\mu_{m1} \cdot 1_{k_2-1\geq 0, f_h(k_1+1,k_2-1)\geq 0} \cdot \pi(k_1 + 1, k_2 - 1) = 0
\end{aligned}
\tag{3}
$$

From [10], the global balance equation is represented as $Q \cdot \overrightarrow{\pi} = 0$, and normalization equation as $\sum_{k_1,k_2 \in E} \pi(k_1, k_2) = 1$. The Q matrix can be generated by the

above equation, and the steady state probability at each state can be also computed, which can be used for obtaining the blocking probability of a new call, a handoff call, and a modulation changed call respectively. The performance of the CAC scheme can be evaluated from the blocking probability of each type of call.

3 Numerical Result

We employ two modulations, QPSK and 16QAM. At the beginning of cell planning, the network provider may organize the area initially for QPSK and later move to 16QAM, 64QAM for higher efficiency. If the subscribers are uniformly distributed, then new call arrival rate is proportional to the size of the area. Let $\lambda_{n1} + \lambda_{n2}$ denote the new call arrival rate, ranging from 0.1 to 0.5. In a fluid flow model [11], [12], the handoff arrival rate can be expressed by $\lambda_h = \frac{\alpha\rho VL}{\pi}$, where L is the perimeter of the cell. $\alpha\rho$ is a active population density and V is the average user's velocity. The handoff departure rate can be represented by $\mu_h = \frac{16V}{\pi L}$. According to this mobility model, the handoff arrival rate is proportional to the perimeter, to which the handoff departure rate is inversely proportional. Suppose that λ_{h1} is a relative value to $\lambda_{n1} + \lambda_{n2}$, assuming that V and L are constant. λ_{h2} is set to 0 assuming no handoff arrival with high modulation types. The cell residence time for each modulation and call closure type is assumed to be exponentially distributed with mean 200 sec. Then μ_1 and μ_2 can be 1/100 1/200 respectively, also assuming no handoff departure with high modulation types. The modulation change rate can be estimated from its radius and handoff rate. μ_{m1} is related with handoff arrival rate λ_{h1}, and μ_{m2} is connected to handoff departure rate μ_{h1}.

Since the best-effort service can be always admitted without regard to the system resources, we consider the CAC operation only for the guaranteed service. We divide the bandwidth allocated for the guaranteed service into 64 blocks of time slots using OFDM or TDMA. Applying the same channel coding rate for each modulation type, the bandwidth assignment for the connection with modulation type 1 can be two blocks, and that for connection with modulation type 2 can be one block. In this case, there can be a total of 32 guaranteed connections using only QPSK modulation.

From the Markov chain on Fig. 3, we generate the state transition matrix and compute the steady state probability. From the steady state probability obtained, we calculate the call blocking probability for each call type considering the arrival rate in each state. Let P_b be the probability that an arriving call is blocked, and P_b can be calculated as

$$P_b = \sum_{j_1, j_2 \in B} \frac{\lambda \cdot \pi(j_1, j_2)}{\lambda \cdot \sum_{k_1, k_2 \in E} \pi(k_1, k_2)} \tag{4}$$

E is the total state space and B is the bounded area where arriving call will be blocked.

To verify the analytical results, 100 hours of simulations on the proposed AMC system, in which only the CAC operation in a single cell was implemented,

have been performed. A set of simulations was repeated 20 times, each time with a different random number generator seed. The analytic and 95% confidence interval simulation results are shown in Figs. 4-7. Note that in all figures, the call intensity is $\rho = \frac{\lambda_{n1}+\lambda_{n2}}{\mu_{d1}+\mu_{d2}}$, which varies from about 5 to 50.

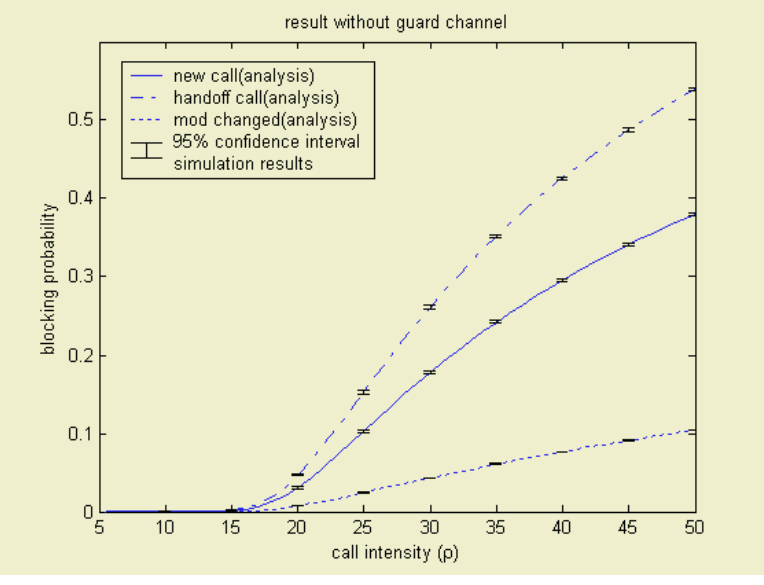


Fig. 4. Blocking probability versus call intensity without guard channels

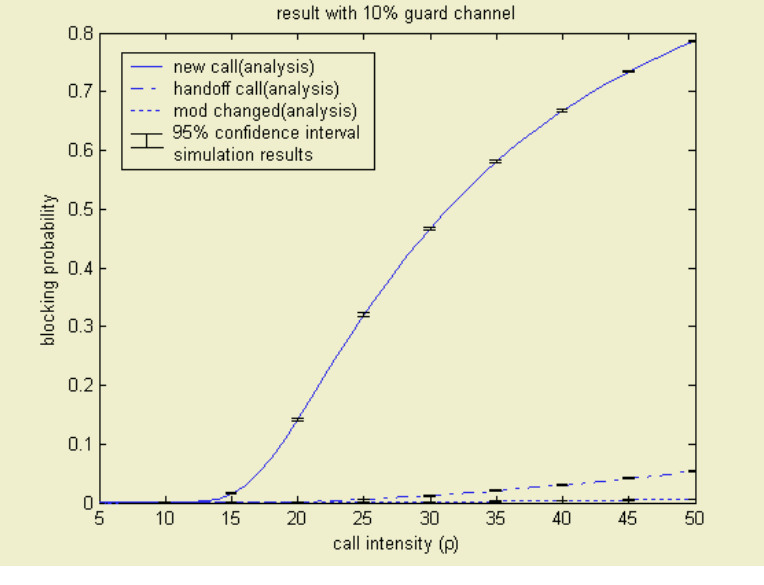


Fig. 5. Blocking probability versus call intensity with 10% guard channels

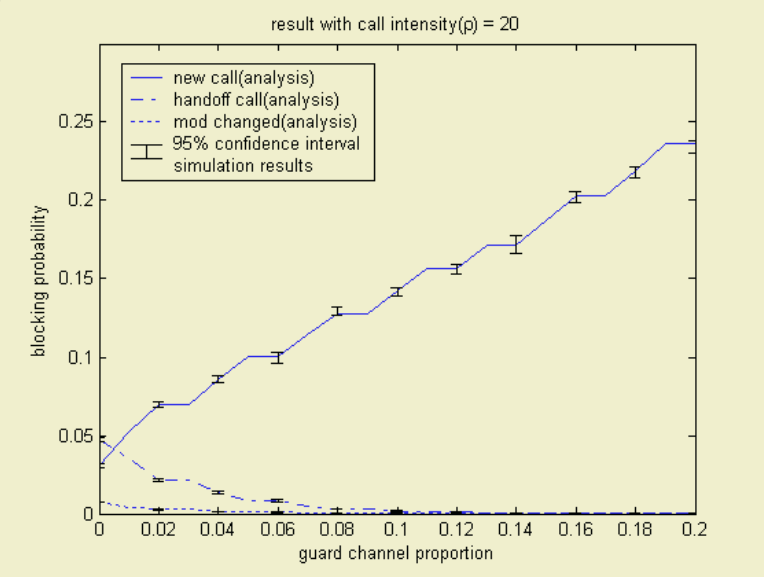


Fig. 6. Blocking probability versus the proportion of guard channels

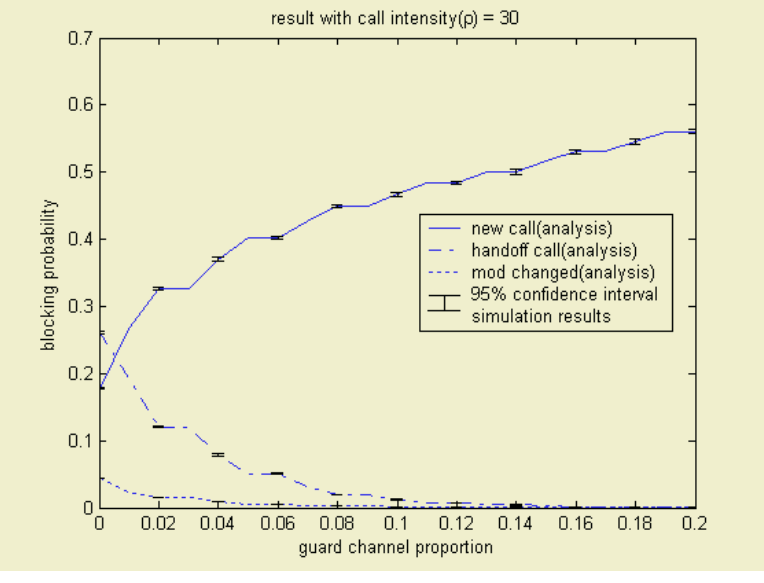


Fig. 7. Blocking probability versus the proportion of guard channels

Figs. 4 and 5 depict the blocking probability for each call type versus call intensity under different CAC schemes. In Fig. 4, shown is the blocking probability without priority, assuming no handoff calls for high modulations. Therefore the blocking probability of handoff calls is higher than that of a new call. Fig. 5 com-

pares the blocking probability in the case of using 10% guard channels. From the figure, it is apparent that the call blocking probability increases with call intensity. The difference between Fig. 4 and Fig. 5 confirms that guard channels can help to reduce the blocking probabilities of handoff calls and modulation changed calls. The blocking probabilities of handoff calls and modulation changed calls can be reduced at the cost of the bandwidth limitation for new calls.

In Figs. 6 and 7, shown is the blocking probability for each call type versus the relative guard channel, under different call intensities. It is observed that when the proportion of the guard channel to the total bandwidth is increased, the blocking probabilities of handoff calls and modulation changed calls are decreased. We can compute the call blocking probability of each call type with given parameters. Therefore, it can be concluded that this model is effective in deciding appropriate thresholds of guard channels for an AMC system.

4 Conclusion

CAC schemes are essential to support required QoS in the wireless mobile networks. In this paper, we investigated the AMC-induced call dropping at a CAC process. We analyzed the guard channel CAC scheme with focus on three aspects: new call arrival rates, handoff call arrival rates, and modulation changed rates. We described a Markov model for the CAC scheme under the influence of long-term AMC. Simulation results show that as the proportion of guard channels increases, blocking probabilities of handoff calls and modulation changed calls decrease. This improvement was achieved by limiting the bandwidth for new calls. In setting up of IEEE 802.16 systems, an appropriate number of guard channels should be selected to maintain the call dropping ratio close to the intended level. Our model and analysis can be a guide to configure those AMC-based wireless networks. For future work, the analysis and simulation in this paper should be confirmed with real IEEE 802.16 networks. We will have to classify the connections further with various QoS parameters such as traffic class, delay, maximum rate, guaranteed rate, and packet loss rate. For systems with more than two modulation types, it is necessary to extend the analysis to additional dimensions.

References

[1] IEEE Standard, "IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std 802.16-2004, June 2004.

[2] Yuguang Fang and Yi Zhang, "Call Admission Control Schemes and Performance Analysis in Wireless Mobile Networks," *IEEE Transaction on Vehicular Technology*, vol. 51, Mar. 2002.

[3] Carlos Oliveria, Jaime Bae Kim, Tatsuya Suda, "An Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, Aug. 1998.

[4] Dervis Z. Deniz and Nagla O. Mohamed, "Performance of CAC Strategies for Multimedia Traffic in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, Dec. 2003.
[5] Daehyoung Hong and Stephen S. Rappaport, "Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Non-prioritized Handoff Procedures," *IEEE Transaction on Vehicular Technology*, vol.35, Aug. 1986.
[6] R. Ramjee, D. Towsley, and R. Nagarajan, "On optimal call admission control in cellular networks," *Wireless Networks*, vol.3, 1997.
[7] Bernard Sklar, "Digital Communications: Fundamentals and Applications, Second Edition," *Prentice Hall*, 2001.
[8] Qingwen Liu, Shengli Zhou, Georgios B. Giannakis, "Cross-Layer Combining of Adaptive Modulation and Coding With Truncated ARQ Over Wireless Links," *IEEE Transaction on Wireless Communications*, vol.3, Sep. 2004.
[9] Bernard Fong, Nirwan Ansari, A. C. M. Fong, G. Y. Hong, Predrag B. Rapajic, "On the scalability of Fixed broadband Wireless Access Network Deployment," *IEEE Communicatin Magazine*, vol.42, No.9, Sep. 2004.
[10] Leonard Kleinrock, "Queueuing Systems, Volume I; Theory," *Wiley Interscience Publication*, 1975.
[11] Timothy X Brown and Seshadri Mohan, "Mobility Management for Personal Communications Systems," *IEEE Transaction on Vehicular Technology*, vol.46, May 1997.
[12] Wei Wu, Archan Misra, Sajal K Das, Subir Das, "Scalable QoS Provisioning for Intra-Domain Mobility," *Globecomm*, 2003.

Comparison of Incentive-Based Cooperation Strategies for Hybrid Networks

Attila Weyland, Thomas Staub, and Torsten Braun

University of Bern,
Institute of Computer Science and Applied Mathematics,
Neubrückstrasse 10, 3012 Bern, Switzerland
{weyland, staub, braun}@iam.unibe.ch

Abstract. Today's public Wireless LANs are restricted to hotspots. With the current technology, providers can only target a small audience and in turn charge high prices for their service to generate revenue. Also, providers can not react appropriate to dynamic changes of the demand. With multi-hop cellular networks the coverage area can be increased and the installation costs and investment risks for the provider can be reduced. However, the individual customers play an important role in such networks and their participation must be encouraged. Therefore, we propose a cooperation and accounting scheme which introduces monetary rewards. We compare our scheme called CASHnet with the Nuglet scheme using simulations under the criteria of network liveliness as well as goodput, overhead and packet error rate.

1 Introduction

The current wireless network installations consists of a number of access points deployed in selected areas, where they are expected to serve a minimum amount of customers to bring revenue to the provider, e.g. at airports or railway stations. Potential customers outside the area covered by the access point can not be served. Besides the financial risk limiting the deployment of access points, location properties can also be restricting factors.

With multi-hop cellular networks, also called hybrid networks, the single-hop limit does not exist any more. Customers act as packet forwarders (like in mobile ad hoc networks) and a gateway offers the connection to the the Internet. This gives the provider a greater coverage area with more customers and reduces the network installation costs. Customers get connectivity outside hotspot areas and can reduce their energy consumption due to shorter next-hop distances. The advantages of mobile ad hoc networks come together with the disadvantages such as maintaining accurate route information, protecting customers from attacks as well as the need to encourage cooperation among customers for keeping the network alive.

Although individual customers may have a common interest in obtaining connectivity, customers tend to prioritize their self-generated packets over packets to be forwarded from other customers when energy is regarded as precious, limited good. Thus cooperation among selfish individuals (customers) can not be

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 169–180, 2005.

taken for granted, but can either be enforced or made attractive. We believe that in civilian applications without a single authority, enforcement is not attractive to individual customers.

Early work enforced cooperation by not allowing any non-cooperative participants [1] or by threat of punishment in case of non-cooperative behavior [2]. In [3] rewards have been introduced as incentive for cooperation in mobile ad hoc networks. The authors of [4] and [5] extended this notion to the multi-hop cellular network environment. They both heavily rely on centralized accounting and security mechanisms.

In a previous publication [6], we proposed a scheme called CASHnet (Cooperation and Accounting Strategy in Hybrid Networks), which allows selfishness, but at the same time makes cooperation a rewarding alternative. We took a highly decentralized approach for the accounting as well as for the security architecture. Accounting is done on the device and authentication is based on public key cryptography. We also allow cost sharing between sender and receiver located in different subnetworks. Each of them pays an amount related to their respective distance to the gateway. Distance related charges generate revenue at the location in the network where the expenses occur. Because all intermediate customers only participate if they get rewarded, longer distances with more intermediate customers on the path to the gateway imply more rewards and thus raise the cost for obtaining the service at a distant location. The alternative would be for the provider to install a new hotspot at the distant location, with all the financial risks implied or for the customer to have no service at all.

In this paper we compare our proposal with the Nuglet [1] scheme in terms of liveliness of the network and overall performance. We find that even with a single, centered Service Station in the network, CASHnet performs better than Nuglet. With an increasing number of Service Stations, the goodput in CASHnet is much higher compared to Nuglet. We also analyze the generated overhead, the packet drop reasons and discuss further improvements to CASHnet.

The rest of the paper is structured as follows. In section 2 we present and compare the Nuglet and the CASHnet schemes. Section 3 describes our evaluation process. In Section 4 we discuss the results we obtained. We conclude our paper and give an outlook in Section 5.

2 Cooperation Schemes

In the introduction we presented some of the available cooperation schemes in the literature and we motivated our approach. The following two sections describe the CASHnet and the Nuglet scheme, which will be compared through simulations later on. From the available cooperation schemes we chose the Nuglet approach because it is - like CASHnet - a decentralized approach with similar requirements and therefore easier to compare. In the description of the schemes we focus on the aspects important for this comparison. More detailed explanations can be found in the given references.

2.1 CASHnet

The CASHnet charging and rewarding mechanism works as follows: Every time a node (customer) wants to transmit a self-generated packet, it has to pay with *Traffic Credits*. The amount is related to the distance in hop counts to the gateway. Every time a node forwards a packet, it gets *Helper Credits*. Traffic Credits can be bought for real money or traded for Helper Credits at Service Stations. A Service Station is similar to a low-cost terminal for loading prepaid cards and has a secure, low-bandwidth connection to the provider, which is used for authentication and payment operations.

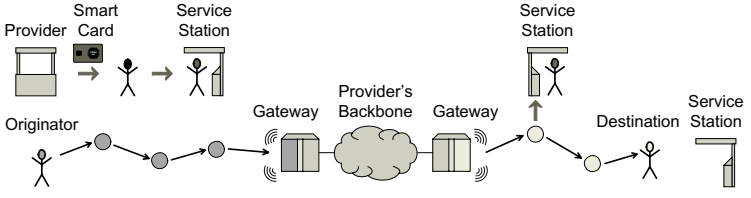


Fig. 1. CASHnet example scenario

Figure 1 displays an example scenario of the CASHnet scheme in operation. After preparation a node is ready to execute the different operation phases illustrated as flow graphs in Figure 2. Suppose a node called *Originator* wants to communicate with another node called *Destination* located in a different subnetwork. First, the *Originator* obtains a smart card from the provider (Preparation). Then it authenticates to all nodes along the path to the destination (Authentication Message Generation, Reception & Forwarding Phases). Now it can start to transmit self-generated packets (Packet Generation Phase). The *Originator* will only pay for the distance to the gateway (in hop counts) of its subnetwork and the *Destination* will pay for the distance to its corresponding gateway. An intermediate node gets rewarded (Rewarding Phase), after the forwarded packet reaches the next hop along the path toward the destination (Packet Reception Phase, Packet Forwarding Phase). All packets are digitally signed and verified upon reception to ensure non-repudiation, i.e. data integrity and data origin authentication. For a more detailed description of our scheme we refer to [6].

Preparation: Node N obtains a personal smart card from the provider X with an unique identifier ID_N, a public/private key pair K_N/KP_N, a certificate $Cert_X(ID_N, K_N)$ issued by provider X for N and the provider's public key K_X. It then performs the following steps:

- load Traffic Credits account TCA at provider's Service Station by paying with real money and/or by transferring from Helper Credits account HCA (as necessary) and
- update certificate $Cert_X(ID_N, K_N)$ (as necessary)

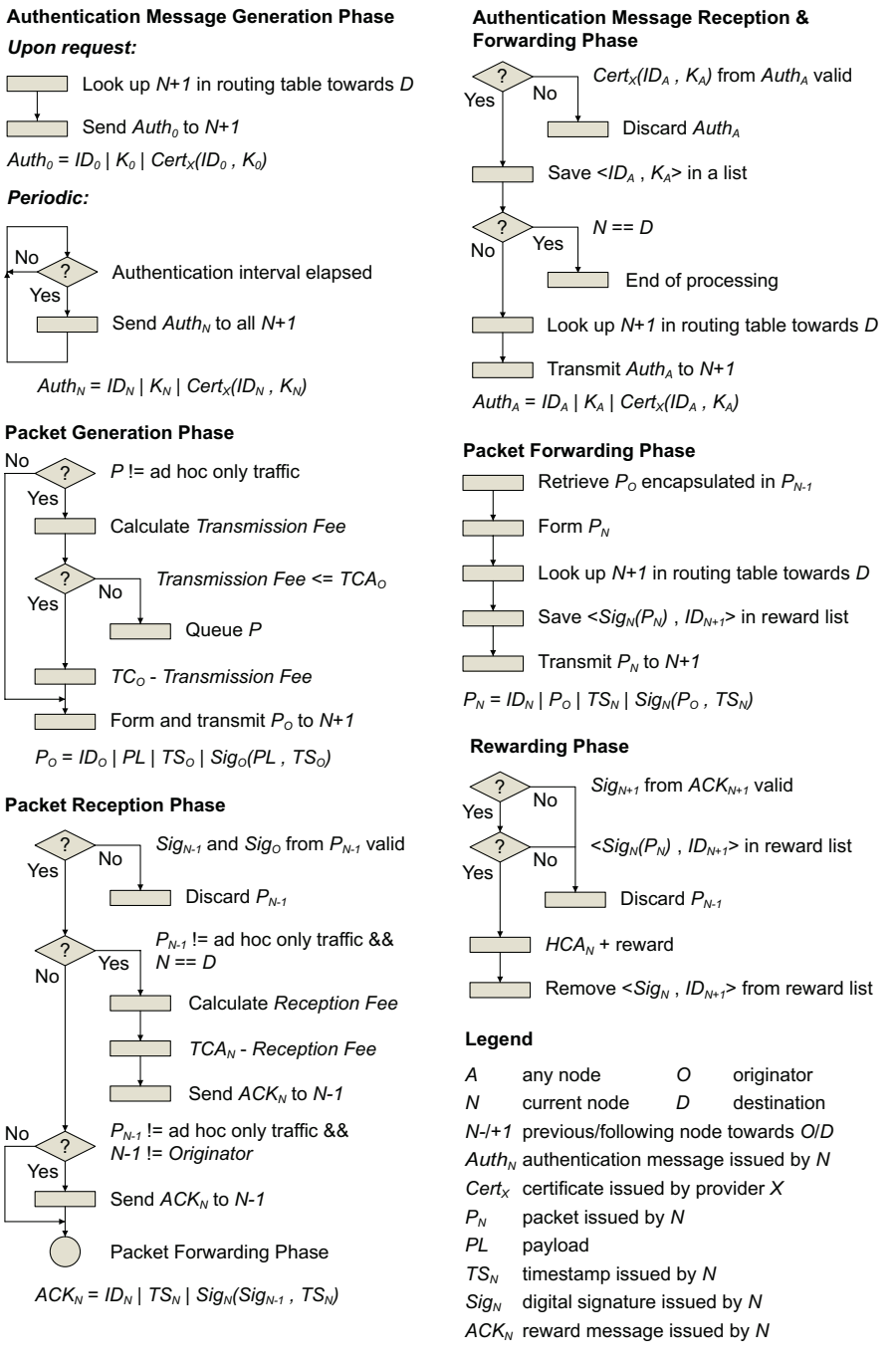


Fig. 2. CASHnet operation in detail

2.2 Nuglet

The Nuglet [1] cooperation scheme has the following main principle: Every time a node wants to transmit a self-generated packet, it has to pay with *Nuglets*. The amount corresponds to the estimated number of nodes between the sender and receiver (intermediate nodes). Every time a node forwards a packet it receives one *Nuglet*.

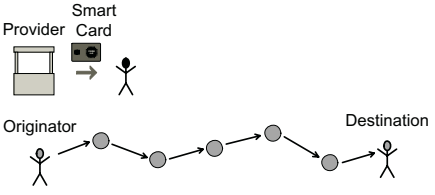


Fig. 3. Nuglet example scenario

Figure 3 shows an example scenario for the Nuglet scheme. As the Nuglet scheme was designed for ad hoc only networks, *Originator* and *Destination* reside in the same network. Also here, the node obtains a smart card. The authentication is done on a node-to-node session basis as the packet travels to the destination. The *Originator* will pay for the estimated number of intermediate nodes located on the path toward the destination. Each node stores the rewards for nodes, from which it has previously received forwarded packets. A synchronization protocol runs in a periodic interval to transfer all pending rewards to reachable nodes. A detailed description of the Nuglet scheme can be found in [1].

2.3 Comparison

Both schemes rely on tamper resistant hardware and public key cryptography. Although the two schemes were targeted at different networks, they both follow a decentralized design pattern. The two schemes charge for the transmission of self-generated packets. In CASHnet the cost is related to the hop count to the gateway, in Nuglet to the number of intermediate nodes to the destination. If a node has not enough virtual money (Traffic Credits or Nuglets), it is not allowed to transmit its own packets. Both schemes stimulate the cooperation among nodes (forwarding packets) through rewards. A node, which forwards a packet receives 1 or more Helper Credits or 1 Nuglet respectively. In the Nuglet scheme, a node can only earn its right for transmission, i.e. it must forward enough packets to be able to send its own packets. The CASHnet scheme additionally allows a node to buy its right for transmission, using additional infrastructure in the network (i.e. Service Stations). Another difference lies in the distribution of rewards. In the Nuglet scheme, each node collects rewards for nodes from which it has received a forwarded packet. In a periodic interval these pending counters are synchronized, in a way that all so far collected Nuglet are transmitted to the

current reachable nodes. The remuneration in CASHnet happens immediately after a node receives a forwarded packet such that it sends an ACK message to the previous hop.

Because our current implementation of the two schemes does not yet include the cryptographic functionality, the security mechanisms are left out from the evaluation. In Nuglet each pair of communicating nodes establishes a symmetric key session to reduce the computational overhead. CASHnet only uses public key cryptography. The high mobility in ad hoc networks is a disadvantage for the session establishment in Nuglet, whereas the power constraints of mobile devices might be a minor disadvantage for the public key operations in CASHnet.

3 Simulation Scenarios

We evaluate both schemes through simulations where we measure the amount and frequency of starving events in the network, i.e. nodes that can not transmit, because they run out of virtual money (Traffic Credits or Nuglets) and use this as an indicator for the liveliness of the network. Also we give results on the overall packet delivery ratio as well as generated overhead and packet drop reasons. We adjusted our schemes' parameters to match the Nuglet scheme to provide a solid foundation for the comparative evaluations.

For the simulation we use ns-2 [7], where we implemented a version of the Nuglet and the CASHnet scheme including the charging and rewarding functionality and leaving out the security mechanisms. In particular, we used the wireless and mobility extensions [8] with an extended version of the AODV protocol called AODV+ [9], which adds Internet gateway discovery support.

Figure 4 shows our simulation scenarios. We only consider a single multi-hop cellular network to be compatible with the Nuglet schemes, which is targeted at mobile ad hoc networks. All nodes in the network send their packets to the gateway. The simulation scenario for Nuglet differs from the CASHnet scenario by removing the Service Stations and replacing the gateway by a normal mobile node.

Table 1 lists the parameters for the simulations. Except for the scheme specific properties, all parameters are identical. Within an area of 1500m x 800m we deploy 40 nodes. The nodes move according the random waypoint model us-

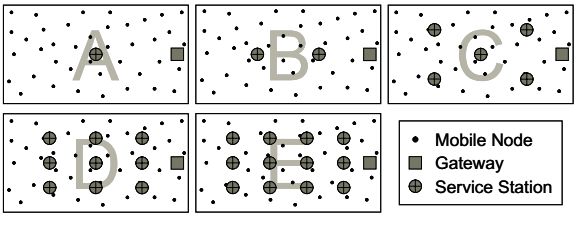


Fig. 4. Simulation Scenarios

Table 1. Simulation parameters

Parameter	Value	
	Nuglet	CASHnet
Space	1500 m x 800 m	
Number of nodes	40	
Transmission range	250 m	
Mobility model	random waypoint	
Speed	uniformly distributed between 1 and 10 m/s	
Pause time	uniformly distributed between 0 and 20 s	
Packet generation rate	0.1, 0.2, 0.5 or 1.0 pkt/s	
Routing	AODV	
Simulation time	900 s	
Initial virtual money account state	100 Nuglets	100 Traffic Credits
Initial real money account state	—	500
Nuglet synchronization interval	5 s	—
Traffic/Helper Credits exchange rate	—	1:1
Exchange thresh. at Service Stations	—	10 Helper Credits
Distance thresh. to Service Stations	—	50 m
Number of Service Stations	—	1, 2, 5, 9 or 12

ing pre-generated movements files. We vary the number and the distribution of deployed Service Stations (none for Nuglet and 1, 2, 5, 9 and 12 for CASHnet) as shown in Fig. 4 as well as the packet generation interval at the CBR traffic sources (1, 2, 5 and 10 s). In total we investigate 24 (6 x 4) simulation scenarios and for each of the scenarios we conduct 20 simulation runs using 20 independent movement files.

In both schemes, the amount of initial virtual money is set to 100. To reflect the ability of a CASHnet node, to buy its right for transmission from available Service Stations, each node also has a real money account initially set to 500. Real money does not exist in the Nuglet scheme and is not equal to virtual money, as it must be exchanged first. Therefore, we believe the comparison to be fair in a sense that both schemes have the same initial situation according to their abilities. In the Nuglet scheme, a node needs to find other nodes and forward their packets to earn Nuglets. In the CASHnet scheme, a node needs to find a Service Station to exchange the Helper Credits and the real money against Traffic Credits. The exchange threshold defines the minimum amount of Helper Credits necessary before a node exchanges them into Traffic Credits at the Service Station. The distance threshold specifies the maximum distance between a node and a Service Stations to be able to exchange the Helper Credits. We measured the frequency of occurrence of starving events and the duration of the starvation. In addition we analyzed the overall goodput, the generated overhead and the reasons for dropped packets.

4 Simulation Results

First we investigate the starvation properties of both schemes. Then we discuss the overall protocol performance. Figure 5, 7, 8, 9 and 10 combine the mean results over the 20 simulation runs from all 24 scenarios. Each label marking on the x-axis consists of two lines. The first line indicates the number of Service Stations used (1, 2, 5, 9, 12 for CASHnet and 0 for Nuglet). The second line lists the packet generation interval (1, 2, 5, 10 s). The four packet generation intervals are separated by vertical lines.

4.1 Starvation

With starvation we describe the nodes inability to transmit self-generated packets due to lack of virtual money (Traffic Credits or Nuglets). Figure 5 contains the average starvation length for a node. Considering the total simulation time of 900 seconds, the two schemes perform poorly under high network load, with CASHnet being a little better. We find that CASHnet performs quite well under low network load, much in contrast to Nuglet.

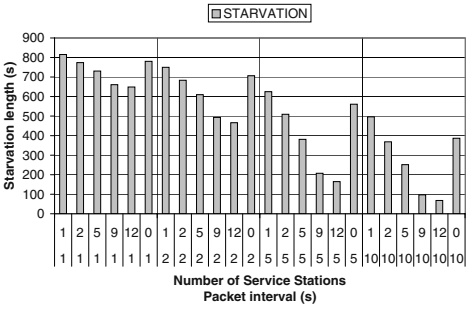
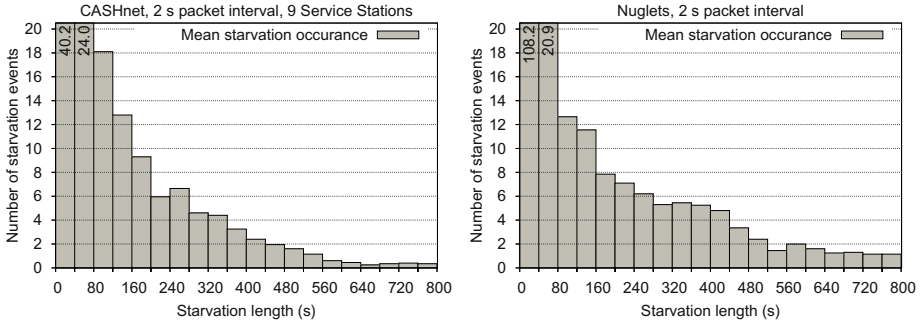


Fig. 5. Starvation length of a node in CASHnet & Nuglet

Both schemes charge for sending self-generated packets and reward for forwarding packets. In Nuglet, a node has only one source of income for virtual money (Nuglets): it has to forward packets from other nodes. In CASHnet, a node has two sources of income for virtual money (Traffic Credits): it can exchange the virtual money earned while forwarding packets from other nodes (Helper Credits) or pay with real money. We find that a self-perpetuating cycle of virtual money, which is assumed by the Nuglet scheme, is difficult to achieve. In such a cycle, each node always receives enough virtual money to be able to transmit self-generated packets. Under low network load (packet interval 10 s), a node in Nuglet starves in average for 43% of the simulation time, whereas in CASHnet the average starvation length is only 8% of the simulation time in the same scenario. This shows, that a node can not cover the cost of sending its own packets solely by forwarding packets from other nodes.

CASHnet performs worse than Nuglet in scenarios with 1 Service Station. In CASHnet, Traffic Credits can only be obtained at Service Stations, whereas in Nuglet only one virtual currency exists and is distributed directly to reachable



(a) CASHnet (b) Nuglet

Fig. 6. Mean number of starvation events per duration category

nodes. When deploying 1 Service Station, the simulation area can not be covered sufficiently in a way that most of the nodes get enough opportunities to fill their Traffic Credits account.

To see the actual distribution of the starvation events, we categorized the events according to their lengths. Figure 6(a) and 6(b) show the average distributions of 20 simulation runs for the scenario with a packet interval of 2 s and 9 Service Stations for CASHnet and Nuglet respectively. We see that the average number of nodes starving for the complete simulation time is rather low.

4.2 Goodput, Overhead and Packet Drop Reasons

Figure 7 shows the goodput. We define goodput as the number of received packets divided by the number of sent packets. The goodput in CASHnet is worse or equal to Nuglet in scenarios with 1 and 2 Service Stations and better with 5, 9 and 12 Service Stations. CASHnet performs considerably better than Nuglet under high packet generation rate. CASHnet provides a 88% increase in goodput compared to Nuglet with a packet interval of 1 s and 12 Service Stations. Under low network load the improvement for CASHnet is lower. 39% increase in goodput with a packet interval of 10 s and 12 Service Stations. However, the goodput is very low for both schemes, for different reasons, which we analyze in the following paragraphs.

In Figure 8 the outcome of the sent packets is shown. We distinguish between received packets, packets dropped because of lack of virtual money and packets dropped for other reasons. The other packet drop reasons will be discussed afterwards. We see that while the number of Service Stations increases, the number of received packets follows as expected, but at the same time the number of packets dropped for other reasons increases too. While we can increase the initial virtual money account on the nodes to reduce the drops caused by lack of money, we have to consider the overhead introduced by the CASHnet scheme. In

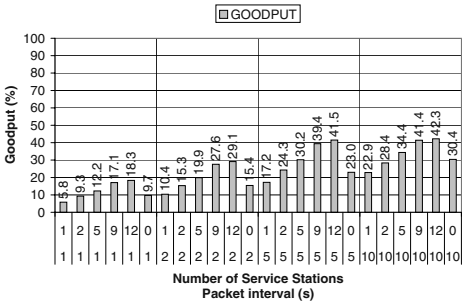


Fig. 7. Goodput for CASHnet & Nuglet

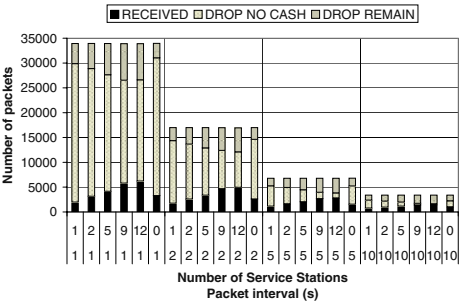


Fig. 8. Packet sent outcome for CASHnet & Nuglet

CASHnet we see, that increasing the number of Service Stations does not automatically decrease the number of packets dropped due to lack of Traffic Credits. This behavior can be observed when we compare the results for 9 and 12 Service Stations for a packet interval of 1 s.

Figure 9 illustrates the actual overhead introduced by both schemes. It shows the amount of packets sent and received as well as reward messages (CASHnet ACK/Nuglet SYNC). In CASHnet, the overhead is much higher than in Nuglet because every packet is rewarded immediately on a per-hop basis by an ACK packet, which increases the nodes Helper Credits account. In Nuglet, each node collects the rewards for its neighbors in personal accounts and transfers this virtual money periodically to all reachable nodes. A former neighbor node, that is not reachable at the time of the synchronization looses all its earned virtual money on that node. When comparing scenario B (2 Service Stations) with scenario C (5 Service Stations) from Figure 4, we see that increasing the number of Service Stations not automatically increases the overhead in CASHnet, and at the same time helps to increase the number of received packets.

The different reasons for dropped packets are displayed in Figure 10. We retrieved the following drop events from the trace files: lack of virtual money (NO

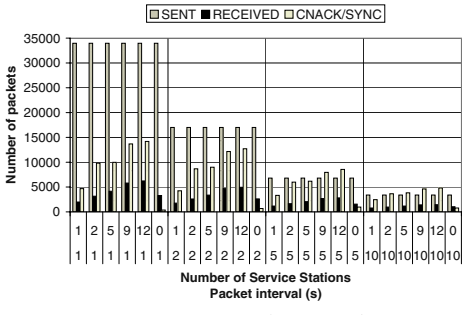


Fig. 9. Overhead for CASHnet & Nuglet

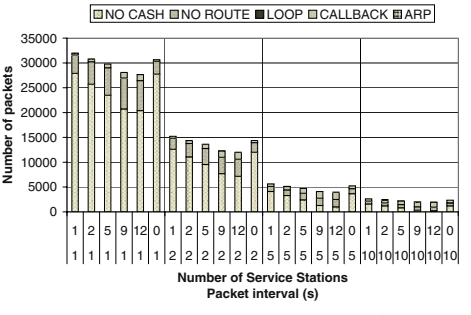


Fig. 10. Packet drop reasons for CASHnet & Nuglet

CASH), no available route (NO ROUTE), routing loop (LOOP), MAC layer callback timer (CALLBACK) and delay in ARP (ARP). However, only NO CASH, NO ROUTE and CALLBACK events have considerable impact. LOOP and ARP events occur very rarely and their frequency does not change much between the different scenarios. In both schemes, the main reason for packet drops is the lack of virtual money. In Nuglet, it is difficult to generate enough traffic to build up a self-perpetuating cycle of virtual money. In CASHnet, a node has the possibility to buy its right for transmission. However, due to the dependency on the fixed Service Stations as the only place for obtaining Traffic Credits, the positive effect of having two sources of income for virtual money is reduced. The second major packet drop reason is the unavailability of a route. In the simulation runs we used an extended version of AODV. When a route is not available in AODV, a route request is send and the packet will be retained until the route requested succeeds or times out. In CASHnet, we suspect the high protocol overhead to be reasons for the high number of unsuccessful route requests.

5 Summary and Outlook

We presented CASHnet, our cooperation and accounting strategy for hybrid networks, which uses a highly decentralized accounting and security architecture. It allows selfish nodes and supports cost sharing between sender and receivers located in different subnetworks. To put the performance of our scheme in context with other work in this area, we compared the CASHnet with the Nuglet scheme, which was also explained in this paper. We implemented both schemes in ns-2 and evaluated them through simulation runs. We monitored the network liveliness and the overall network performance.

As a result from the evaluation we see that the goal of the Nuglet scheme, that is a self-perpetuating cycle of virtual money, is difficult to achieve. We find that in CASHnet nodes already starve less than in Nuglet with only 2 Service Stations deployed. However, the high protocol overhead of CASHnet weakens the positive effect of an additional source of income for virtual money.

For CASHnet we see room for improvement in the granularity of the charging and rewarding mechanisms. This would help to reduce the overhead. In our current approach, we use Service Stations as low-bandwidth, low-cost terminals for buying and exchanging virtual money (similar to loading a prepaid card). We are currently investigating the possibility of changing the role of the Service Station. The simple combination of gateway and Service Station is more expensive and therefore might pose a risk for the provider. It also might not be possible to install gateways at certain locations. However, to keep the multi-hop cellular network alive, the nodes need possibilities (Service Stations) to fill their Traffic Credits account.

Using other mobility models with more realistic user behavior and adapting the deployment of Service Stations accordingly could also greatly improve our schemes performance. Additionally, the generic behavior of the customers them-

selves could be made more realistic, e.g. when running out of virtual money, the movement direction changes to the closest Service Station.

Further work will include more extensive simulation runs to determine the amount and location of Service Stations required for minimal packet loss as well as the implementation of a prototype of our CASHnet scheme using SmartCards. We will analyze our security mechanisms in terms of effectiveness against different attack types and resource consumption. Also, we will study possible extensions to our scheme and optimize the relation between charging and remuneration.

References

1. Buttyán, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. ACM Mobile Networks & Applications **8** (2003) 579–592
2. Buchegger, S., Boudec, J.Y.L.: Performance analysis of the confidant protocol (cooperation of nodes - fairness in dynamic ad-hoc networks). In: Proceedings of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Lausanne, Switzerland (2002) 226–236
3. Zhong, S., Chen, J., Yang, Y.R.: Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: Proceedings of 22nd IEEE INFOCOM. Volume 3., San Francisco, CA, USA (2003) 1987–1997
4. Salem, N.B., Buttyán, L., Hubaux, J.P., Jakobsson, M.: A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In: Proceedings of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, USA (2003) 13–24
5. Lamparter, B., Paul, K., Westhoff, D.: Charging support for ad hoc stub networks. Elsevier Journal of Computer Communications **26** (2003) 1504–1514
6. Weyland, A., Braun, T.: Cooperation and Accounting Strategy for Multi-hop Cellular Networks. In: Proceedings of 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), Mill Valley, CA, USA (2004) 193–198
7. Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., Yu, H.: Advances in network simulation. IEEE Computer **33** (2000) 59–67
8. 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. In: Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Dallas, TX, USA (1998) 85–97
9. Hamidian, A.: A study of internet connectivity for mobile ad hoc networks in ns 2. Master's thesis, Lund University, Sweden (2003)

Analysis of Decentralized Resource and Service Discovery Mechanisms in Wireless Multi-hop Networks

Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt,
and Piet Demeester

Department of Information Technology (INTEC), Ghent University - IMEC vzw,
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
{jeroen.hoebeke, ingrid.moerman, bart.dhoedt,
piet.demeester}@intec.UGent.be

Abstract. The last few years, research in wireless multi-hop networks is mainly driven by a search for efficient routing protocols. From an application point of view, nodes (users) will only setup connections with a specific goal, i.e. in order to use services and resources available in or reachable through the ad hoc network. Consequently, resource and service discovery (R&SD) protocols that allow nodes to learn available services in the network are indispensable. In this paper we compare the performance of two basic decentralized R&SD techniques, proactive and reactive discovery, through simulations and theoretical analysis. Our results show that the choice between them is not straightforward. It highly depends on the network and service characteristics and on the interaction with the underlying routing protocols. Therefore, our analysis provides some guidelines for developing new or extending existing R&SD protocols for operation in mobile ad hoc networks.

1 Introduction

Mobile ad hoc networks are self-organizing mobile, wireless networks that do not rely on any fixed infrastructure for their operation and have some salient characteristics [1]. During the last few years, a lot of research efforts were, and still are, focused on the development of efficient routing protocols, as establishing connections between nodes is one of the primary functions the network has to perform. From a higher level, it is clear that nodes will only setup connections with a specific goal, i.e. in order to use services and resources that are available in or reachable through the ad hoc network. Possible services or resources include data storage, database access, files, network printer, Internet gateway… Therefore, R&SD protocols, which allow nodes to automatically locate available resources and services or to advertise their own capabilities, are also major component of mobile ad hoc networks, which operates in close relation with the routing protocol. This article discusses the advantages of decentralized solutions and presents a comparison - both through simulations and analytically - of two decentralized R&SD techniques, namely reactive discovery, in which nodes request a service when needed, and proactive discovery, in which nodes periodically announce their services.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 181–191, 2005.

2 Related Work

R&SD architectures can be dichotomized into centralized and decentralized architectures [2]. In centralized architectures, nodes register their services with service brokers or service agents. When a node needs a service, a request is sent to a broker, which sends back a reply message containing the requested information. The use of agents or brokers improves the scalability, reduces the response time and can be used for load balancing. Most existing resource and service discovery protocols such as the Service Location Protocol, Jini, Universal Plug and Play and Salutation protocol basically rely on directories for their operation, although some of them are able to function without central agents [3].

Another approach is the use of a decentralized architecture that uses reactive discovery or proactive discovery. When using proactive R&SD, nodes periodically announce the services they offer by broadcasting service announcements (SANN) in the network. On reception of these SANNs, nodes extract the information from which they learn about the available services in the network and forward the SANNs to their neighboring nodes. During reactive R&SD, nodes that need a service send out service requests (SREQ), which are propagated in the network. Nodes offering services actively listen for such messages. If they receive a SREQ for a service they support, a reply message (SREP) is generated and sent back to the requesting node. The Bluetooth Service Discovery Protocol is an example of a decentralized architecture specifically designed for small-scale Bluetooth networks.

3 General Discussion of Centralized and Decentralized R&SD Techniques

Management Overhead. A centralized R&SD protocol requires that one or multiple nodes are assigned the task of collecting the service information, exchanging this information amongst each other, keeping it up-to-date and providing service information to nodes that request services. Deploying a centralized solution in an ad hoc network environment can create considerable overhead in managing the central agent(s). In such an environment, no dedicated central agents are present. This implies that a distributed agent selection algorithm is needed to select the most appropriate device that can take up the role of service agent. As the devices can be mobile, can leave and join the network, and are battery-powered, service agents need to be reselected from time to time and their information needs to be transferred. In addition, all nodes in the environment need to become aware of these changes. In a decentralized R&SD solution, whether proactive or reactive, no management overhead is present, as each node independently decides on the actions taken.

Scalability. In centralized solutions most R&SD control traffic will be unicast (or multicast), whereas in decentralized solutions broadcasting is mainly used. The unicast (multicast) will put a lower burden on the wireless multi-hop network. In addition, in centralized solutions all control traffic is directed to the central agents and

does not propagate throughout the entire network. Finally, using central agents allows better scalability in terms of the number of services that can be handled.

Resilience. With centralized R&SD protocols, resource and service discovery functionality entirely relies on central agents. This means that these agents can form a single point of failure. In dynamic wireless network environments where nodes are mobile and battery-powered or agent functionality can be reassigned, resilience can become an issue. Therefore, a decentralized solution can provide an alternative or backup method, as functionality is guaranteed at all times.

Network Load. In a centralized solution, all services are registered at a single location. This implies that the load on this location will increase strongly when the number of nodes that need services increases. In an ad hoc network this could result in an unfair network load in specific parts of the network. In decentralized solutions, the network load will be more equally spread over the network. Of course, the exact trade-off will strongly depend on the R&SD patterns in the networks.

Latency. In centralized solutions, the latency to find a requested service depends on the time to forward the request to the central agents and the time to receive a reply. Of course, this latency will mainly consist of routing and forwarding delays. In decentralized solutions, the latency to discover a service or resource is strongly dependent on the type of R&SD: reactive or proactive. The latency in reactive discovery mainly consists of the time to propagate the service request up to a node that offers a service and the time for the service reply to arrive at the requesting node. In proactive discovery, no latency to discover the service is involved.

The above discussion makes clear that decentralized R&SD mechanisms have some interesting characteristics that are highly suited for mobile ad hoc networks.

4 Performance Evaluation

This section presents a simulation analysis of proactive and reactive R&SD techniques. This analysis is focused on the network aspects of the techniques and does not make any assumptions on message syntax, semantics… Fig. 1 and Table 1 present the reference scenario used in the simulations and the relevant network parameters.

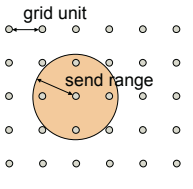


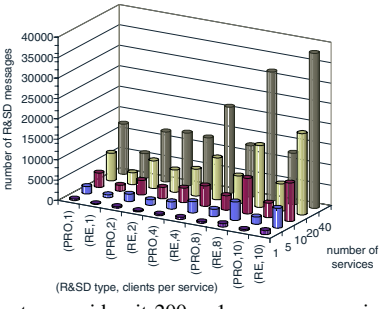
Fig. 1. Reference scenario

Table 1. Network parameters

Link/Radio layer	802.11b radio (2.4 GHz, DCF), 11 Mbps, two-ray path-loss, accumulated noise, send range +/- 250 m.
Routing Protocol	WRP [5] (proactive) or AODV [6] (reactive)
R&SD	number of services, number of clients per service (per period of 150 s), number of servers per service, proactive R&SD: service announcement interval, announcement broadcast delay, reactive R&SD: starting hop count, hop count increment
Application layer	CBR traffic from client to server over UDP
Mobility	36 nodes placed in grid, mobility: none or random waypoint with constant speed and pause time 0 s

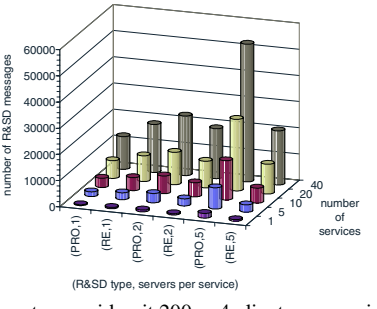
Protocol Overhead and Delay

Fig. 2 and Fig. 3 show the protocol overhead, expressed as the number of R&SD messages (SREQs, SREPs and SANNs), as a function of the number of services and the number of clients and servers per service. Fig. 2 clearly shows that proactive R&SD scales with the number of clients, as its overhead only depends on the number of servers. On the other hand Fig. 3 proves that reactive R&SD scales with the number of servers, as it only depends on the number of clients that request services. The choice whether to deploy proactive or reactive R&SD strongly depends on the service context. This means that developing a scalable decentralized R&SD mechanism should be a hybrid that deploys both reactive and proactive R&SD, where the choice between proactive and reactive depends on the service context and can even be different for different service types. Frequently used services should announce their presence proactively through SANNs, rarely used services should be requested by the clients through SREQs. This also implies that the preferred decentralized protocol is capable of adapting itself dynamically to the service context in the network.



Parameters: grid unit 200m, 1 server per service, announcement interval 60s, no mobility

Fig. 2. Overhead of proactive and reactive R&SD on top of AODV as a function of the number of services and clients per service



Parameters: grid unit 200m, 4 clients per service, announcement interval 60s, no mobility

Fig. 3. Overhead of proactive and reactive R&SD on top of WRP as a function of the number of services and servers per service

Wireless multi-hop networks require routing protocols, which also rely on proactive and reactive mechanisms. This observation motivates the need to investigate the interaction of decentralized R&SD mechanisms with the underlying routing protocols. To this end, we evaluated the performance of proactive and reactive R&SD on top of a proactive (WRP) and reactive (AODV) routing protocol (Note that in this study the R&SD messages and routing messages are completely separated). In case a reactive routing protocol is used, different degrees of coupling with the reactive R&SD are possible.

In case there is no coupling, AODV will create a route to the client when the server wants to send a service reply (SREP) to the requesting client. In this case, the SREP is sent directly to the client on IP level, which means that the SREP is transparent for the intermediate nodes. This option rules out the possibility of caching without violating

the layered protocol structure. In case of loose coupling, the SREQ will interact with the routing protocol and will be used to create a backward path to the client. In this case, the SREP is sent to the client on a hop-by-hop basis and is processed by the R&SD of each intermediate node. Once the client wants to send data to the server, a forward path still needs to be created (but there is no additional routing overhead for sending back the service reply). The last option is strong coupling, where the SREQ will be used to create a backward path and the SREP will be used to create a forward path. In this case, the reactive R&SD protocol will create a bi-directional path if a client requests a service.

Fig. 4 and Fig. 5 illustrate the impact of R&SD and routing protocol combinations on the R&SD delay (time it takes to find the service) and the routing delay (time between finding the service and the delivery of the first data packet). Note that AODV means loose coupling and AODVb means strong coupling. It can be seen that by using proactive R&SD the R&SD delay is reduced to 0 as all nodes in the network know all services. Also, as expected, the use of a proactive routing protocol results in much smaller routing delays. Fig. 5 also proves that a strong coupling between the R&SD protocol and AODV significantly reduces the routing delay. Again we can observe that the choice of decentralized R&SD mechanism and its interaction with the routing protocol strongly influences the delay. In the ideal case, the R&SD and the routing protocol should adapt their behavior to the network, not only to reduce the network load, but also to deal with the delay requirements imposed by the applications that need services.

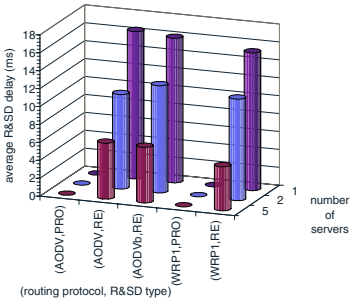


Fig. 4. R&SD delay of proactive and reactive R&SD on top of AODV (reactive) and WRP (proactive)

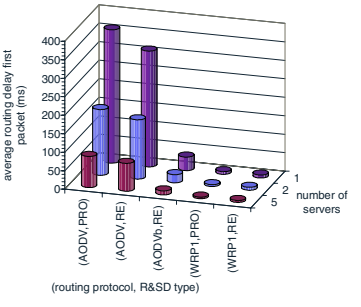


Fig. 5. Routing delay of proactive and reactive R&SD on top of AODV (reactive) and WRP (proactive)

Parameters: grid unit 200m, 5 services, 2 clients per service, announcement interval 60s, no mobility

In the above simulation results, basic versions, without any optimizations, of the reactive and proactive R&SD mechanisms were used. For reactive R&SD this means that the SREQ is propagated throughout the entire network. For proactive R&SD this means that servers announce their presence each announcement interval by broadcasting a service announcement in the network. However, a number of optimizations that improve the scalability can be used.

Improvements to Reactive R&SD: Expanding Ring Search and Caching

Instead of broadcasting the SREQ throughout the entire network, an expanding ring search could be used, where nodes first start to look for the services they need in their immediate environment. If no service has been found, the search area is gradually expanded. Fig. 6 shows the overhead reduction that can be obtained by using the expanded ring search concept. The type of expanding ring search (combination of starting hop count and hop count increment) is influenced by the average number of hops to reach a server (and thus by the number of servers per service).

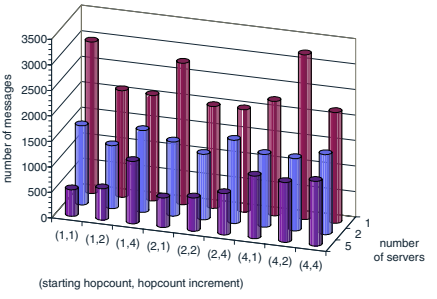
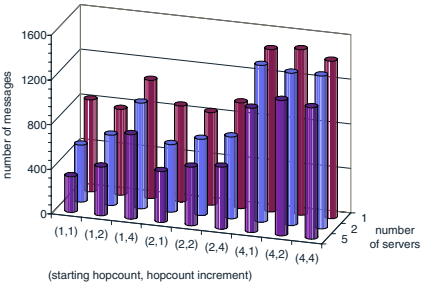
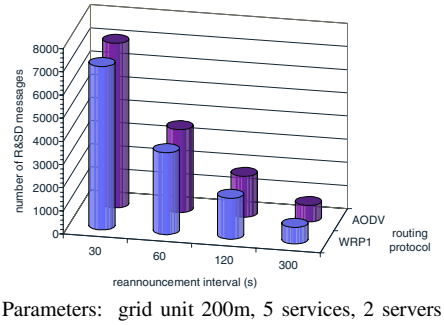


Fig. 6. Impact of expanding ring search on reactive R&SD



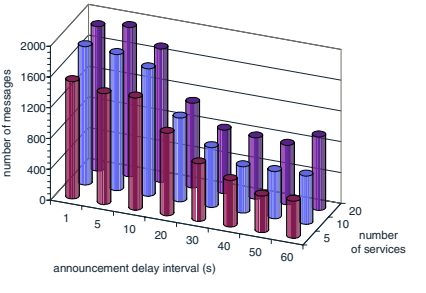
Parameters: grid unit 200m, 5 services, 4 clients per service, no mobility, AODV

Fig. 7. Impact of expanding ring search and caching by clients on reactive R&SD



Parameters: grid unit 200m, 5 services, 2 servers per service, 2 clients per service, no mobility, AODV

Fig. 8. Impact of announcement interval on proactive R&SD



Parameters: grid unit 200m, 4 clients per service, 2 servers per service, announcement interval 60s, no mobility, AODV

Fig. 9. Impact of message aggregation on proactive R&SD

Another improvement to reactive R&SD could be the use of caches. Clients that have discovered a service or nodes that store service information contained in SREPs can answer the SREQ from other nodes. Of course, information within these caches is only valid for a limited time, which in turn depends on the type of service. Also, by using caches some important information, such as the distance to the server, will become unavailable. In Fig. 7, clients that have discovered a service will also answer

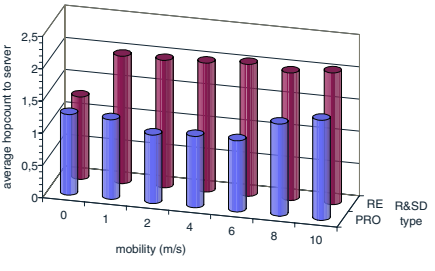
SREPs, and thus constitute a sort of service cache. The results show that caching reduces the number of hops the SREQ needs to travel to find the address of a server that offers the service, which in turn improves the efficiency of using an expanding ring search.

Improvements to Proactive R&SD: Announcement Interval and Message Aggregation

Fig. 8 and Fig. 9 show how the value of the announcement interval and the use of message aggregation help reducing the protocol overhead. The announcement interval determines with which interval servers announce the services they offer. Of course, the optimal value of this parameter cannot be increased as is, without taking into account the type of services and the network context. Message aggregation means that nodes store all SANNs that arrive during a certain time period (i.e. announcement broadcast delay interval). At the end of the interval, these announcements are combined into one bigger message, thereby reducing the protocol overhead (or several messages if the value of the announcement broadcast delay is too high, which of course reduces the effect of the aggregation). Note that as we do not make any assumptions on message syntax, we only take into consideration the number of messages, and thus the number of times access to the channel is needed, as a measure for the protocol overhead. Once a choice on the message syntax has been made, more detailed overhead calculations in terms of bytes are possible.

Impact of Mobility

If using reactive R&SD, a that node has found a service can start using it. However, the node will not detect if a better service (e.g. the same service but offered by another server closer to this node) has become available, unless it periodically performs a new discovery. When using proactive R&SD, nodes will become aware of better servers and can switch to a new server when appropriate. Fig. 10 shows how switching to a new server can improve the quality of the delivered service (in this case by reducing the hop count when switching to another server that delivers the same service). Of course, the service should support switching to a new server. An interesting example is gateway discovery.



Parameters: grid unit 150m, 2 services, 5 servers and 2 clients per service, announcement interval 30s

Fig. 10. Impact of mobility on proactive and reactive R&SD

5 Theoretical Analysis

In this section we theoretically derive the performance of proactive and reactive R&SD. A theoretical derivation is given for the message overhead of proactive R&SD, with and without message aggregation, and reactive R&SD, with and without

expanding ring search. Analog expressions for the R&SD latency can be derived similarly based on the per hop latency. Table 2 shows the notations that are used.

Table 2. Notations

Notation	Description
N	number of nodes
l(i)	Prob[route length between 2 random nodes is i hops]. For the reference scenario in Fig. 1 the values l(0) to l(10) are the following: 36/1296, 120/1296, 196/1296, 232/1296, 232/1296, 200/1296, 140/1296, 80/1296, 40/1296, 16/1296, 4/1296.
L(i)	Prob[route length between 2 random nodes is ≤ i hops] = $\sum_{k=0}^{i} l(i)$
N(i)	number of nodes in a region of i hops around a random node. The following holds: $N(i) \approx L(i)\, N$
l_{MAX}	maximum route length in hops ($\sum_{i=0}^{l_{MAX}} l(i) = 1$)
L	average route length ($\sum_{i=0}^{l_{MAX}} i\, l(i)$)
S	number of services
s	number of servers per service
f_{req}	number of service request per second and per service
f_{ann}	number of service announcements per second (per server and per service) for proactive R&SD
f_{del}	broadcast announcement delay frequency
M_{pro}	message overhead of proactive R&SD (messages per second)
M_{re}	message overhead of reactive R&SD (messages per second)
Notations for calculating the overhead of reactive R&SD with expanding ring search	
S(k)	Prob[minimum 1 server within k hops]
s(k)	Prob[nearest server is at k hops]
$M_{re}(k)$	message overhead of reactive R&SD (messages per second) if the nearest server is at k hops
h_{start}	starting hop count
h_{inc}	hop count increment

Basic Proactive and Reactive R&SD

The overhead when using proactive R&SD consists of the service announcements that are broadcasted periodically throughout the entire network. Note that, as blind flooding is used, each broadcast costs N transmissions in a network of N nodes. For basic reactive R&SD, the overhead consists of the SREQs that are propagated throughout the entire network and the SREPs that are sent back by all corresponding servers. This leads to the following simple expressions for the message overhead.

$$M_{pro} = S\, s\, f_{ann}\, N$$
$$M_{re} = S\, f_{req}\, N + S\, s\, f_{req}\, L \tag{1}$$

Proactive R&SD with Message Aggregation

The amount of aggregation depends on the announcement broadcast delay interval. In the optimal case, the choice of f_{del} will be such that at the end of the interval multiple

SANNs can be aggregated into one single message. Too small values of the interval will reduce the efficiency, as at the end of the interval there are not always SANNs available for aggregation; too high values will also reduce the efficiency as in this case the SANNs will not fit into a single message anymore. For the optimal case, the overhead is approximated by the following expression:

$$M_{pro} = f_{del}\ N \quad with\ f_{del} < S\ s\ f_{ann} \tag{2}$$

Reactive R&SD with Expanding Ring Search

The probabilities S(k) can be calculated as follows:

$$S(k) = \begin{cases} 1 & ,if\ N - N(k) < s \\ 1 - \prod_{i=0}^{s-1} \frac{N - N(k) - i}{N - i} & ,if\ N - N(k) \geq s \end{cases} \tag{3}$$

From S(k) we can easily derive the probabilities s(k).

$$s(0) = \frac{S(0)}{S(l_{MAX})}\ ,\ s(k) = \frac{S(k) - S(k-1)}{S(l_{MAX})} \quad for\ k = 1..l_{MAX} \tag{4}$$

Let us define i_{MAX} as the value for which the following equation holds:

$$h_{start} + (i_{MAX} - 1) h_{inc} \leq k \leq h_{start} + i_{MAX}\ h_{inc} \tag{5}$$

Now we can calculate the values of $M_{re}(k)$ as follows:

$$M_{re}(k) = N(\min(l_{MAX}, h_{start} - 1)) + k + (s-1) S(\min(l_{MAX}, h_{start})) \frac{k + \min(l_{MAX}, h_{start})}{2}, if\ k \leq h_{start} \tag{6}$$

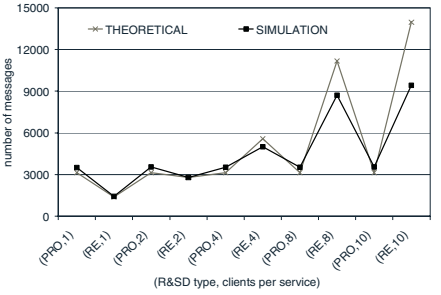
$$M_{re}(k) = \sum_{i=0}^{i_{MAX}-1} N(h_{start} + i\ h_{inc} - 1) + N(\min(l_{MAX}, h_{start} + i_{MAX}\ h_{inc} - 1)) + k + (s-1) \begin{bmatrix} S(\min(l_{MAX}, h_{start} + i_{MAX}\ h_{inc})) \\ - S(h_{start} + (i_{MAX} - 1)\ h_{inc}) \end{bmatrix} \frac{k + \min(l_{MAX}, h_{start} + i_{MAX}\ h_{inc})}{2}, if\ k > h_{start} \tag{7}$$

The first equation, which holds if only one iteration in the expanding ring search is needed, consists of three terms: the SREQ, the SREP of the closest server, the SREPs of other servers in the area covered by the SREQ. The second equation, which holds if there are $i_{MAX} + 1$ ($i_{MAX} > 0$) iterations needed in the expanding ring search, consists of four terms: the SREQs of the first i_{MAX} unsuccessful iterations, the SREQ of the last successful iteration, the SREP of the nearest server and the SREPs of other servers in the additional area covered by the last SREQ. Based on the probabilities s(k) and the values of $M_{re}(k)$, we can compute the total overhead as:

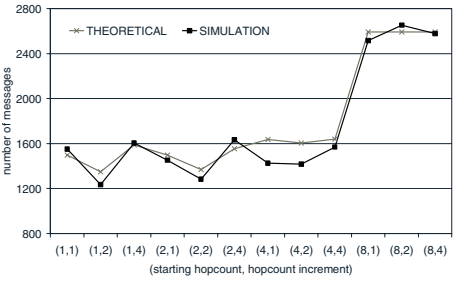
$$M_{re} = \sum_{k=0}^{l_{MAX}} M_{re}(k)\ s(k) \tag{8}$$

In Fig. 11, a comparison is made between the theoretical and analytical results.

We can observe that the obtained results are quite similar, which makes the analytical approach useful and reliable for evaluating the overhead in larger ad hoc networks. The only requirement is the knowledge of the probabilities l(i). For random ad hoc networks, the calculation of these probabilities is a research topic on its own.



Overhead of proactive and reactive R&SD. Parameters: grid unit 200m, 1 server per service, announcement interval 60s, 10 services.



Expanding ring search. Parameters: grid unit 200m, 5 services, 4 clients per service, 2 servers per service.

Fig. 11. Comparison of analytical and simulation results (no mobility)

6 Conclusion

In this paper we have discussed the tradeoffs between centralized and decentralized R&SD techniques. As decentralized solutions have some interesting advantages in mobile multi-hop networks, we have evaluated the performance of reactive and proactive R&SD both through simulations and analytically. Our results show that the choice between them is not straightforward. It highly depends on the network and service context and on the interaction with the underlying routing protocols. Ideally, the R&SD and routing protocols should be developed in close cooperation or even be integrated and should be able to adapt their behavior to the demands of the network and applications [7]. Therefore, our analysis provides protocol developers with some guidelines for developing new or extending existing resource and service discovery protocols for operation in mobile ad hoc networks.

Acknowledgements. This research is partly funded by the Belgian Science Policy through the IAP V/11 contract, by The Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT) through the contract No. 020152, by the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium) and by the EC IST integrated project MAGNET (Contract no. 507102). Jeroen Hoebeke is Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.-V., Belgium)

References

1. C-K. Toh, “Ad Hoc Mobile Wireless Networks: Protocols and Systems”, Prentice Hall, 2002.
2. U.C. Kozat and L. Tassiulas, “Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues”, *Ad Hoc Networks*, vol. 2, no. 1, 2004, pp. 23-44.

3. C. Lee and S. Helal, "Protocols for Service Discovery in Dynamic and Mobile Networks", *International Journal of Computer Research*, vol. 11, nr. 1, 2002, pp. 1-12.
4. X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks", *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS'98)*, Banff, Alberta, Canada, May 1998.
5. S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *Mobile Networks and Applications*, vol. 1, no. 2, Oct. 1996, pp. 183-197.
6. C.E. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector", *Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, U.S.A, Feb. 1999, pp. 90-100.
7. J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester, "Adaptive Multi-Mode Routing in Mobile Ad Hoc Networks", *Proceedings of the 9th International Conference on Personal Wireless Communications 2004 (PWC 2004)*, Delft, The Netherlands, September 2004, pp. 107-117.

Location Assisted Fast Vertical Handover for UMTS/WLAN Overlay Networks

Tom Van Leeuwen[1], Ingrid Moerman, Bart Dhoedt, and Piet Demeester

Department of INformation TEChnology - Gent University,
Sint-Pietersnieuwstraat 41,
9000 Ghent, Belgium
tom.vanleeuwen@intec.ugent.be
http://www.intec.ugent.be

Abstract. UMTS/WLAN integration offers considerable benefits for the users as well as for the mobile network. As soon as coverage is available, a mobile user should be able to switch seamlessly from the ubiquitous low bandwidth UMTS to a high bandwidth WLAN. This will improve his quality of service, but this also increases capacity in the cellular network. To be able to change between radio access technologies, an inter-RAT handover protocol is needed. If this handover protocol is not fast enough, the vehicular user will penetrate deep into the cell, underutilizing the available bandwidth with a lower overall performance as a result. In this paper we introduce a new vertical handover protocol which uses location information from the vehicle to predict where and when the handover should occur, thus optimizing user throughput and network performance. We consider its deployment on three UMTS/WLAN integration scenarios.

1 Introduction

Beyond 3G systems are considered to be heterogeneous networks with multiple of radio access technologies (RATs) as well as reconfigurable user terminals in order to allow mobile users to enjoy seamless wireless services irrespective of their location, speed or time of day. Users will be able to choose its access technology according to his or her own needs. This concept of "Always Best Connected" (ABC) [1] is shared by the ETSI 3rd Generation Partnership Program (3GPP) and it has therefore laid the basis for a UMTS/WLAN interworking specification. While Wireless Local Area Networks can offer high bandwidth, cellular networks provide a (nearly) full coverage. It is obviously advantageous for the mobile user to connect to the WLAN network as soon as coverage and capacity are available. The user terminal will therefore regularly scan the corresponding frequency bands and try to detect beacons of UMTS or WLAN cells. If signal strength is adequate, a handover to the other network tech-

[1] Research funded by a Ph.D. grant for Tom Van Leeuwen of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 192–202, 2005.

nology is attempted. The handover of connections between different networks with different radio access technologies is also called vertical handover, but these inter-RAT handover procedures have not yet been defined by the 3GPP, except for GSM-UMTS/UMTS-GSM handover. Most of papers in the literature discuss the aspects related to the common management of radio resources and authentication, authorization and accounting (AAA) among heterogeneous networks from an architectural point of view, describing constraints, potential advantages and drawbacks of various integration levels [2]. However, fast mobility management for vehicular users in heterogeneous networks is in a very early stage and still requires many research efforts [3] [4] [5]. Some research suggest Mobile IP [6] or and end-to-end mobility scheme like SIP [7] for heterogeneous networks. Mobile IP is a well known IP network layer mobility management scheme, in which packets from and to the mobile host are tunneled through a home agent at its home network so that the corresponding node that communicates with this mobile host is unaware of the mobility of the mobile host. Unfortunately, Mobile IP suffers from high delay and packet loss and is therefore not suited for fast moving mobile users. End-to-end based mobility management protocols can handle mobility without additional support from the network elements. The drawback of this end-to-end approach is that, especially in the case of fast moving users requiring fast handover, this protocol suffers from delay as both parties can be geographically spread. Additionally, this mobility scheme is only valid for applications that are SIP-aware.

In this paper we propose a proactive inter-RAT vertical mobility management protocol for all IP UMTS/WLAN interworking access networks called APACHE (A ProACtive Handover Enhancing protocol). It uses location information from the mobile user, for example acquired by a GPS system, and tries to proactively determine to which WLAN access point or UMTS base station the mobile user will switch and when this handover should occur. Our goal is to optimize the use of the available bandwidth for the user, keeping handover delay as low as possible and avoiding packet loss. In the next section we discuss possible UMTS/WLAN interworking architectural scenarios that allow inter-RAT mobility. Section 3 discusses the requirements for a location assisted proactive handover approach and introduces the APACHE protocol. The implementation of APACHE on the different interworking scenarios is also investigated. Following this section, we compare the performance of the APACHE protocol with a hard and a forwarding based handover scheme. We finish this paper with some conclusions.

2 UMTS/WLAN Inter-RAT Handover

In a UMTS/WLAN overlay network, one has typically large UMTS cells that provide nearly ubiquitous coverage, while in areas with high user density and high bandwidth demand, small WLAN cells will be deployed (e.g. in a town center). Vehicular users will have the choice of handing over from the low bandwidth UMTS cell to the higher bandwidth WLAN cell. Inside the WLAN access network, the UE is addressable by a local IP address assigned by a Dynamic Host Configuration Protocol (DHCP) server

(figure 1). In the UMTS network a fixed external IP address will be assigned by the GGSN and makes the UE addressable to any host in the internet. The Packet Data Gateway (PDG) in the WLAN access network binds the local address of the UE with its external address and performs network address translation on any packet from or to the UE in the WLAN. We assume the local AAA server interacts with the AAA server in the subscriber's home UMTS network.

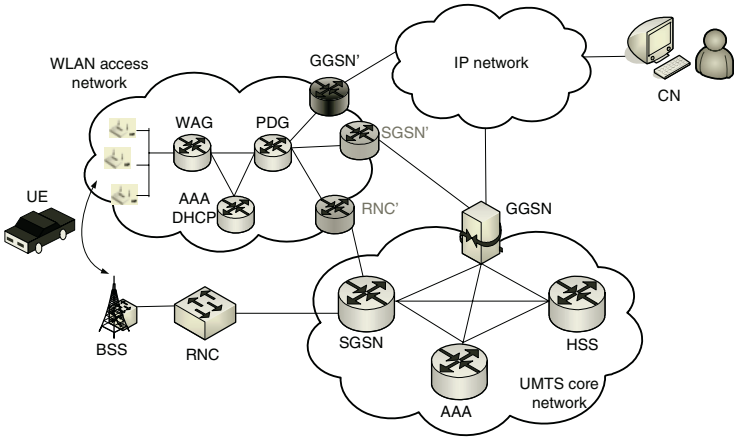


Fig. 1. Three degrees of UMTS/WLAN coupling

The inter-RAT mobility management procedures are different depending on the degree of coupling between both access networks. The WLAN and UMTS interworking architecture is standardized in 3GPP with the following alternatives [8][9] (although other variants are possible).

- **Very tight coupling**

Basically, in the very tight coupling approach the WLAN access network is considered as a generic UMTS RAN, connected to the UTRAN core network. Interworking is realized via the Iu interface between the UTRAN core network's SGSN and a RNC emulator in the WLAN access network (RNC'). In this case the management of user's mobility between the two access network technologies is performed through basic UMTS intra-SGSN handover specifications. This approach offers the lowest handover delay, but poses high hardware and software requirements on the RNC emulator that may be technologically or economically unfeasible.

- **Tight coupling**

In this case we have a WLAN connected to the UMTS core network via an SGSN emulator (SGSN') through the Gn interface. The WLAN coverage area appears like another routing area to the UMTS core network, but governed by a different SGSN. Mobility from UMTS to WLAN or from WLAN to UMTS will result in inter-SGSN routing area update procedure. All following packets from the internet will arrive at the GGSN and will be tunneled from GGSN to SGSN' over the core network and

from SGSN' to the UE. The vertical handover delay is higher than the very tight coupling approach due to packet forwarding and PDP context transfer from SGSN to SGSN'/PDG and vice versa.

- **Loose coupling**

In the loose coupling approach, the WLAN access network is not directly connected to the UTRAN core network, but with a SGSN/GGSN emulator (GGSN') to the internet via the Gi interface. Packets, that were not delivered to the UE, need to be forwarded from the SGSN (GGSN'/PDG) to the GGSN' (GGSN) in case of UMTS to WLAN (or WLAN to UMTS) handover. All following packets from the internet will arrive at the GGSN and will be tunneled from the GGSN to the GGSN' via a dedicated tunnel over the internet and from the GGSN' to theUE. This internet tunnel will also be used for signaling traffic towards the GGSN in the UMTS core network. In this case no existing handover specifications are applicable and are yet to be defined. Note that in these interworking scenarios, mobility across WLAN and UMTS should not result in a change of external IP address for the mobile user. The mobile user will always be reachable through his external IP address assigned by the UMTS core network. The core network's GGSN is considered as the gateway to the internet and thus also to any corresponding internet host (CN). This way we avoid the need for an external mobility management system like Mobile IP or SIP. The correct access network location of the user is always maintained by the Home Subscriber Server (HSS) in the UMTS core network.

3 Location Assisted Proactive Vertical Handover

3.1 Introduction

To achieve seamless mobility management, a small handover delay is critical. In the APACHE protocol we use location information of the mobile user to determine to which next base station or access point the subscriber will connect and to set up the routing path proactively. The authors of [5] use a database of offline discrete power measurements, and use pattern matching to determine the location of the mobile user based on his online power measurement reports. However, we will show that a reasonable accurate location determination is needed to be able to outperform the existing forwarding based vertical handover protocol. This is however impossible without external location determination systems. Given the low cost of GPS devices nowadays, we predict that future vehicles will have default an accurate GPS system on board. We use this information in our proactive handover protocol.

3.2 APACHE Protocol

In mobile overlay networks it is important to choose the network with the best performance for the customer as soon it comes into range. If the access network with higher bandwidth or QoS is not detected fast and the handover delay is large, then the mobile user will have penetrated deeply in the coverage area of the cell before it can take advantage of the additional available resources. This is especially the case for

fast moving vehicular users. The APACHE handover protocol uses a prediction algorithm to decide when to switch from one access technology to the other. As already mentioned, we assume that the mobile user transmits on a regular basis its GPS coordinates to the network at which it is currently connected. These GPS coordinates have typically a fixed accuracy and precision depending on the manufacturer, e.g. an accuracy of 10m for a precision of 95%. This means that there is a probability of 95% that the user is located in an area with radius 10m around the calculated GPS coordinates. Given the hardware constraints regarding the frequency at which these coordinates are updated (in the order of seconds), we use an extrapolation technique to determine the next position of the mobile user. In the UMTS system, the UE reports on a regular basis its power managements, including the time at which a base station beacon has been received. The frequency of these updates is much higher than the location updates from the GPS system, e.g. in the order of milliseconds. Because the time between those beacons is as good as fixed, we extrapolate the next coordinates of the location of the UE starting from the last GPS coordinates taking into account the number of beacons the UE has received, the speed of the vehicle and the direction in which the vehicle is driving based on his previous GPS coordinates. In this way we can estimate the user's following position without having to wait for new GPS coordinates, provided that the vehicle's movements are not random. Note that according to the IEEE 802.11 standard

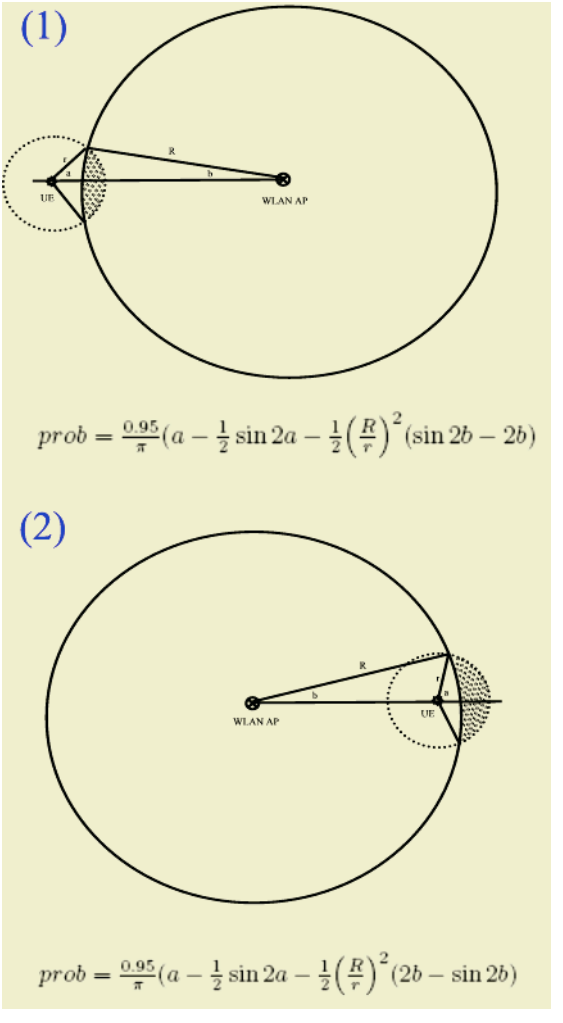


Fig. 2. Calculating the probability that a GPS equipped user has crossed the handover barrier when entering or exiting the WLAN cell respectively.

specification, the user is not required to give feed back on its beacon measurements to the WLAN network. However, the IEEE 802.11k study group proposes a new standard for radio resource management and aims to provide key client feedback to WLAN access points. Here we assume that the UE also provides beacon measurement reports in the WLAN as in the UMTS access network. Once the estimated location of the mobile user is determined for the next expected measurement report, the probability that the user will have crossed the handover barrier is calculated. Figure 2 gives these probabilities (1) for UMTS to WLAN and (2) for WLAN to UMTS handover. These probabilities are equal to the relation of the shaded area to the surface of the GPS accuracy circle, multiplied by the GPS precision (here 95%). If this probability is higher than a threshold value eta, then the handover process will be initiated by the APACHE protocol. The handover barrier itself, must be predetermined by measurements in the field and can be stored in a database system, for example for each road entering and exiting the WLAN coverage area.

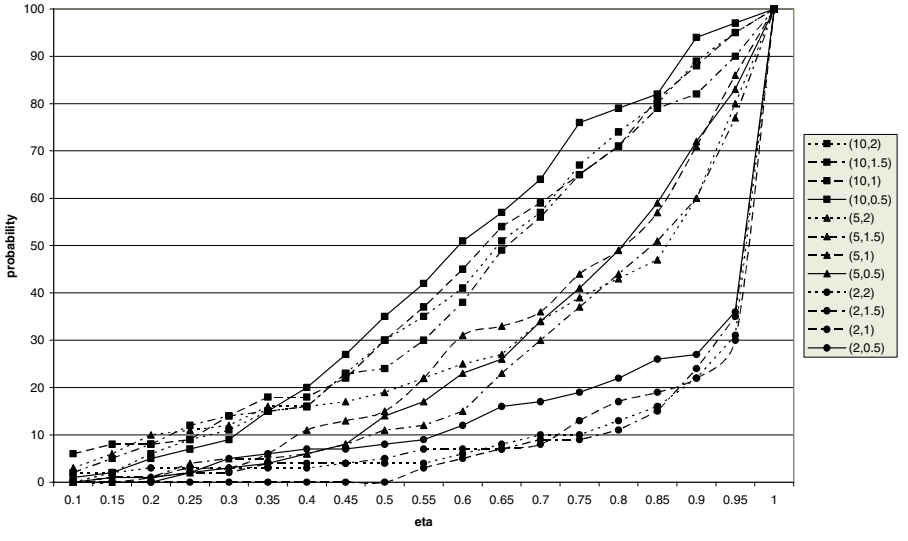


Fig. 3. Cumulative distribution of the optimum value of eta for different GPS accuracies and update periodicities

Figure 3 shows the cumulative distribution function for the optimum value of eta for a fixed mobile user speed of 20 m/s. The GPS accuracy goes from 10m to 2m and the GPS update periodicity from 0.5s to 2s. For lower GPS accuracies of e.g. 10m, a good value for eta is difficult to choose, because for every value of eta, there will be an equal probability that (non-proactive) handover happens sooner or later. If GPS accuracy is 5m and we choose a value for eta of 0.5 (in this case the shaded area in figure 2 is halve the GPS accuracy circle), then we can expect that only 20 % of the vehicles will make a handover sooner than we expected. The figure also shows that

with our estimation technique variances in the GPS update periodicity do not have an important impact on the distribution of eta.

Figure 4 depicts the influence of the variance in the mobile user's speed on the estimations. The user moved with an average speed of 20m/s, the GPS updates had a periodicity of 1s up to 2s and the value of eta was in this case 0.5. For a constant speed (variance = 0 m/s), the APACHE estimation made the right decision in 70% of the cases. In 20 % of the cases, the handover occurred sooner and in 10% of the cases later. If more variance in the speed is introduced, then the number of correct handover decisions will decrease.

However, in this case the percentage of handovers that happened sooner than our estimations will increase more than the percentage of handovers that were later. These results have different meaning if handover occurs from UMTS to WLAN or from WLAN to UMTS in the APACHE handover protocol.

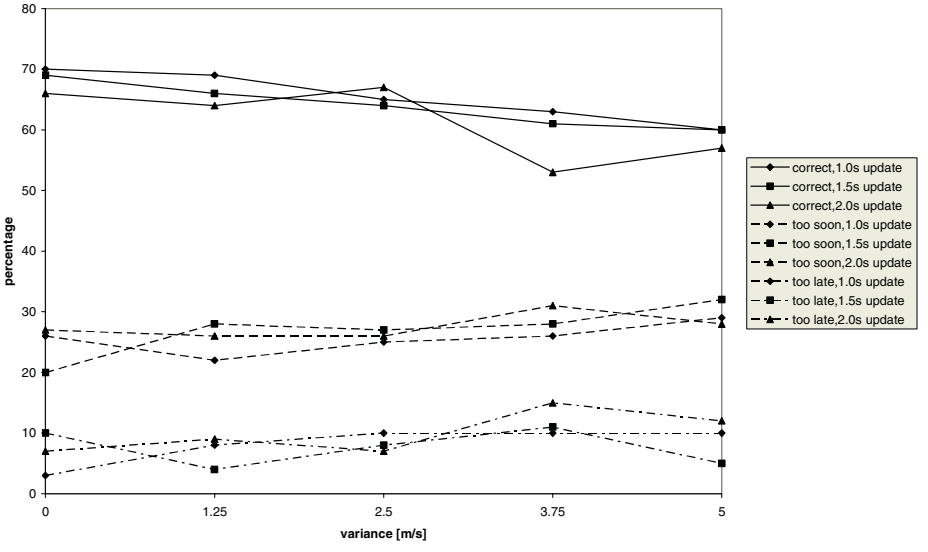


Fig. 4. Influence of the variance of the user's speed on the APACHE estimation algorithm

We first consider the UMTS to WLAN handover situation. Off course, if estimations were correct than a path to the next WLAN access point will be set up (handover preparation phase) and all new arriving packets will be directed towards that access point. If an inter-RAT handover occurs sooner than we had expected, the APACHE estimations are canceled and normal forwarding based handover is initiated. Upon arrival at the WLAN access point, a path to the GGSN will be set up and packets are forwarded from the old SGSN. If handover occurred later than we expected, the user will disconnect from the UMTS network and begins actively scanning for the appropriate WLAN access point until it comes into range. In the mean time, all packets will mount up in the buffer of the WLAN access network until the

user makes the actual handover. Thus, no packets are lost, but some jitter is experienced by the user.

In the case of WLAN to UMTS handover and the handover happens sooner than we expected, again the APACHE estimation is canceled and the forwarding mechanism initiated. Because we are considering a UMTS/WLAN overlay network and handover to the UMTS network is always possible due to its omnipresence, handover that happens too soon has a different meaning in this context. In this case the user is unable to take full advantage of the coverage area of the WLAN because the APACHE protocol will order it to switch to the UMTS network before the WLAN cell edge is reached.

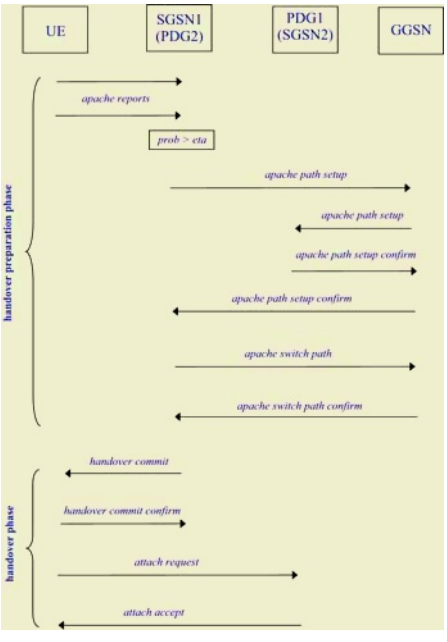


Fig. 5. APACHE protocol message chart

Figure 5 shows the different steps of the APACHE protocol for UMTS to WLAN and WLAN to UMTS handover.

4 Implementation Requirements

In the following sections we discuss the implementation of the APACHE protocol in the different scenarios of UMTS/WLAN interworking. We consider the UMTS to WLAN handover. APACHE procedures can be applied analogously to WLAN-UMTS handover.

- **Very tight coupling**

This architecture offers the lowest delay for the UMTS/WLAN inter-RAT handover. After the UE has associated with the WLAN access network, acquired a local IP address, executed AAA functions and has established the tunnel with its PDG, instead of forwarding further packets to the RNC of the last UMTS BSS, the SGSN now needs to redirect them to the RNC' of the WLAN. For UMTS to WLAN handover, APACHE offers only limited performance improvement as in this case only context transfer is avoided after handover if APACHE's estimations were correct. However in the WLAN to UMTS case, the forwarding of packets and the context transfer from the PDG to the SGSN is avoided.

- **Tight coupling**

If APACHE predictions were correct, the UE's contexts should already be available at the PDG and the SGSN' after the UE has made the handover. The tunnel between the GGSN and the SGSN' has already been setup by the APACHE protocol and packets, buffered at the PDG, can immediately be delivered to the mobile user. On the other hand, if the path has not yet been set up by APACHE, normal inter-SGSN handover procedures are performed between the UMTS SGSN and the WLAN SGSN'. In this case packets destined for the UE are forwarded from the old SGSN to the WLAN PDG and handover delay increases. Further packets arriving at the GGSN will be tunneled to the SGSN' over the UMTS core network and from the SGSN' to the UE.

- **Loose coupling**

In the loose coupling approach, the APACHE protocol will have proactively set up a tunnel between the GGSN of the UMTS core network to the WLAN GGSN' when the mobile user attaches to the WLAN access network. Again, packets buffered at the PDG can immediately be delivered to the UE. If the APACHE prediction algorithm was unsuccessful, this tunnel has to be set up and contexts need to be canceled at the previous SGSN after the UE has registered itself to the WLAN PDG, again increasing the handover delay.

5 Results

In this section we show that our location assisted vertical handover approach (APACHE) is able to achieve a higher performance level than current solutions in terms of user experienced throughput. The APACHE protocol was implemented in the NS-2 [10] simulator. In figure 6 the mobile user is moving at a speed of 20 m/s and is downloading a file with an FTP application from a server in the internet. The user makes a handover from a 2 Mb/s UMTS cell to a 5 Mb/s WLAN cell. The TCP advertised receive window allows an average TCP throughput of around 2.4 Mb/s, although the user is unable to achieve such a throughput due to bandwidth restrictions in the UMTS cell, until he has arrived at the WLAN. It is important to mention that the WLAN network had a 10% faster link to the GGSN or SGNS in these simulations. We observe that the TCP throughput increases after the handover, but the APACHE handover protocol allows a much faster increase than the hard handover or forwarding

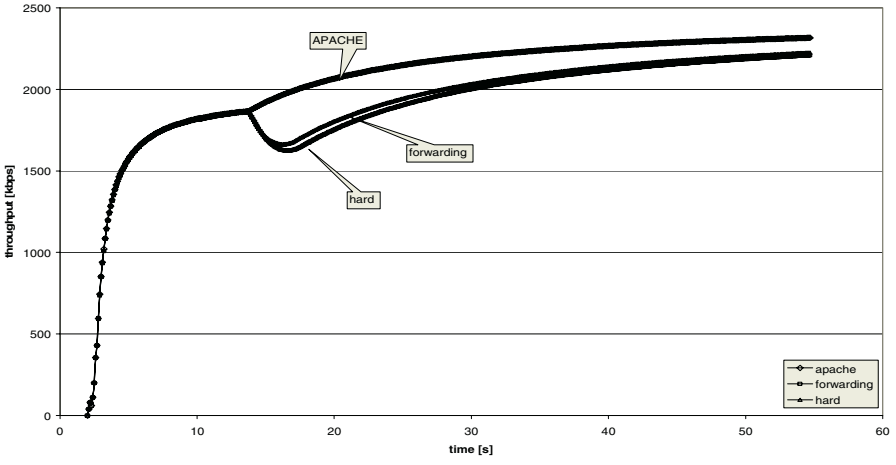


Fig. 6. TCP throughput comparison between APACHE, hard and forwarding handover protocol

handover protocol. In case of hard handover and forwarding based handover, packet loss and packet reordering respectively cause a TCP retransmission timeout and spurious retransmissions after handoff. This limits the user's observed throughput. The APACHE handover protocol on the other hand achieves a seamless handover when the prediction was correct. Note that if the WLAN cell would have had an equal or higher access network delay than the UMTS cell, then the TCP throughput of the forwarding based handover protocol would be the same as the APACHE TCP throughput because the packets arrive in-order. This is however not always guaranteed, for example in the loose UMTS/WLAN coupling approach where the WLAN network is LAN or even gigabit LAN based with low link delay.

6 Conclusions

In this paper we introduce the APACHE handover protocol for vertical handover in UMTS/WLAN overlay networks. Using location information from the vehicular user we can achieve fast and seamless handover from a low to a high bandwidth radio access technology and vice versa. We have discussed its possible deployment in three UMTS/WLAN integration scenarios. The APACHE handover protocol approaches the speed of a hard vertical handover protocol without data loss such that a true seamless user experience is possible. However some overhead on the wireless link due to the location updates from the mobile user is required.

Acknowledgement. Part of this research is funded by the Belgian Science Policy Office through the IAP (phase V) Contract No. P5/11 and by the Flemish IWT through the GBOU Contract 20152 'End- to-end QoS in an IP Based Mobile Network'.

References

1. Gustafsson, E.; Jonsson, A.: Always best connected. IEEE Wireless Communications, Vol. 10, 2003, pp.49-55
2. Salkintzis, A.K., Fors, C., Pazhyannur, R.: WLAN-GPRS integration for next-generation mobile data networks, IEEE Wireless Communications, Vol. 9 (5), 2002, pp. 112- 124
3. Varma, V.K., Ramesh, S., Wong, K.D., Barton, M., Hayward, G., Friedhoffer, J.A.: Mobility Management in Integrated UMTS/WLAN Networks, proceedings of ICC 2003, 2003, Vol. 2, pp. 1048 - 1053
4. Guo, C., Guo, Z., Zhang, Q., Zhu, W.: A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks. IEEE Journal on Selected Areas in Communications, Vol. 22 (5), 2004, pp. 834 – 848
5. Siebert M., Schinnenburg M., Lott M., Goebbels S.: Location Aided Handover Support for Next Generation System Integration, proceedings of EW 2004, 2004, pp. 195-202
6. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6, http://www.ietf.org/internet-drafts/draft-ietf-mobileipv6-24.txt, work in progress, 2003
7. Snoeren, A.C., Balakrishnan, H.: An end-to-end approach to host mobility, proceedings of MOBICON, 2000, pp.155-166
8. 3GPP TR 22.934 v6.2.0: Feasibility study on 3GPP system to Wireless Local Area Network (WLAN) interworking
9. 3GPP TR 23.234 v6.1.0: 3GPP system to WLAN interworking; System Description.
10. NS-2 simulator: www.isi.edu/nsnam/ns/

A Methodology for Implementing a Stress Workload Generator for the GTP-U Plane*

David Navratil, Nikolay Trifonov, and Simo Juvaste

University of Joensuu, Department of Computer Science,
Post Box 111, 80101 Joensuu, Finland
{david.navratil, nikolai.trifonov, simo.juvaste}@cs.joensuu.fi

Abstract. We present a framework for developing a traffic generator that produces massive, realistic network payloads. The techniques and methods in this article can be easily applied to any stress workload generator for network traffic simulation. Here, as the system to be tested, we use the UMTS/GPRS backbone including SGSN and GGSN, which utilizes GPRS Tunnelling Protocol (GTP-U) user plane messages to carry user data packets. The proposed workload generator system is characterized by high, real traffic load, economical standard hardware, scalability, and flexible extensibility. A large number of independent participants, such as mobile users and Internet servers, are modelled. The realism of traffic is achieved by using a layered modelling approach starting from the user/application level and ending at the network layer. High system throughput is obtained by exploiting preconstructed packet buffers (templates), packet filters, network interface polling, and an efficient, adjustable time resolution scheduler.

1 Introduction

General Packet Radio Services (GPRS) introduced new network elements in the public land mobile network architecture. The new elements are the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The role of the SGSN is to handle mobility management, authentication, and register functions. The GGSN provides access to a public data network, such as the Internet or the X.25. From the external networks' point of view, it is a router to a subnetwork. These two nodes are directly interconnected in the network architecture and form the core of the mobile data network. Therefore, the requirements on them are very high and careful attention must be paid to network planning and testing before the network is launched for public use. If the performance of the network does not satisfy customers' demands, then the situation can lead to the loss of customers. It is not unusual that a GSM network includes only one GGSN, through which all data transfers between Internet servers and subscribers' terminals are routed. In 3G networks (such as UMTS) the role of the nodes is similar, but the requirements are even higher. The services for 3G networks include high bandwidth, low latency applications, such as video-calls.

* This work was supported by Nokia Research Center.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 203–212, 2005.

Network testing can be seen as a two-phase process. A communication network is a distributed system with many different components communicating through standard interfaces. Thus, the communication between nodes has to be tested first. This type of test can be called an interoperability test. Interoperability tests cross-check the functionality of network elements. In the second phase of network testing, the goal is to test the throughput of the network and to find the limits of and bottlenecks in the system. Such tests are known as stress tests. To perform the stress test, a workload generator, which creates data traffic for the network being tested, is needed. Also, the stress test generator measures the performance of the system being tested.

In case of SGSN and GGSN stress testing, the performance and quality requirements for the workload generator are at least as high as the requirements for the actual support nodes. As mentioned above, a GPRS/UMTS network can include only one GGSN; this node must be able to handle data packets at a rate of several Gbps. Therefore, if engineers want to massively test the core network (i.e., SGSN and GGSN of the network), they need a workload generator that is able to generate data traffic at such a high rate that it saturates the network. Moreover, in addition to answering questions such as “How many packets per second can the nodes handle?” and “What is the actual throughput?”, engineers are also interested in testing the quality of services (QoS) and in observing the behaviour of the network from the mobile user's point of view. Thus, the workload generator must allow engineers to describe the generated traffic in terms of mobile users and applications.

Our approach to using a workload generator for testing the SGSN and GGSN consists of two parts, a control and signalling module and a data generation module. The first module performs signalling and control communication with the node being tested; the second module takes care of data generation. This article concentrates on the data generation module, which utilizes the GPRS Tunnelling Protocol (GTP). We describe a methodology for an efficient and inexpensive implementation of a stress workload generator.

Existing Generators and Related Work

Generally, a generator, which sends data through a network at some level of abstraction, simulates a real process or an application. The sending and receiving of data packets, which takes place in a discrete time, is considered to be an event in the system. Thus, the generator can be seen as a discrete event simulator. Methods and modelling techniques developed for discrete event simulation are used during an implementation of the generator. The level of abstraction used during the design of a workload generator plays an important role. Some workload generators use models represented by a stochastic process that simulate only the packet size and the arriving time of the packets at a low-level network interface. Other types of workload generators are built on models that try to describe user interaction with a system. Such models have a layered architecture and allow a user to create a large set of various user-behaviour profiles [5]. Moreover, the layered models can produce traffic patterns that are more realistic. Naturally, the generators implementing the layered approach are more complex and require more processing time during simulation than generators

using one stochastic process. As a result, however, the performance of the system can be observed at both user and application levels.

Demands of the current market for such generators and testing tools have inspired companies around the world to create the appropriate products. Among such tools are MGTS *i3000* by Catapult Communications [2], Cellular Performer Analyser by RADCOM[11], Cutting edge GPRS Support Node Testing offered by Hughes Software Systems [6], and EAST for UMTS by ipNetfusion [8]. These products are complex test suites for testing many network interfaces.

2 Architecture

The endpoints communicating via a GPRS/UMTS network are mobile terminals and Internet servers. We designed the data generation module (DGM) framework to be based on the simulation of these elements. Therefore, the basic item in the data generation module is a model that represents an application running on a mobile terminal (e.g., a WAP or WWW browser) or, correspondingly, an Internet server providing a service (e.g., an HTTP server). Obviously, the data traffic that is generated depends on the models running in the system. In other words, the data generation process is controlled through the models. The control over models includes creating, starting, stopping, and deleting models and setting model characteristics.

Scalability
DGM is designed to run on "inexpensive, standard pieces of hardware", which are, in practice, personal computers or entry-level servers. To increase generator throughput, we can add more computers to DGM. These issues lead us to the framework shown in Figure 1.

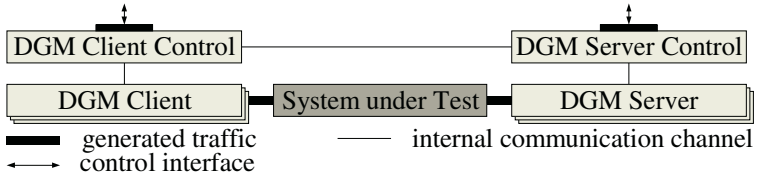


Fig. 1. DGM scheme

DGM consists of two control nodes and several data nodes. The controls nodes are DGM Client Control (DGCC) and DGM Server Control (DGSC). They provide control interfaces for users of DGM. DGM Client (DGC) and DGM Server (DGS) are data nodes that host the models. Data traffic is generated between the DGM data nodes.

We consider the GPRS/UMTS core network as the system under test as shown in Figure 1. We simulate the mobile terminals and Internet servers communicating over the network. However, the communication channel between the end elements also

includes other components of a mobile network, see Figure 2. In order to test SGSN and GGSN, the DGM data nodes replace all other parts of the communication channel. The number of replaced components depends on each test scenario. An example of possible test scenario is displayed in Figure 3. The scenario shows that the DGM clients can occupy the place of MT, GERAN/UTRAN, and possibly SGSN. Correspondingly, the DGM servers substitute Internet servers located in the public data network (PDN) and possibly GGSN during testing.

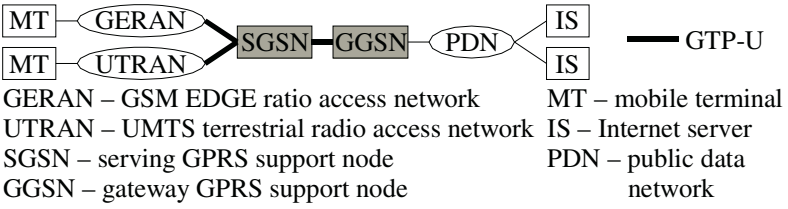


Fig. 2. GTP-U in GPRS

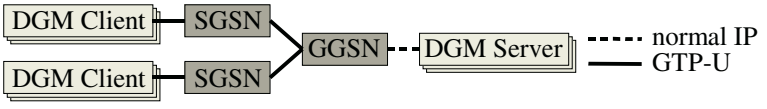


Fig. 3. An example of test scenario

2.1 Control Nodes

The control nodes provide an interface for simulation management. The duties of the control node include three main control tasks: monitoring and regulation of the load on system resources, user/application model manipulation, and statistical information reporting.

The data that are generated are sent through GTP-U tunnels; thus, the responsibilities of the resource management include taking care of the assignment of tunnel end-point identifiers and distributing load among the data nodes. In the simplest scenario, the client control and the server control distribute the load among the data node uniformly by assigning the same number of models to each data node. If system speeds of the data nodes vary, the loads of the nodes have to be monitored. In this case, a new model would be created at the node with the lowest load.

DGCC chooses the DGC where a model is going to be created. This information is not known outside the DGM; therefore, all model manipulation has to be done through the control interface of DGCC. For this purpose a unique identification number is associated with each model. Model manipulation is implemented by sending commands, including the identification number, to DGCC.

Statistics reporting utilizes the client-server architecture. The statistics about the simulation can be retrieved from the system by any process connected to DGCC and by requesting a report of the statistics. After the request is received by DGCC, the

statistics are periodically sent to the process. Then the process can visualize and/or store the statistics data in its own way. The statistics measured by the system include 36 values, such as current traffic volume (bits/s, packets/s), average round trip time, TCP retransmissions, and number of dropped IP packets. The values are measured for each protocol and traffic class.

2.2 Data Nodes

DGC and DGS are discrete event simulator components of the DGM. Both client and server have the same design schema, which is shown in Figure 4. The process flow of the data node is based on the processing of events that appear asynchronously in the system.

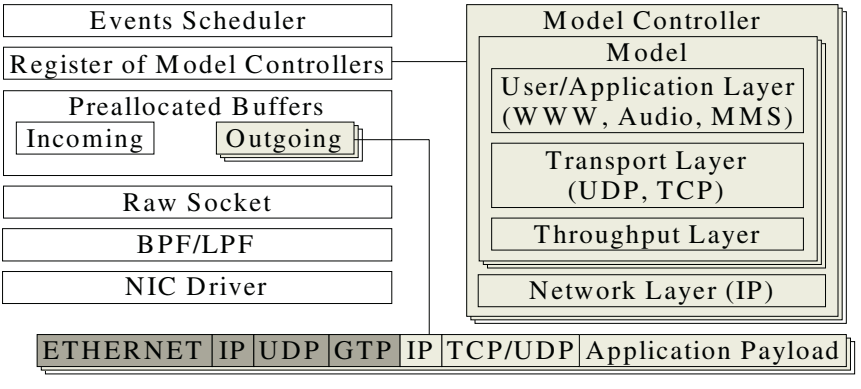


Fig. 4. Data node scheme

A Layered Approach to Implementing Models

The model is an object that simulates an application from the user's point of view. Inside DGM, the model is a composite of smaller model entities arranged in a layered hierarchy. Because DGM simulates network applications, it is obvious that the model entities correspond to the protocol hierarchy; see Figure 4. The lowest model entity simulates the network layer protocol, which is in our case Internet Protocol (IP). The throughput layer is not usually part of the ordinary protocol stack. Because the workload generator is intended to simulate a mobile environment, where the connection speed is limited by Packet Data Protocol context (PDP), this layer is included in the hierarchy. The purpose of this layer is to artificially limit the communication speed according to the PDP context assigned to the model. The next highest layer is the transport layer. Model entities of transport protocols such as UDP and TCP are found at this level. The User/Application model entity is situated at the top of the layered hierarchy. This layer is responsible for the simulation of user and application behaviour. The top layer may be further divided into two separate layers. This division offers more flexibility during concurrent simulations of different user's profiles [5]. In any case, the hierarchy presented here already gives testers the

possibility to observe the behaviour of each protocol layer as the load in the tested system increases. This is achieved by gathering statistics for each modelled layer.

The model entities in the layered hierarchy need to interact with other models. For this purpose, every layer provides a predefined set of services to a higher layer. The communication between layers is accomplished by means of service primitives. The service primitives are divided into four sets: requests, indications, responses, and confirmations. The request primitive is used when a higher layer requests a service from the next lower layer. The lower layer invokes the indication primitive in the next higher layer in order to provide information about activity on the lower level. The response primitive is used to confirm the receipt of the indication primitive received from the next lower layer. In the same sense, the confirmation primitive notifies the next higher layer about the successful accomplishment of the activity invoked by the request primitive. One request primitive usually requires a couple of other request primitives in the lower layer, and a couple of indications in the lower layer are needed to generate the indication in the next higher layer. All this depends on the services being modelled.

A distributed system is often characterized by its stochastic behaviour and time constraints within the system. In our case, stochastic automata are used to describe the behaviour of each model [1][3]. A stochastic automaton does not differ much from a finite automaton. The main difference is that a stochastic automaton has clocks. The clock can be set to a value upon the entry of a desired state. After the clock is set, a countdown of the clocks begins. The automaton can move from one state to another if there is an edge between the states, the input symbol is the same as the symbol assigned to the edge, and the set of clocks associated with the edge are all zeros. Further, we extend the stochastic automaton to the input/output automaton by changing the transition function. The transition function includes input/output events. An event will be generated in the system when a transition from one state to another is implemented and an output event is assigned to the transition used. We implement the countdown process through the scheduler. The event representing the achievement of zero in the clock is scheduled. The scheduler notifies the appropriate model by invoking the indication primitive.

Model Controller

We present a model controller that acts as a user on a client node and as a multi-service server on a server node. The model controller stores all running models of the user/server.

Scheduler

DGM simulates various network entities by running their models through the life cycle of the simulation. They behave as independent participants and the events produced by models are asynchronous by nature. Every event is marked with a logical time-stamp and put to the scheduler. The scheduler implements a logical time line and triggers the scheduled events.

The scheduler is a dual ring buffer consisting of slots that stand for discrete time units of the simulation. Every slot consists of a list of events scheduled to the same

time unit. The DGM time accuracy is defined by the time slot resolution of the scheduler. The default resolution is 1 ms in our generator. The length of the scheduler buffer can be extended months ahead to support simulation scenarios using user/application models with very low message frequency. We use dual buffers to achieve both high resolution and to schedule events very far in the future. Events scheduled to trigger within, e.g., the next minute are stored in the buffer with 1ms resolution. More distant events are stored in the low resolution buffer until they are moved to the high resolution buffer, just before the scheduled time. The models progress by processing the list of events in the slot. The logical time corresponds with real time accurately, except for occasional short term gaps, mostly in the case of process switches.

System Register

The system register is a container of model controllers. It is represented as a hash table. We use the open addressing hash table, with a double hashing, as a collision resolution method. The hash functions are optimised for usage of IP addresses as keys of the hash table.

Preallocated Buffers

Considering the performance of the system, sending packets is also critical as an efficient implementation of models. If we use a standard OS UDP/TCP implementation, the preparation of outgoing packets usually involves buffer copying. It is one of the bottlenecks of stress generators because it consumes memory bandwidth and CPU time. To avoid this problem, we use preconstructed templates for outgoing packets. A packet template is a preallocated buffer with already filled-in fields of all network protocol headers and payloads that are the same for a set of models. There are various templates that we use in DGM clients and DGM servers. Considering headers of Ethernet, IP, UDP, and GTP protocols, which are shown in light grey in Figure 4, most of their fields remain unchanged during the simulation. For example, source and destination ports of a UDP tunnel header can be fixed for the specific template. The processing of arriving packets also can involve undesirable buffer copying. Because incoming packets are processed in a sequential order of arrival, only one preallocated buffer for incoming packets is needed.

Raw Socket

To apply templates of the Ethernet datagrams explained above, we decided to use raw sockets. Such a socket allows us to communicate directly with the network driver when sending or receiving a packet; i.e., the usual protocol stack-handling, such as IP/TCP or IP/UDP processing, is avoided. In a Linux system, this socket is presented by a PF_PACKET protocol family starting from post-2.0 kernel releases.

Filter

Having opened a raw socket, the application has to process all incoming packets, even those that are not relevant to the application. To avoid this, the elegant solution is to

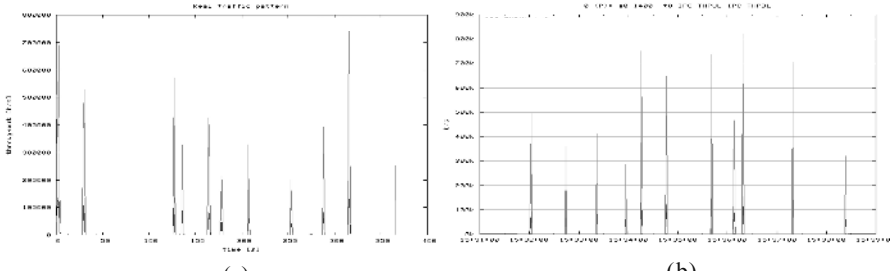
use a filtering mechanism right after the network driver to drop the packets which do not pass through the filter. Such filters are presented in [10] and [7].

Network Interface Card Driver
The usual way to notify the processor of a network event, such as the arrival or transmission of a packet, is done through interrupts. However, an interrupt is a too time-consuming operation to be called, e.g., 200,000 times/s. The alternative method, which is more efficient for high bandwidth networks, is known as polling. The possible hybrid implementation of NIC device driver combining both interrupt and polling techniques is presented in [4].

3 Results and Conclusions

We have presented a framework for implementing a stress workload generator. Our primary goal was combining real-like traffic patterns with the high efficiency of a generator.

Modelling Users and Applications
Our DGM implementation has built-in capabilities for simulation of different applications, which include WWW, unidirectional UDP and TCP audio streaming, E-MAIL, and MMS. All these models can be simulated concurrently in the system. The concurrent simulation of any combination of models gives a user the possibility to test the GSN nodes under lifelike conditions.



(a) (b)

Fig. 5. WWW down-link traffic: a - real, b - generated

Statistics of real (a) and generated (b) network traffic of a user reading the news with the Firefox browser in a fast fixed line network are presented in Figure 5. Real statistics were collected applying Ethereal for recording and Perl script for parsing WWW user session. Generated statistics were collected by DGM system. The application/user model parameters were modified according to this fast connection special case. The average values of the parameters were 30 seconds for reading time, 500B for request size, 10KB for reply size, and the number of embedded objects was around 20.

By setting up specific model parameters, predefined model behaviour is expected. However, it is typical that the network is utilized by various applications. Multiple traffic flows are transmitted through the same network elements. That leads to such situations as network congestions and queue overflow. In other words, all network participants, consuming limited network resources, affect each other.

Let us consider how web traffic competes with other applications for the network resources. We created one thousand WWW PDP contexts, and then ran one thousand UDP audio streaming models to produce additional network load. The snapshot of the seven minutes simulation can be seen in Figure 6.

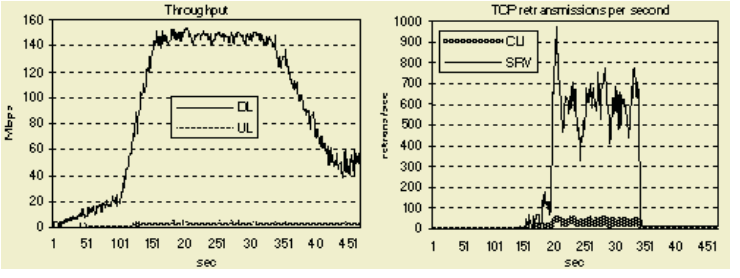


Fig. 6. WWW and UDP streaming traffic pattern

The downlink direction is more interesting to analyse because it is always more loaded for both model types. At the beginning only WWW models were created in the system. The number of TCP retransmissions was zero at that moment. After about 100 seconds of the simulation, UDP streaming users started their activity. The throughput peak of about 150 Mbps shows clearly this situation. As it can be seen in the second graph, after the UDP streaming traffic appeared in the network, the number of TCP retransmissions from web servers to mobile clients increased rapidly. This is usually caused by congestion or by other problems in the network.

In order to make proper tests, a real network has to be monitored and the parameters must be set according to actual conditions. Some studies on how to rapidly parameterise models have already been made [9]. Moreover, some assumptions about the evolution of the network and future requirements on the network should be taken in account.

The DGM performance tests were made with two PCs, each hosting one data node process. Both server stations had the same hardware configuration: Intel Xeon 2.40GHz, 512MB RAM, PCI-X (64 bits/100 Mhz), Intel 82544EI Ethernet controller. The current version of the DGM was able to produce about 500 Mbps with 90 000 packets per second in this configuration. Verification tests of DGM with real network elements were also conducted.

Future Work

We will consider the extension of the system with new models such as Voice over IP applications, which will also require a new model entity for Real-Time Transport

Protocol (RTP). Although new models can be easily added to the layered hierarchy, there is a small drawback, which is related to the integration of new models, to the current system. The implemented models are completely built into the system and the addition of new models requires small changes inside the core of the system.

The DGC and DGS are implemented as one-thread applications. Properly threaded applications can benefit hyper-threading and dual-core technologies by increasing their operational speed. Hence, the next step for improving the performance of the DGM would be to implement the data nodes with a few threads to utilize multi-threaded, multi-core, and multi-processor systems. In order to change the structure of the data nodes, the relation between the events within the system has to be found and the sets of independent events identified. The disjointed sets can be processed in parallel by using threads.

References

1. Bryans, J., Derrick, J.: Stochastic Specification and Verification. In *Proceeding of 3rd Irish Workshop on Formal Methods, Electronic Workshops in Computing,* pp. 20, July 1999.
2. Catapult Communications: *MGTS i3000.* Internet WWW-page, URL: http://www.catapult.com (30.11.2004).
3. D'Argenio, P. R., Katoen, J.P., Brinksma, E.: An algebraic approach to the specification of stochastic systems (extended abstract). In *Proceeding of the IFIP Working conference on Programming Concepts and Methods,* pp. 126-147, 1998.
4. Dovrolis, C., Thayer, B., Ramanathan, P.: HIP: Hybrid Interrupt-Polling for the Network Interface. In *ACM SIGOPS Operating Systems Review,* vol. 35, iss. 4, pp. 50 – 60, October 2001.
5. Hlavacs, H., Kotsis, G.: Modeling User Behavior: A Layered Approach. In *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems,* pp. 218-225, October 1999.
6. Hughes Software Systems: *Cutting edge GPRS Support Node Testing*. Internet WWW-page, URL: http://www.hssworld.com (30.11.2004).
7. Insolvibile, G.: Inside the Linux Packet Filter, Part II. *Linux Journal*, vol. 2002, iss. 95, pp. 7, March 2002.
8. ipNetfusion: *EAST for UMTS*. Internet WWW-page, URL: http://www.ipnetfusion.com (30.11.2004).
9. Lan, K.-C., Heidemann, J.: Rapid model parameterization from traffic measurements. In *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol.12, iss. 3, pp. 201-229, July 2002.
10. McCanne, S., Jacobson, V.: The BSD packet filter: A new architecture for user-level packet capture. In *Proceedings of the Winter 1993 USENIX Conference,* pp. 259—269, January 1993.
11. RADCOM: *Cellular Performer Analyzer*. Internet WWW-page, URL: http://www.radcom.com (30.11.2004).

Traffic Characteristics Based Performance Analysis Model for Efficient Random Access in OFDMA-PHY System

Hyun-Hwa Seo[1], Byung-Han Ryu[1], Choong-Ho Cho[2], and Hyong-Woo Lee[3]

[1] Department of Mobile Transmission Technology Research,
Electronics and Telecommunication Research,
Institute, Daejeon, Korea 305-350
{hhseo, rubh}@etri.re.kr
[2] Department of Computer and Information Science,
[3] Department of Electronics and Information Engineering,
Korea University, Choongnam, Korea 339-700
{chcho, hwlee}@korea.ac.kr

Abstract. Currently, IEEE 802.16a wireless MAN supports the contention based OFDMA-CDMA ranging subsystem for ranging operation (Initial Ranging, Periodic Ranging, Bandwidth Request)[1]. This system uses essentially the slotted ALOHA protocol. However, the number of code-slots/frame for random access transmission tends to be much greater than one and a simple Markov chain analysis may not be numerically feasible. At the same time, the frame size in number of code-slots may be dynamically adjusted based on the traffic load. In order to evaluate delay-throughput performance and stability measure of the random access protocol, we first examine the possible traffic load to be carried throughput the ranging subchannel. We the present performance analysis and numerical examples.

1 Introduction

In the IEEE 802.16a CDMA based OFDMA-PHY, Subscriber Stations(SS) access to Base Station(BS) that uses ranging subchannel to transmit code based Initial Ranging(IR), Periodic Ranging(PR) and Bandwidth Request(BR) in random access mode[1]. Initial ranging transmission is used in initial maintenance interval when beginning connection. Bandwidth-request transmissions are for requesting uplink allocations from the BS, and periodic ranging transmissions are sent for periodic adjustment for the system reflecting the channel conditions (timing and power adjustments). This procedure can reduce guard time through timing alignment on uplink and achieve power adjustment. For this, the SS shall choose randomly a ranging slot (i.e. OFDM symbol number, subchannel, etc.) as the time to perform the ranging, then it chooses randomly a ranging code and sends it to the BS(as a CDMA code). This system is basically similar to Slotted ALOHA[2], but the frame size (code-slot/frame) allocated for random access or

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 213–222, 2005.

ranging request is relatively big and can be altered by the dynamic adjustment of downlink and uplink subframe size on the MAC frame of OFDMA-PHY.

Although the performance of existing random access protocols such as slotted ALOHA [2], WLAN [3] and HIPERLAN[4][5] has been widely studied, to the best of our knowledge an analysis of the random access protocol employed by the OFDMA-CDMA ranging subchannel is not available in the open literature.

Considering that one of the characteristics of the system is that the frame size in number of code-slots can be dynamically adjusted according toe traffic load, we will briefly examine traffic load of IR, PR and BR. The estimated traffic load, then, can be used to appropriately allocate code-slots to different types of traffic according to the performance requirements.

This paper is organized as following. In section II, IEEE 802.16a OFDMA-CDMA environment is described. In section III, each traffic model is explained by generation rate of initial ranging, periodic ranging and bandwidth-request to do random access. In section IV, performance analysis is given. Numerical examples are examined in section V, followed by a conclusion in section VII.

2 Ranging Procedure [1]

System environment for analytic model in our study have followed std. IEEE 802.16a CDMA based OFDMA-PHY. It was designed based on TDD, and each frame consists of uplink and downlink subframes. The structure of ranging subchannel in an OFDMA MAC frame.

Initial ranging transmission is used in initial maintenance interval when beginning connection. Bandwidth-request transmissions are for requesting uplink allocations from the BS, and periodic ranging transmissions are sent for periodic adjustment for the system reflecting the channel conditions (timing and power adjustments).

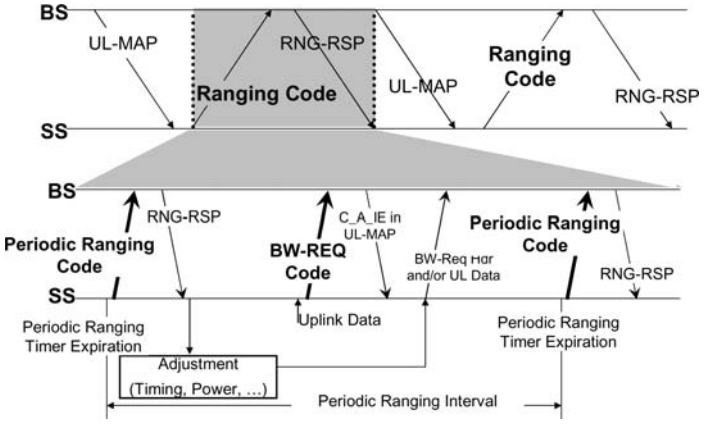


Fig. 1. Ranging procedure: initial ranging, periodic ranging, bandwidth-request

3 Traffic Modeling for Ranging Requests

3.1 Initial Ranging and Periodic Ranging

Initial ranging transmissions are used by SSs who want to synchronize system channel. The arrival of initial ranging packets can be modeled by a Poisson process whose rate depends on the number of SSs which are in idle state.

Periodic ranging is performed by active SSs periodically to adjust system parameters such as timing and power level. If the previous request was successful, the SS sends a request to BS at every 30 seconds. Hence, periodic ranging packets can be generated at every 30 seconds by an active SS.

3.2 Bandwidth-Request

Bandwidth-request transmissions are used to demand required bandwidth when a SS has data to send. Since bandwidth request transmission of each traffic class is different.

Real-Time Service. Real-time traffic is voice and video. A lot of studies of voice and video traffic model are reported until now. The ON/OFF model is the most simple and representative of voice traffic model. Here, bandwidth-request is generated at the beginning point of ON period (i.e. Bandwidth-request transmission is occurred at the beginning instant of every an ON period, and regenerated at every 2.35 seconds in average). Also, bandwidth-request transmission in video traffic[10],[11] can property piggyback subsequent transmission of data if first transfer succeeds.

HTTP. A model for web browsing in shown in [6]. In the model a session consists of alternating cycles of HTTP ON and HTTP OFF periods. A HTTP ON period, in turn, consists of a main object and a number of embedded objects. One main object includes multiple embedded objects. But, this is just downlink traffic model; uplink traffic model is not established yet. Therefore, we propose traffic model for uplink transmission.

First, the main/embedded objects in TCP are transmitted as IP packets, which cannot be larger than the maximum transfer unit (i.e., the MTU in Ethernet system is 1500 bytes). In response to a main/embedded object, ACKs are generated by a destination SS. To measure the number of ACKs according to the size of the main/embedded objects, we use TCP dump based ANYPA-LAN tool, and execute excommunications with different file sizes (1/3/5 Mbytes), different web servers, and different up and download times. In these experiments, we identified that an ACK is generated for every two data packets on the average which coincides with the *delayed ACK policy*[12] in TCP and *ACK-every-other-segment policy*[13][14]. Therefore, possible number of BRs including the number of ACKs is given by $\lceil$Main(or Embedded) Object size$/(2MTU)\rceil$ number of objects. And each BR size can be modeled as shown in Fig. 2.

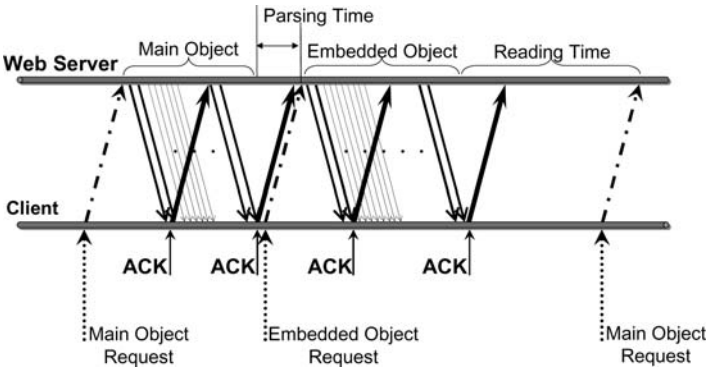


Fig. 2. Web traffic model at SSs

Best Effort Service. Messages arrives to mailboxes can be modeled by Poisson process. In case of receiving messages, the client host sends approximately a half times of the maximum number of packets[1] as shown in Fig. 2. On the other hand, in case of uplink emailing, the number of $\lceil$e-mail size$/MTU\rceil$ is transmitted because the whole message becomes a IP packet by capsulization. Therefore, from the client host, In cases of reception and transmission, approximately $\lceil$e-mail size$/MTU/2\rceil$ number of ACKs and $\lceil$e-mail size$/MTU\rceil$ number of IP packets are generated respectively.

4 Performance Analysis Model

To gain some understanding of the ranging subsystem, we formulate the analytical model by two steps as follow. First, we use a Discrete Time Markov Chain(DTMC)model for a more precise model. This model is divided into "Exact analysis" that applies Feller's classical occupancy theorem[6] and "Approximate analysis" (or "Asymptotic analysis") that uses an approximation by a Poisson distribution for the number of successful transmission in a frame. However, the DTMC model has numerical precision problem when the frame size is large. In order to circumvent the numerical precision problem, we ignore the discrete nature of the frame structure and use a Continuous Time Markov Chain(CTMC) which is a generalized M/M/1 queue. Therefore, the DTMC model is given for the derivation of CTMC and understanding the characteristics of each performance model under the small frame size. Finally, numerical results are shown illustrating the trading relations among the throughput-mean delay time and throughput-first exit time for system stability under the big frame size by using generalized M/M/1.

[1] The maximum number of packets is $\lceil$file size$/MTU\rceil$.

4.1 Assumptions

We assume the following.

Table 1. A Summary of System Parameters

Parameters	value
System	OFDMA/TDD
New ranging request arrival rate(λ)	$[0.02\ 0.3] * n$
The number of code-slot/frame(n)	$5\ 50, 12, 576$
The number of symbol and subchannel per frame	fixed
Multiple Access Interference(MAI) between codes	no
Retransmission randomization	no
Immediate feedback	yes
Frame period	5 ms

4.2 Mean Delay Time(MDT) Analysis; Discrete Time Markov Chain

For a more precise analysis, we derive a DTMC model, which is analyzed exactly by applying Feller's classical occupancy theorem [6] and approximately by an asymptotic application of the Poisson distribution.

Let $X^{(t)}$ be a random variable representing the total number of collided code slots in the t^{th} frame. Let us also define $\pi_k^{(t)}$ to be the probability of finding the system in channel state k in the t^{th} frame, that is,

$$\pi_k^{(t)} = P[X^{(t)} = k] \tag{1}$$

where, k is the number of collided code-slot.

We then define the probability vector $\underline{\pi}^{(t)}$ in the t^{th} frame as

$$\underline{\pi}^{(t)} = (\pi_0^{(t)}, \pi_1^{(t)}, ..., \pi_k^{(t)}, ...) \quad and\ so \quad \underline{\pi}^{(t+1)} = \underline{\pi}^{(t)} P \tag{2}$$

where, $P = p_{ij}$ is the transition probability matrix[2].

Under the assumption that the system is stable, the steady state probability can be written as $\pi_k = \lim_{t\to\infty} \pi_k^{(t)}$, which satisfies the following equations:

$$\underline{\pi} = (\pi_0, \pi_1, ..., \pi_k, ...), \underline{\pi} = \underline{\pi} P, \tag{3}$$

$$\sum_{k=0}^{k_{max}} \pi^{(t+1)} = 1 \tag{4}$$

From these, we calculate the sequence of values $\underline{\pi}$ and then the MDT($\overline{D}$) can be written as:

$$\overline{D} = \frac{\sum k\pi_k}{\lambda} \tag{5}$$

[2] transition probability is $p_{ij} = Pr[X^{(t+1)} = j | X^{(t)} = i]$.

In order to obtain p_{ij}, we use

$$p_{ij} = \sum_{ij} P[R_{new}] \cdot P[(i+a-j)R_{suc} | (i+a)R_{trn}] \tag{6}$$

where, $R_{new}, R_{suc}, R_{tran}$ are each request of a new arrivals, request of a successes, and request of a transmissions. In (6), the conditional probability of the successful code-slot in a frame is obtained by applying the classical occupancy theorem derived by Feller[6].

$$\frac{(-1)_{i+a-j} n!(i+a)!}{(i+a-j)! n^{i+a}} \times \sum_{l=i+a-j}^{min(n,i+a)} (-1)^l \frac{(n-l)^{i+a-l}}{(l-i-a+j)!(n-l)!(i+a-l)!} \tag{7}$$

where i is the number of collided code-slot in the previous frame, a is the number of new BRs or ranging requests, and n is the maximum number of code-slot per frame.

Also, as proven by Feller[6], if the number of successful code-slot in a frame is approximated by a Poisson distribution with parameter α $P[(i+a-j)R_{suc} | (i+a)R_{trn}]$ can be approximated as (9).

$$\alpha = (i+a)e^{-(i+a)/n} \tag{8}$$

$$P[(i+a-j)R_{suc} | (i+a)R_{trn}] = \frac{e^{-\alpha}\alpha^{i+a-j}}{(i+a-j)!} \tag{9}$$

4.3 MDT and FET; Continuous Time Markov Chain

If we regard the behavior of the continual ranging subframes as continuous time and do not distinguish between new bandwidth requests and retransmission requests, then we can assume that the bandwidth request arrival process is Poisson, the service times are exponentially distributed, and there is a single server. Consequently, the system can be modeled as a birth-death process with arrival rate $\lambda_k = \lambda$ and a state dependent service rate $\mu_k = \mu(= ke^{-k/n})$, where k includes the total number of backlogged and new bandwidth requests and n is the number of code-slot per frame. Under M/M/1 formulation, we first can get probability of the number of packets in system, k, p_k as

$$p_k = \frac{\lambda_{k-1}}{\mu_k} p_{k-1} = p_0 \left(\prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \right) = \left[1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} \frac{\lambda_i}{\mu_i} \right]^{-1} \left(\prod_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i} \right) \tag{10}$$

The MDT, $\overline{D}$ is given by

$$\overline{D} = \left(\frac{1}{\lambda} \sum_{k=1}^{\infty} k p_k \right) - \frac{1}{2} \tag{11}$$

where 1/2 accounts for the average waiting frame time for new bandwidth requests due to frame synchronization which was not included in the discrete time models.

Secondly, we consider the First Exit Time(FET). FET is the average time for the system to make a first exit into the unstable region starting from an initially empty system, which means that all code-slot in the frame are empty. In other words, the stability definition is as follows. A ranging subchannel is said to be stable if its service rate μ_k is greater than λ. Otherwise, the channel is said to be unstable.

Define k_{cr} to be the critical state of channel, which indicates the minimum k for the system to enter the unstable state for the first time.

$$k_{cr} = \min_{\mu_k < \lambda} k \tag{12}$$

Let the transition probabilities from state i to $i+1$ and from state i to $i-1$ be $\theta_i = \frac{\lambda_i}{\lambda_i + \mu_i}$ and $\overline{\theta}_i = 1 - \theta$, respectively. Then the mean transition time $t_{i,i+1}$ from state i to $i+1$ can be written as

$$t_{i,i+1} = \frac{1}{\theta_i}\left[\frac{\theta_i}{\lambda} + \overline{\theta}_i\left(\frac{1}{\mu_i} + t_{i-1,i}\right)\right] \tag{13}$$

where

$$t_{01} = \frac{1}{\lambda}. \tag{14}$$

Using (12),(13), we can express the FET $t_{0,k_{cr}}$ as

$$t_{0,k_{cr}} = \sum_{i=0}^{k_{cr}-1} t_{i,i+1} \tag{15}$$

5 Analysis Result

In order to verify the validity of the analysis and to provide a number of examples of how these analyses can be used to evaluate and compare the performance of the contention based OFDMA-CDMA ranging subchannel, this section presents some numerical and simulation results on the MDT and the FET by numerical computations and simulations using MATLAB. In the simulations of the ranging subchannel, we assume that the arrival process of access requests at each SS is Poisson with rate λ, and let the number of backlogged requests vary according to the specified protocol. The duration of each simulation run is 1000 seconds. Other basic assumptions are the same as given in Section 3. Using the equations in the previous subsections, we numerically compare with the results of the "exact analysis" and "approximate (or asymptotic) analysis" using DTMC. We also compare the analytical results from DTMC and CTMC with simulation results for a system with a small number of code-slots/frame. Applying the CTMC model to a system with a large frame size, we investigate the tradeoff between throughput and FET. Finally, we observe system stability to be determined by CTMC M/M/1 for estimating the arrival rate, the critical state, and the number of backlogged users.

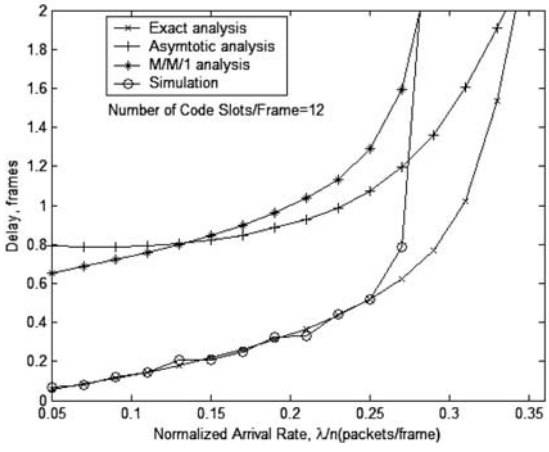


Fig. 3. Web traffic model at SSs

In Fig. 3, we compare the MDT from the three analytical models and simulations, under a small frame size (number of code-slots/frame n=12). As expected MDT increases as the normalized arrival rate λ/n(throughput) of bandwidth requests per code slot is increased. We observe that the results from the exact analysis and simulations are very close over a broad range of arrival rates below saturation, which are in turn lower than those from the approximate and M/M/1 models. We also observe that MDTs from the approximate analysis and M/M/1 are greater than the corresponding values of the exact analysis and simulations. We can conclude from these results that analysis using the approximate M/M/1 model yields more pessimistic results than the performance of the real system.

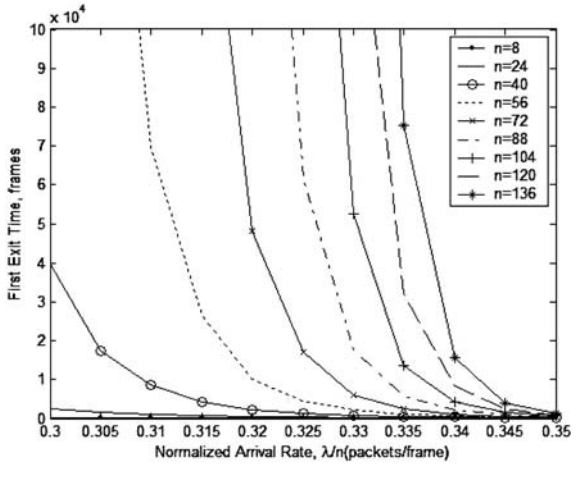


Fig. 4. Web traffic model at SSs

To further understand the stability characteristics of the system, we compute FET from (15) over the range of normalized arrival rates of interested identified above, i.e., between 0.3 and 0.35. Results are shown in Fig. 4. It is apparent that FET increases with the number of code-slots, n, and decreases as normalized arrival rate is increased. This is because the number of collisions increases when n is reduced or the arrival rate is increased, resulting in a decrease in FET. Currently, the expected number of code-slot in the OFDMA-CDMA ranging subchannel of IEEE 802.16a is about 5 to 12 symbols per frame and 5 to 48 codes/symbol, for a total of 25 to 576 available code slots per frame.

6 Conclusion

We have presented a theoretical analysis of the contention based OFDMA-CDMA ranging subchannel in IEEE 802.16a wireless MANs and validate the theoretical models by simulations. To evaluate delay-throughput tradeoffs and system stability, the MDT and FET performance of the ranging subchannel have been analyzed by means of an exact DTMC model that becomes untractable for systems with a large number of code slots/frame, and an approximate CTMC M/M/1 model based upon the DTMC model that can be applied to systems with any number of code slots/frame. Numerical results show that the approximate CTMC M/M/1 model yields somewhat pessimistic results that are nevertheless close to those obtained using the exact model. The approach presented in this paper is also applicable to other multi-channel multi-slot random access systems such as the random access channel in HIPERLAN2 [5].

In addition, radio environment has various restriction(i.e., limited resources, channel state, mobile, etc.). For example, even if channel status is good, it is difficult to assign radio resource efficiently. Most packets in Internet traffic are bigger in size than MTU. The packets in TCP are transmitted as IP packets. Then as a reply to successful transfer of packets that exceed MTU size, ACKs are generated in TCP. The ACK packets consume radio resource for random access. So, in case of Internet traffic modelling, must consider ACK certainly.

Performance analysis result of this paper can be performance reference mark for random access protocol in OFDMA-CDMA ranging system and traffic modeling can be used to do to prepare network resources appropriately in radio network.

References

1. IEEE Standard for Local and Metropolitan Area Networks: Part16: Air Interface for Fixed Broad Wireless Access Systems-Medium Access Control Modification and Additional Physical Layer Specification for 2-11 GHz", IEEE Std. 802.16aTM-2003 (2003)
2. Leonard, Kleinrock, Simon, S. Lam: Packet Switching in a Multiaccess Broadcast Channel:Performance Evaluation IEEE Transaction on Communications, Vol. Com-23, No.4, April (1975), pp.410-423

3. S. Simoens, P. Pellati, J. Gosteau, and K. Gosse: The Evolution of 5 GHz WLAN toward Higher Throughputs IEEE Wireless Communications, Vol. 10, pp.6-13, December 2003
4. G.-H. Hwang and D.-H. Cho: Adaptive random channel allocation scheme in HIPERLAN type 2, IEEE Communications Letters, vol. 6, pp. 40-42, January (2002)
5. G. Anastasi and L. Lenzini: HIPERLAN/1 MAC Protocol: Stability and Performance Analysis, IEEE Journal on Selected Areas in Communications, vol. 18, pp. 1787-1798, September (2000)
6. William Feller: An Introduction to Probability Theory and its Applications, Vol. 1 Comell University, Jan (1950)
7. Leonard Kleninrock: Queueing Systems, Vol.1:Theory, 417 pp. John Wiley and Sons-Interscience, (1975)
8. Alex Brand, Hamid Aghvami: Multiple Access Protocols for Mobile Communications: GPRS, UMTS and Beyond, WILEY, (2001)
9. Traffic Models for IEEE 802.20 MBWA System Simulations, IEEE C802.20-03/66
10. 1xEV-DV Evaluation Methodogy, 3GPPe/TSG-C. R1002. Document available in 802.20 drop-box
11. Hun-Jeong Kang, Myung-Sup Kim, and James W. Hong: Streaming Media and Multimedia Conferencing Traffic Analysis Using Payload Examination, ETRI Journal, vol. 26, no.3, June (2004), pp.203-217.
12. Gary R. Wright W. Richard Stevens: TCP/IP Illustrated, Vol. 2, Addison-wesley professional computing series.
13. [RFC793] Transmission Control Protocol
14. [RFC879] The TCP Maximum Segment Size and Related Topics

Collision Reduction Random Access Using *m*-Ary Split Algorithm in Wireless Access Network

You-Chang Ko[1], Eui-Seok Hwang[2], Jeong-Jae Won[3], Hyong-Woo Lee[4], and Choong-Ho Cho[5]

[1] LG Electronics Inc. Mobile Handset R&D Center
ycko@lge.com
[2] Univ. of Washington Dept. of Electrical Engineering
eui@ee.wahsington.edu
[3] Univ. of British Columbia Dept. of Electrical and Computers Engineering
wonjj@ece.ubc.ca
[4] Korea Univ. Dept. of Electronics & Information Engineering
hwlee@korea.ac.kr
[5] Korea Univ. Dept. of Computer & Information Science
chcho@korea.ac.kr

Abstract. In the high performance radio access networks, the number of random access channels can be used for mobile stations to transmit their bandwidth requests in contention mode via multiple random access. In this paper a collision reduced random access scheme based on *m*-art split algorithm in the centralized medium access control protocol is presented. In this method the splitting algorithm is used in two fold. On one hand the whole random access channels are exclusively separated into two parts; one is for initial contention where only initial access mobile stations can be served and the other is for retransmit contention where the mobile stations whose initial random access was not successful can join for their retransmission. On the other hand the *m*-ary split algorithm applied in the retransmit contention area to resolve the collisions. By doing so the proposed scheme achieves considerably greater performance in terms of maximum throughput, mean access delay, and delay jitter, which is one of the important criteria for real-time traffic. Through numerical examples and computer simulations the effect of the various parameters of the algorithm, initial number of random access report(s) and the split size *m*, on the system performance is examined.

Keywords: Wireless Access Network, Random Access, *m*-ary split algorithm, Multimedia QoS.

1 Introduction

ETSI broadband radio access network (BRAN) project is currently developing standard for various types of wireless broadband access networks. High performance radio local area network type 2 (HiperLAN/2) is one of these proposed to operate in

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 223–233, 2005.

the 5GHz band which provides high-speed communications between mobile terminals and various broadband infrastructure networks[1]. The medium access control (MAC) protocol is based on dynamic time division multiple access/time division duplexing (TDMA/TDD). In a HiperLAN/2, fixed-sized messages are transmitted based on the scheduling performed by an access point (AP) or a central controller (CC) either in centralized mode or direct mode, respectively. Without loss of generality, we will only concern with the AP for the remainder of this paper.

Mobile terminals (MTs) report their uplink transmission requirements in Resource Request (RR) messages to the AP. The AP then allocates the resources according to the RRs and sends the allocation of the resources via Resource Grant (RG) messages. When an MT does not have an uplink transport channel allocated to transmit its RR, it transmits its RR in one of the available Random Channels (RCHs) in a contention mode. The outcome whether the MT's access attempt in a frame is successful or not is informed by the AP via Access Feedback Channel (ACH) in the subsequent frame. The number of RCHs may vary according to traffic load in the RCHs. A fixed-sized frame consists of downlink and uplink transport channels among which are the RCHs. Although an RCH slot is only 9 bytes long, because it uses the most robust modulation/coding (BPSK, 1/2 rate) and because the guard time between successive RCHs is usually not negligible [1,3], unnecessarily large number of RCHs can waste the channel resources. On the other hand, insufficient number of RCHs cause prolonged access delay and channel instability common to most contention schemes.

In [1], the problem of channel instability is partially solved using a version of the binary exponential backoff algorithm for retransmissions. Roughly, the contention window size (CW) of an MT in number of RCH slots is doubled after every collision experienced by the MT until the corresponding RR succeeds or the CW reaches the maximum size, thereby, spreading the retransmission attempts. The number of RCHs in a frame is recursively determined in [2] using the number of idle and successful RCHs in the previous frame. Compared with a scheme which uses a fixed number of RCHs, their scheme reduces access delay and possibly achieves higher maximum channel throughput. However, according to their simulations in [2], the maximum normalized channel throughput which is defined as the fraction of successful RCHs is still limited to $1/e$. Moreover, the access delay variation is rather large.

In this paper, we propose an m-ary split algorithm for dynamically adapting the number of RCHs in the current frame based on the observation of the RCHs in the previous frame. Essentially, by allocating m RCHs for each collided RCHs in addition to the RCHs allocated for RRs generated by newly arriving MTs, the maximum normalized throughput can be increased to approximately 0.43 and the delay variation, which is one of the important system criteria for real-time traffic are substantially reduced compared with the schemes using the binary backoff algorithm [1,2]. It is, however, necessary to modify the content of the ACH slot such that the outcome as to whether an RCH contains a collision or not is informed to the MTs instead of the outcome as to whether it contains a success or not. Also, retransmission algorithm is simplified by making the binary exponential backoff unnecessary.

This paper is organized as follows. In section 2, we describe the proposed m-ary split algorithm. Analysis of the mean access delay and throughput of the proposed

scheme is presented in section 3. Since the analysis is exact only if the maximum number of RCHs allocated in a frame is not limited and does not provide the delay variance, we performed computer simulations. In section 4, we present numerical examples and computer simulation results demonstrating the superiority of the proposed algorithm compared with the previous algorithm based on the binary exponential backoff algorithm in terms of mean and variance of delay and throughput. We also examine the effect of the degree of split, *m*, and the number of RCHs allocated for initial access on the performance. Finally we end with conclusions in section 5.

2 *M*-Ary Split Algorithm

As described in the introduction, the AP allocates a number of RCHs for random access transmission of Resource Requests (RRs) by mobile terminals (MTs). The set of RCHs which are placed at the end of a frame is referred to as *RCH train*. In the proposed scheme an RCH train as shown in Fig. 1 consists of $N_a(\geq 1)$ RCHs for transmission of the RRs generated by newly arriving MTs and N_c RCHs for retransmission of the collided RRs attempted in the previous frame.

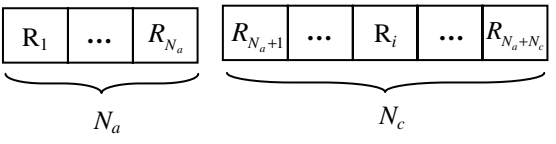


Fig. 1. The RCH train structure of the proposed algorithm

According to [3], which evaluates throughput of HiperLAN/2 MAC protocol taking all guard time spaces into account, the maximum available number of RCHs in an AP would be much less than those defined in [1]. We, therefore, limit the maximum RCH train size to R_{MAX}. Then, the number of RCHs in the $(t+1)^{th}$ frame, $r(t+1)$, is given by equation 1.

$$r(t+1) = \min[\, N_a + m*c(t),\, R_{MAX}], \tag{1}$$

where $m(\geq 2)$ is the degree of split and $c(t)$ is the number of collided RCHs in the t^{th} frame. For a simple description of the proposed algorithm, we will assume that $R_{MAX} - N_a$ is divisible by m. The algorithm is as follows.

1. An RR arriving during the t^{th} frame will be transmitted in one of the N_a RCHs chosen at random in the subsequent frame, $(t+1)^{th}$ frame.
2. An MT which transmitted an RR in the t^{th} frame is notified of the outcome of its attempt in the ACH. Moreover, it is able to find the number of RCHs involved in a collision prior to the RCH in which it transmitted its RR. If its contention is unsuccessful one of the followings are performed.

a) If $N_a + m(\theta_i + 1) \le R_{MAX}$, it retransmits its RR during the $(t+1)^{th}$ frame in one of the RCHs chosen at random in $[R_{N_a+m\cdot\theta_i+1}, R_{N_a+m(\theta_i+1)}]$, where i is the index of RCH which the MT has accessed in the previous MAC frame and θ_i is the number of collided RCH indices which are less than i, $\theta_i \ge 0$.

b) Otherwise, it attempts retransmission in one of the N_a RCHs during the $(t+1+\delta)$ frame, where δ is a uniform random variate whose interval is chosen appropriately depending on the traffic intensity.

In the following equations RF(i) is the retransmission function which determines its position of RCH in the subsequent frames.

$$\text{Initial attempt: random access within } [1, N_a] \tag{2}$$

$$\begin{aligned} &\text{RF}(i) = \text{random access within } [R_{N_a+m\cdot\theta_i+1}, R_{N_a+m(\theta_i+1)}] \\ &\qquad \text{if } N_a + m(\theta_i + 1) \le R_{MAX} \text{ or,} \\ &\text{random access within } [1, N_a] \text{ after random delay} \\ &\qquad \text{if } N_a + m(\theta_i + 1) > R_{MAX} \end{aligned} \tag{3}$$

Figure 2 shows an example how the proposed method works based on m-ary split algorithm with N_a=2, m=3, and R_{MAX}=11. In the figure S, C, and I denote success, collision, and idle respectively. Suppose that collisions occur at R_1, R_3, R_5, and R_6 in the t^{th} MAC frame and Q_1, Q_3, Q_5, and Q_6 are the set of the MTs which have collided at R_1, R_3, R_5, and R_6 respectively. Then $|Q_i| \ge 2$ and the MTs in Q_1, Q_3, Q_5, and Q_6 in the t^{th} MAC frame will contend again in the $(t+1)^{th}$ MAC frame within N_c area. R^m_i is the split size in $(t+1)^{th}$ MAC frame for the MTs in Q_i. The MTs in set Q_6 cannot retransmit their RRs within the $(t+1)^{th}$ MAC frame due to the limitation of the maximum RCH size. They retransmit their RRs in one of the N_a RCHs in the $(t+1+\delta)^{th}$ MAC frame. Their RRs are treated as if they are newly arrived RR in the $(t+\delta)^{th}$ MAC frame.

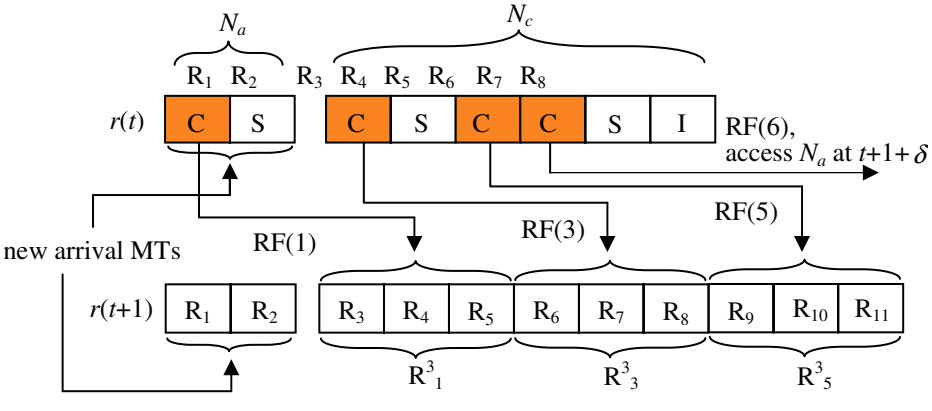


Fig. 2. RCH structure in the t^{th} and the $(t+1)^{th}$ MAC frame for N_a=2, m=3, R_{MAX}=11

3 Anaplysis of the Mean Access Delay and Throughput

The following notations and assumptions are used.

1. Let A_t denote the number of RR packets arriving during the t^{th} frame. We assume that the sequence of A_t's are independent and identically distributed (iid) random variables. Although it is not essential for the analysis, we assumed that A_t is Poisson distributed with mean λ_t RR packets/frame. λ_t denotes the offered load in the AP's perspective whereas λ represents the offered load by each MT. As stated in the previous section, all the newly arriving RRs are transmitted in an RCH during the subsequent frame.
2. For the sake of tractability, we assume that there is no limit on the number of RCHs in a frame. That is, $R_{MAX}=\infty$. This restriction will be relaxed in the computer simulations.
3. Let $D(n)$ denote the sum of mean delays in number of frames experienced by the RRs conditioned on that they belong to a group of n RRs initially transmitted in one of the N_a RCHs. We note that $D(0)=D(1)=0$.
4. Let $N(n)$ denote the mean of total number of RCHs required to resolve a collision involving n RRs. We note that $N(0)=N(1)=0$.

3.1 Mean Access Delay

We have the following recursive equation for $n \geq 2$.

$$D(n)=\sum_{s_1+\ldots+s_m=n}\left[\alpha(s_1,s_2,\ldots,s_m;n)\cdot\left\{n+\sum_{i=1}^{m}D(s_i)\right\}\right].$$

Solving for $D(n)$ in the above equation, we obtain

$$D(n)=\frac{n+\sum_{0\le s_1\le n-1,\ldots,0\le s_m\le n-1}\left[\alpha(s_1,s_2,\ldots,s_m;n)\cdot\left\{\sum_{i=1}^{m}D(s_i)\right\}\right]}{1-m^{-n+1}},\quad \text{for } n\geq 2$$

where $\alpha(s_1,s_2,\ldots,s_m;n)$ is the multinomial probability function. That is, if n items are placed randomly among m bins, the probability that the first bin contains s_1 items, the second bin contains s_2 items, …, and the m^{th} bin contains s_m items is $\alpha(s_1,s_2,\ldots,s_m;n)$.

We have

$$\alpha(s_1,s_2,\ldots,s_m;n)=\frac{n!}{s_1!\cdot s_2!\cdot\ldots\cdot s_m!}(p_1^{s_1}\cdot p_2^{s_2}\cdot\ldots\cdot p_k^{s_m}) \quad = \quad \frac{n!\cdot m^{-n}}{s_1!\cdot s_2!\cdot\ldots\cdot s_m!} \tag{4}$$

Here, p_i is the probability that an RR chooses the i^{th} RCH among the m RCHs allocated for the group of RRs involved in a collision. Since an RR chooses one of the m RCHs at random, we have $p_i=1/m$, and the last equality results. We recursively obtain $D(n)$ (n=2,3,…) starting from the initial conditions $D(0)=D(1)=0$. We

define, $\overline{D}$, the mean delay as the time from the first access attempt to the beginning of the frame in which the access is successful. Finally, we have

$$\overline{D} = \frac{N_a}{\lambda_t}\sum_{i=0}^{\infty} D(i)\cdot P(i), \quad \text{where } p(i) = e^{-\frac{\lambda_t}{N_a}}\cdot\frac{\left(\frac{\lambda_t}{N_a}\right)^i}{i!},\ i = 0,1,2,...$$

In the above equation *p(i)* is the probability that *i* RRs attempt transmission in one of the N_a RCHs.

3.2 Throughput

Recalling that $N(n)$ is defined as the additional RCHs needed to resolve a collision involving n RRs, we have $N(0)=N(1)=0$. A similar recursive equation can be obtained for $N(n)$ as that for $D(n)$.

$$N(n) = \sum\left[\alpha(s_1, s_2, ..., s_m; n)\cdot\left\{m + \sum_{i=1}^{m} N(s_i)\right\}\right].$$

Solving for $N(n)$ using the above equation, we have for $n \geq 2$,

$$N(n) = \frac{m + \sum_{0\leq s_1\leq n-1,...,0\leq s_m\leq n-1}\left[\alpha(s_1, s_2, ..., s_m; n)\cdot\left\{\sum_{i=1}^{m} N(s_i)\right\}\right]}{1 - m^{-n+1}}. \tag{5}$$

We define, $\overline{N}$, as the mean number of RCHs in an RCH train. Then, we have

$$\overline{N} = N_a\cdot\left(1 + \sum_{i=0}^{\infty} D(i)\cdot P(i)\right), \quad \text{where } p(i) = e^{-\frac{\lambda_t}{N_a}}\cdot\frac{\left(\frac{\lambda_t}{N_a}\right)^i}{i!},\ i = 0,1,2,... \tag{6}$$

We will use the resource utilization (normalized throughput), ρ, as defined in equation 11 of reference [2]. We, finally, obtain the normalized throughput as

$$\rho = \frac{\textit{total number of successful access attempts}}{\textit{total number of allocated RCHs}} = \frac{\lambda_t}{\overline{N}} \tag{7}$$

4 Numerical Results and Simulations

In this section, we present the numerical results of the proposed method in terms of mean access delay, delay variance, and the normalized throughput. The results here are based on the assumptions that transmission failures due to wireless channel errors are negligible and can be ignored and RR arrivals obey Poisson distribution with

mean λ packets/frame/MT. We also assumed that the number of MTs belonging to an AP is 50.

Figure 3 and Figure 4 show the effect of varying N_a and/or RCH split size, *m*, on the normalized throughput. As seen in the figures the proposed algorithm can achieve maximum normalized throughput of about 0.43 which is considerably higher than about 0.36 as reported in the previous work [2]. In Figure 3, we fix N_a=1 or N_a=5 and vary the split size from 2 to 5. It is observed that the maximum utilization does not improve beyond *m=2*. However, compared with the case when *m=2* the decrease in the throughput when the offered load is greater than the optimum load at which the maximum throughput is achieved is more gradual when *m* is greater than 2. We note that the gradual decrease in throughput has a desirable effect on channel stability.

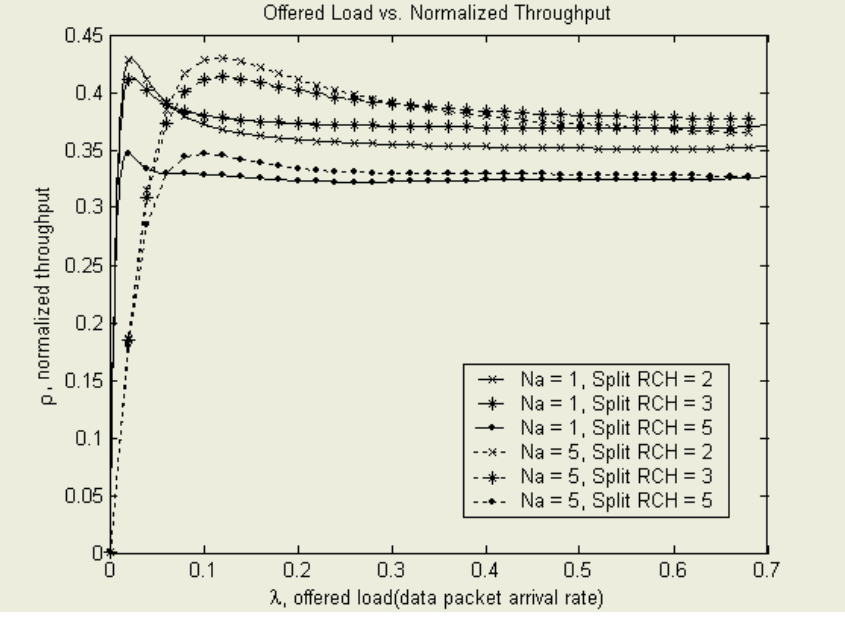


Fig. 3. Offered load vs. resource utilization(throughput) as *m* varies(numerical results)

The results also show that the maximum throughput does not depend on the value of N_a; however, the peak of the throughput curves occur at higher offered load when N_a is larger. At a region where the offered traffic is high, increasing the number of initial RCHs, N_a, is more effective for enhancing system performance than increasing RCH split size, *m*. Although the system suffers from performance degradation in terms of maximum normalized throughput as RCH size increases in general, the system throughput becomes greater after a certain offered load for the case of split sizes 2 and 3. For example, when N_a=1, this threshold of the offered load is approximately 0.07, while it is about 0.32 when N_a=5. In Figure 4, we fix *m*=2 or *m*=5 and vary the N_a from 1 to 5. We have a similar observation as in Figure 3.

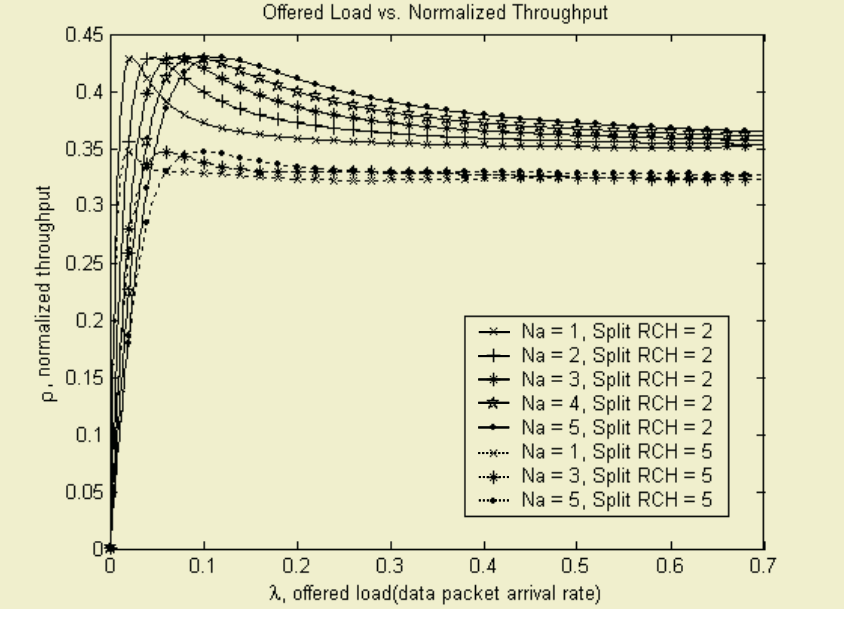


Fig. 4. Offered load vs. resource utilization(throughput) as N_a varies(numerical results)

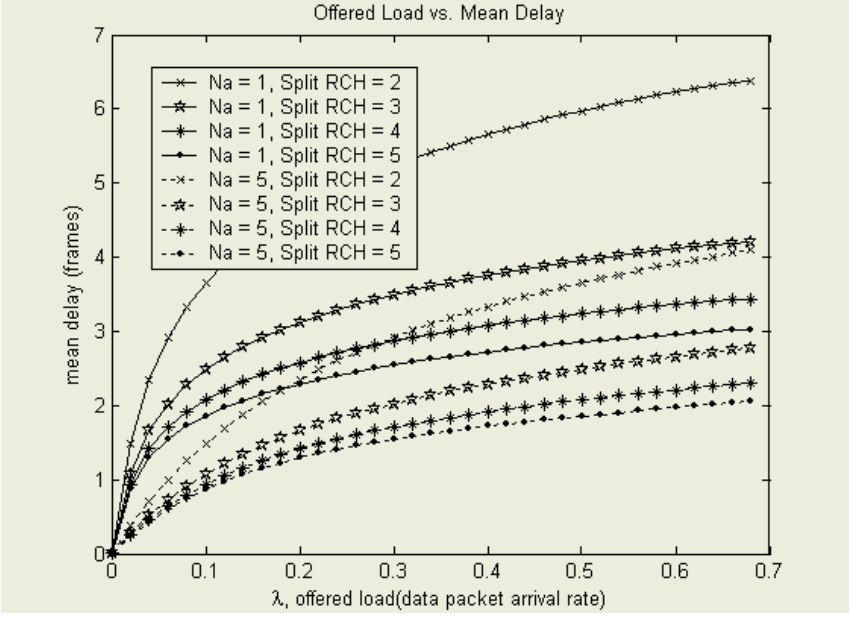


Fig. 5. Offered load vs. mean delay(numerical results)

Figure 5 presents the mean access delay in number of MAC frames required for the users until they successfully transmit their RRs. With various RCH split size, *m,* from 2 to 5, we plot the mean access delay vs. offered load, λ, when initial number of RCHs, N_a is between 1 and 5. The numerical results show that the mean access delay can be decreased by increasing the split size *m* as well as the initial number of slots N_a. By comparing the 5th curve(N_a=1 & Split RCH=5) and the 6th curve(N_a=5 & Split RCH=2), we note that appropriate choice of parameter values (N_a and *m*) has a significant influence on the delay vs. load performance. We can take advantage of adaptively varying N_a and *m*. Experiments also show that the biggest mean delay decrease can be achieved when *m* is increased from two to three. However, increasing *m* size to reduce mean delay would not always be available due to the fact that the number of available RCHs in MAC frame could be limited by the system parameters such as guard times which can be imposed in many places in a MAC frame[3].

The graphs in Figure 6 shows mean delay variance in number of MAC frames and are plotted by computer simulation when N_a is increased from 1 to 5 as *m* varies. As seen in the graphs, by increasing *m* the mean delay variance can be significantly reduced in general. A similar behavior as in mean delay graphs in Figure 5 is observed such that increasing *m* from 2 to 3 is most significant in decreasing the mean delay variance. Simulation results also show that increasing N_a is still effective to reduce mean delay variance for arbitrary *m*. However, the amount of decrease in the mean delay variance by increasing N_a reduces as offered load increases. When referred to [4], which shows delay variance simulation result of the scheme proposed by [2]; it requires approximately 150 frames when offer load is about 0.3, our proposed method substantially reduce the jitter time which is one of important criteria for real-time traffic.

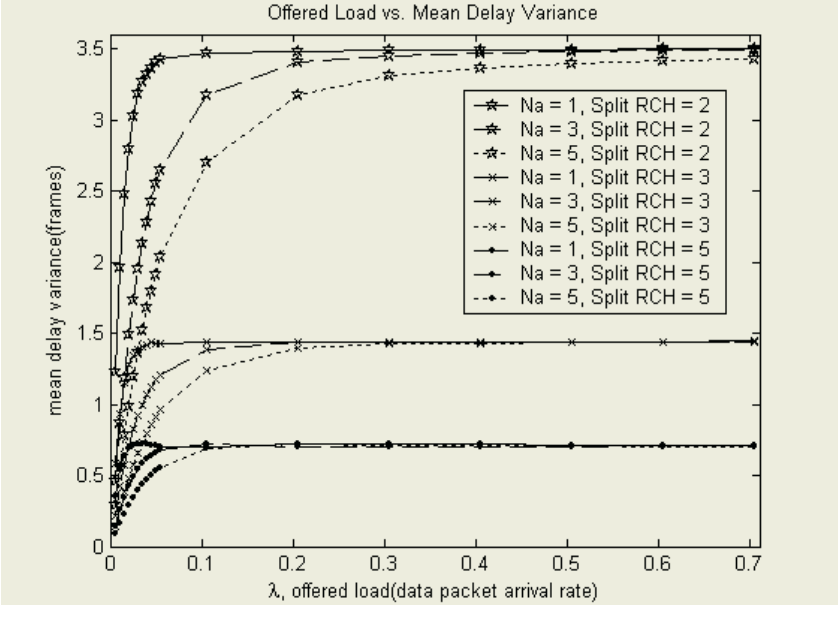


Fig. 6. Offered load vs. mean delay variance(simulation results)

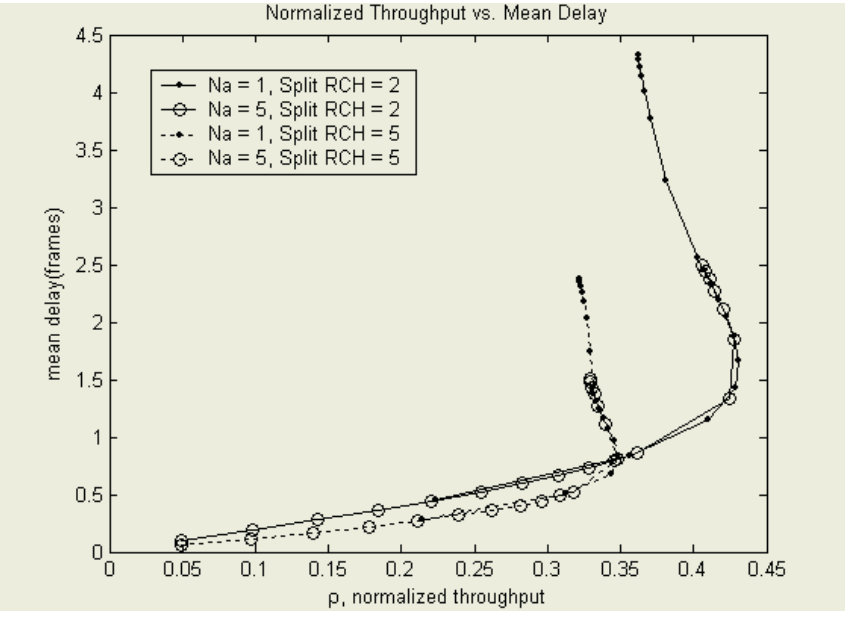


Fig. 7. Resource utilization(throughput) vs. mean delay(simulation results)

In the normalized throughput's perspective Figure 7 illustrates the mean delay characteristics in MAC frames by computer simulation. As seen the two cases when m=2 and m=5, delays are at most less than 4.5 and 2.5 frames respectively. We see generally that the more N_a or m size is the lower mean delay is. We also note the delay difference between minimum and maximum N_a size for each m size; when m=2 there are approximately 2 frames difference between the end of circle mark(N_a=5) and the end of dot mark(N_a=1), whereas when m=5 there are approximately 1 frames difference between the end of circle mark(N_a=5) and the end of dot mark(N_a=1). When compared to the results in [2], with the proposed method when N_a=m=5 we can achieve nearly 10 times less delay frames while preserving the same throughput. Moreover the proposed method can increase throughput up to approximately 0.44 when m=2, which is actually 26% increase compared to the previous method in [2], at the expense of slightly additional delays but remaining more than 6 times less delay frames still.

5 Conclusions

In this paper, we proposed a new method based on m-ary split algorithm to control the number of RCHs dynamically in an RCH train of a HiperLAN/2 MAC frame. The structure of RCH train consists of N_a RCHs for those MTs who access the system for the first time and N_c RCHs for those MTs who retransmit their request after experiencing collision in the previous frame. By letting N_c to be m times number of

collided RCHs in the previous slot, we were able to increase the maximum normalized throughput. We also analyzed the mean access delay, delay variance and the normalized throughput. Through numerical examples using mathematical analyses and computer simulations with various values of N_a and the RCH split size, m, we have shown that the proposed method can achieve much better performance in terms of throughput and delay performance compared with the previous work in [2], and that the proposed scheme is suitable for guaranting QoS for not only non-real-time traffic but also real-time traffic, that is, QoS of multi-media traffic in wireless LAN. We observed that as the traffic load increases increasing N_a area is generally more effective than increasing m. It was also observed that the best performance is achieved with m=2 or 3 regardless of N_a and that it is effective to reduce mean delay frame and its variance by increasing either m or N_a. It will be interesting to find an adaptive scheme for switching the value of m as well as for balancing the size of N_a and m depending on the observed traffic intensity for optimum performance. Although the proposed algorithm is proposed to Hiperlan2 system in this paper to compare with the previous applied method in [2] it can be able to apply to most of wirelss packet access networks where resources are allocated by MT's random access.

References

[1] HIPERLAN Type2; Data Link Control(DLC) Layer; Part1: Basic Data Transport Functions, Broadband Radio Access Networks(BRAN), ETSI TS 101 761-1 V1.3.1(2001-12).

[2] Gyung-Ho Hwang and Dong-Ho Cho, "Adaptive Random Channel Allocation Scheme in HIPERLAN Type2", IEEE Communications Letters, Vol.6, No.1 Jan 2002.

[3] You-Chang Ko, Eui-Seok Hwang, Jeong-Shik Dong, Hyong-Woo Lee and Choong-Ho Cho, "Throughput Analysis of ETSI BRAN HIPERLAN/2 MAC Protocol taking Guard Timing Spaces into Consideration", 9th International Conference on Personal Wireless Communications 2004, Deft Netherlands, September 2004.

[4] Eui-Seok Hwang, Jeong-Jae Won, You-Chang Ko, Hyong-Woo Lee and Choong-Ho Cho, "Random Channel Allocation Scheme in HIPERLAN/2", to appear Pacific-Rim Conference on Multimedia 2004, Tokyo Japan, November 2004.

[5] D. Petra, A. Kramling, and A. Hettich, "MAC Protocol for Wireless ATM: Contention Free versus Contention Based Transmission of Reservation Requests", 7th IEEE PIMRC, 1996.

[6] Benny Van Hooudt and Chris Blondia, "Analysis if an Identifier Splitting Algorithm Combined with Polling (ISAP) for Contention Resolution in a Wireless Access network", IEEE Journal on Selected Area in Communications, Vol.18, No.11, Nov. 2000.

[7] Michael J. Markowski and Adarshpal S. Seti, "Real-Time Wireless Communication Using Splitting Protocols", IEEE GLOBECOM Vol.3, Nov. 1997.

[8] Yu Gong and Michael Paterakis, "A Robust Random Multiple-Access Algorithm for Packet Transmissions over Noisy Channels with Error Memory", IEEE Transactions on Communications Vol.42, No.9, Sep. 1994.

[9] Alberto Leon-Garcia, "Probability and Random Processes for Electrical Engineering", Chapter 2 Basic Concept and Probability Theory, Addison-Wesley, 1994 Second Edition.

Simple, Accurate and Computationally Efficient Wireless Channel Modeling Algorithm

Dmitri Moltchanov, Yevgeni Koucheryavy, and Jarmo Harju

Institute of Communication Engineering,
Tampere University of Technology,
P.O.Box 553, Tampere, Finland
{moltchan, yk, harju}@cs.tut.fi

Abstract. We propose simple and computationally efficient wireless channel modeling algorithm. For this purpose we adopt the special case of the algorithm initially proposed in [1] and show that its complexity significantly decreases when the time-series is covariance stationary binary in nature. We show that for such time-series the solution of the inverse eigenvalue problem returns unique transition probability matrix of the modulating Markov chain that is capable to match statistical properties of empirical frame error processes. Our model explicitly takes into account autocorrelational and distributional properties of empirical data. We validate our model against empirical frame error traces of IEEE 802.11b wireless access technology operating in DCF mode over spread spectrum at 2Mbps and 5.5 Mbps bit rates. We also made available the C code of the model as well as pre-compiled binaries for Linux and Windows operating systems at http://www.cs.tut.fi/~moltchan.

1 Introduction

The grow of the Internet and increase in the number of users that wish to access Internet services 'anytime and anywhere' stimulate development of wireless access technologies. Consequently, air interface is expected to be an integral part of next-generation telecommunications networks.

Due to movement of a mobile user, the propagation path between the transmitter and a receiver may vary from simple line-of-sight (LOS) to very complex ones. To estimate performance of wireless channels, propagation models are often used. Basically, we distinguish between the large-scale and small-scale propagation models (see [2] for review). The former models focus on predicting the received local average signal strength over large separation distances between the transmitter and a receiver and do not take into account rapid changes of the received signal strength. As a result, they cannot be effectively used in performance evaluation studies. Propagation models characterizing rapid fluctuations of the received signal strength over short time duration are called small-scale propagation models. Due to implicit incorporation of small-scale mobility, these models provide better characterization of the received signal strength.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 234–245, 2005.

A major consequence of propagation characteristics is that bit and frame errors of wireless channels are usually not independent but correlated [2, 3, 4]. Techniques such as forward error correction (FEC) and automatic repeat request (ARQ) may allow to recover from these errors. To study performance of these techniques wireless channel models at the data-link layer are needed.

Recently, it was shown that the frame error process of the wireless channels can be sufficiently well represented by doubly-stochastic Markov modulated process [5, 6, 7]. Such a model provides a useful trade-off between complexity of the model and accuracy of fitting to statistical data. However, modeling algorithms developed for this model do not *explicitly* take into account the second-order properties of statistical data leading to incorrect representation of the memory of the frame error process (see [8, 9] among others).

In this paper we develop simple and computationally efficient model that is capable to capture important statistical characteristics of frame error traces. Based on statistical analysis we show that the (normalized) autocorrelation function (NACF) of frame error traces exhibits nearly geometrical behavior and approximate such a behavior by a geometrically distributed component. To find a suitable Markovian model approximating a given NACF, we formulate and solve the inverse eigenvalue problem, i.e. we find such a Markov process that its transition probability matrix posses a predefined eigenvalue approximating behavior of empirical NACF. We also show that there is unique Markov modulated process approximating histogram of relative frequencies of the frame error trace and empirical NACF. The associated fitting algorithm is extremely simple, fast, and computationally efficient. We believe that the proposed model is also suitable for frame error observations on different technologies. We made available the C code of the proposed model as well as pre-compiled binaries for Linux and Windows operating systems at http://www.cs.tut.fi/~moltchan.

Our paper is organized as follows. Background and related work are considered in Section 2. Setup of experiments and statistical characteristics of frame error traces are analyzed in Section 3. The proposed model is formulated in Section 4. Algorithm and practical implications are considered in Section 5. In Section 6 numerical examples are given. Conclusions are drawn in last section.

2 Background and Related Work

There was a lot of effort aimed on developing a suitable model for frame errors at the data-link layer. In [9], to capture statistical characteristics of error traces, authors carried out statistical analysis of IEEE 802.11b frame error traces and used a number of models, including hidden Markov model (HMM), and higher-order Markov chains. They also showed that FSMC may fail to model frame error traces accurately. Statistical analysis of frame error traces was also carried out in [10] and that was the first paper where dependence between successive frame errors has been considered in terms of NACF. It was suggested that with increasing of the number of states, first-order Markov chains are capable to

capture autocorrelation properties of frame error traces. Particularly, in [8] a 512-states first-order Markov chain was introduced to model IEEE 802.11b frame error traces. Due to a large number of states, such models are only suitable for simulation studies of information transmission over wireless channels. In a number of papers [5, 6, 7] Zorzi and Rao have shown that two-states Markov modulated model is sufficient to capture frame error statistics at the data-link layer.

In this paper we propose a model for IEEE 802.11b frame error traces. We found our traces to be covariance stationary ones and explicitly capture statistical characteristics including empirical ACF and probability distribution function (PDF). The proposed approach is not only applicable to frame error traces but can be used to model bit error traces of wireless channels as long as the empirical ACF can be approximated by geometrically distributed component.

3 Experiments and Statistical Data

3.1 Setup and Background Information

In this study we use IEEE 802.11b frame error traces available from [11]. Setup and background information related to collection of traces are given in [8, 10]. In this subsection we review major details that are important for our work.

According to experiments, there were three nodes, two mobile nodes and access point (AP), involved in communication as shown in Fig. 1. Experiments were carried out in office environment. The server and AP were nearby each other within a line-of-sight (LOS). The client and AP were in 'no LOS' environment as there was a wall between them. The communication between AP and client was of interest. All nodes involved in communications used distributed coordination function (DCF) of IEEE 802.11.

The communication was as follows. According to IEEE 802.11, one way transfer of data packet involves Request To Send (RTS) - Clear To Send (CTS) - Data (DATA) - Acknowledgement (ACK) exchange of packets. Every second the server transmitted a data packet of 512 bytes in length to AP using RTS-CTS-DATA-ACK exchange. Upon receiving a packet, AP transmitted it to the client using RTS-CTS-DATA-ACK exchange. In the course of experiments retransmission of incorrectly received packets was disabled, however, normal procedures for IEEE 802.11 DCF contention access were used [12].

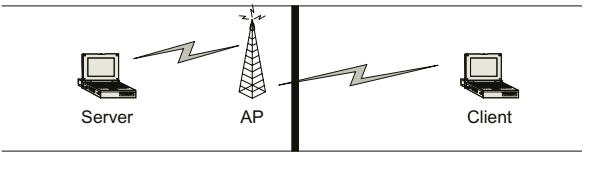


Fig. 1. Configuration of the testbed for collecting error traces

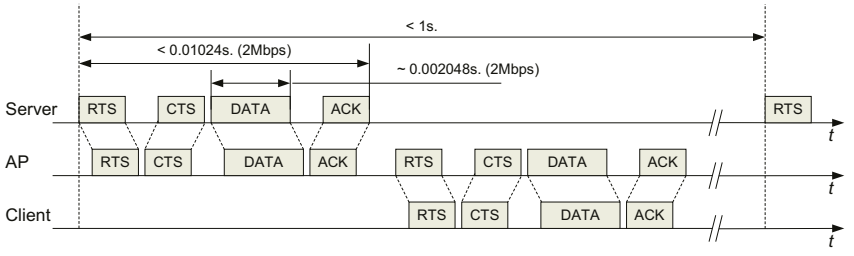


Fig. 2. Exchange of packets between the server and the client via AP

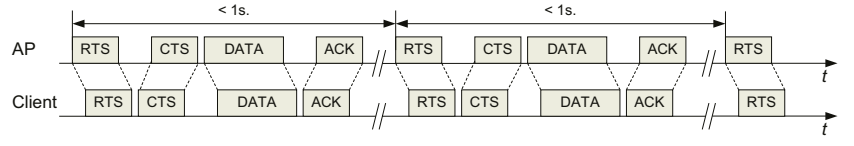


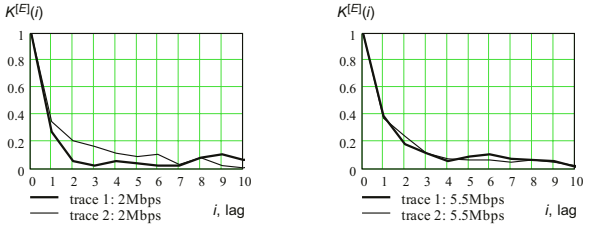
Fig. 3. Exchange of packets between the AP and the client

Experiments were carried out using direct sequence spread spectrum (DSSS) on 2Mbps and 5.5Mbps bit rates [13]. For each rate, three experiments were carried out. In the course of each experiment, error traces were collected on bit, byte, and frame levels. In what follows, we are interested in frame error traces.

It is important to note that no errors were observed between the server and the AP. This is due to LOS environment and absence of interfering nodes. Indeed, time required to transfer 512 bytes packet from the server to AP is less than 0.01s. for 2Mbps bit rate. Therefore, in this testbed frame transmission from the AP to the client can be seen as synchronous in nature, e.g. exchange of RTS-CTS-DATA-ACK between the AP and the client always starts at the same time, separated by a second as illustrated in Fig. 3. Note that the client still provided a feedback in terms of correct and incorrect DATA packet reception in ACK packets.

3.2 Frame Error Traces

Frame error trace is essentially a sequence of successive events of correct and incorrect frame reception at the data-link layer. To use theory of stochastic processes, we redefine the frame error trace to be a sequence of random variables. We assume that '1' represents an incorrectly received frame and '0' represents a correctly received frame. Successive realizations of this random variable compose frame error process $\{W^{[E]}(n), n = 0, 1, \dots\}$, $\{W(n) \in \{0, 1\}\}$. Using the the 'run' test [11] we found that our frame error traces can be considered as covariance stationary ergodic ones. Therefore, we can compute all



(a) ACF for 2Mbps traces (b) ACF for 5.5Mbps traces

Fig. 4. NACF for 5.5Mbps, and 2Mbps rate traces

important statistics using only one realization. We use $f^{[E]}(1)$ to denote probability of frame error as seen by times-averages. Correspondingly, $(1 - f^{[E]}(1))$ denotes probability of correct frame reception. We concentrate our attention on two statistical characteristics of frame error traces. These are PDF and NACF of the frame error process. These characteristics provide sufficient information regarding the stochastic nature of covariance stationary binary processes.

Let us denote by $\{W_x^{[E]}(n), n = 0, 1, \dots\}$, $W_x^{[E]}(n) \in \{0, 1\}$ the empirical frame error traces, where $x \in \{2, 5.5\}$ designates the rate to which the trace corresponds. Histograms of relative frequencies of frame error traces were found to be as follows

$$f_{5.5}^{[E]}(i) = \begin{cases} 0.246, & i = 1 \\ 0.754, & i = 0 \end{cases}, \qquad f_{5.5}^{[E]}(i) = \begin{cases} 0.117, & i = 1 \\ 0.883, & i = 0 \end{cases},$$

$$f_{2}^{[E]}(i) = \begin{cases} 0.096, & i = 1 \\ 0.904, & i = 0 \end{cases}, \qquad f_{2}^{[E]}(i) = \begin{cases} 0.053, & i = 1 \\ 0.947, & i = 0 \end{cases}. \tag{1}$$

It is known that one-dimensional statistical distribution may not completely characterize the frame error process. Indeed, frame errors are intended to group in bursts rather than be distributed evenly in time. Such a grouping may not allow satisfactory data-link error concealment. As a result, it significantly affects quality provided to higher layers. Grouping of errors can be described by NACF. Empirical NACFs of frame error traces for 5.5Mbps, and 2Mbps rate traces are shown in Fig. 4, where i is the autocorrelation interval and $K^{[E]}(i)$ is the value of NACF for lag i.

Observing Fig. 4 one may note that NASFs decrease geometrically fast for small lags (up to $i = 2 \sim 3$). For larger i the decrease is likely not geometrical. However, values of $K^{[E]}(i)$ for $i > 2 \sim 3$ are significantly smaller that those for $i = 1, 2, 3$ allowing to assume that the memory of the process is almost fully determined by first several values of $K^{[E]}(i)$.

4 The Model

To model frame error traces we propose to use doubly-stochastic Markov modulated model with two states of the modulating Markov chain. According to this model, for each pair of states of the modulating Markov chain we may define different probability distribution function. These functions are not limited to analytical ones, but can be general, including arbitrary histogram of relative frequencies. NACF of such a process is geometrically distributed and may produce good approximation of empirical NACF. This process is known as discrete-time batch Markovian arrival process (D-BMAP) in traffic modeling and hidden Markov model (HMM) in signal processing. We refer to such a process as discrete-time batch Markovian process (D-BMP).

4.1 Discrete-Time Batch Markovian Process

Let us briefly review important characteristics of D-BMP. Assume a discrete-time environment, i.e. time axis is slotted, the slot duration is constant and given by $\Delta t = (t_{i+1} - t_i)$, $i = 0, 1, \ldots$. Let $\{W(n), n = 0, 1, \ldots\}$, $W(n) \in \{0, 1, \ldots\}$, be D-BMP. According to it, the value of the process is modulated by a discrete-time Markov process $\{S(n), n = 0, 1, \ldots\}$, $S(n) \in \{1, 2, \ldots, M\}$. Let D be its the transition probability matrix. We define D-BMP process as a sequence of matrices $D(k)$, $k = 0, 1, \ldots$, containing probabilities of transition from state to state accompanied by a value of the process.

Let the vector $\boldsymbol{G} = (G_1, G_2, \ldots, G_M)$ be the mean vector of D-BMP, where $G_i = \sum_{j=1}^{M} \sum_{k=0}^{\infty} k d_{ij}(k)$, $i = 1, 2, \ldots, M$. The mean process of D-BMP is defined as $\{G(n), n = 0, 1, \ldots\}$ with $G(n) = G_i$, while the Markov chain is in the state i at the time slot n. The ACF of the mean process is given by:

$$R^{[G]}(i) = \sum_{l, l \neq 1} \phi_l \lambda_l^i, \qquad i = 1, 2, \ldots, \tag{2}$$

where $\phi_l = \boldsymbol{\pi}(\sum_{k=1}^{\infty} kD(k))\boldsymbol{g}_l \boldsymbol{h}_l (\sum_{k=1}^{\infty} kD(k))\boldsymbol{e}$, λ_i is the ith eigenvalue of D, $\boldsymbol{g}_l$ and $\boldsymbol{h}_l$ are left and right eigenvectors of D, and $\boldsymbol{e}$ is the vector of ones. When D-BMP is used to model binary process (that is $W(n) \in \{0, 1\}$) with only two states of the modulating Markov chain (2) reduces to

$$R^{[G]}(i) = \phi \lambda^i, \tag{3}$$

where ϕ is the variance of the process, λ is the non-unit eigenvalue.[1] NACF is then $K^{[G]}(i) = \lambda^i$. It is clear that the ACF of the mean process of D-BMP exhibits geometrical decay. Such a behavior may produce fair approximation of empirical NACFs exhibiting geometrical decay for small lags.

[1] Transition probability matrix of the irreducible, aperiodic discrete-time Markov chain always posses a unit eigenvalue that is referred to as *simple eigenvalue*.

In general, probabilistic characteristics of $\{W(n), n = 0, 1, \dots\}$ and $\{G(n), n = 0, 1, \dots\}$ are different. For mean process of binary D-BMP the following holds

$$E[G] = E[W], \quad \phi^{[G]} = \phi^{[W]}, \quad K^{[G]}(i) = K^{[W]}(i), i = 1, 2, \dots, \tag{4}$$

where the first property holds for any D-BMP, the latter hold for binary D-BMP.

Without affecting abovementioned autocorrelational properties we allow our D-BMP process to have probability functions that depend on the current state only. In this case, $D(k)$, $k = 0, 1, \dots$, have the same elements on each row. It is important that this process still has ACF distributed according to (2).

4.2 Approximation of the Empirical ACF

To approximate the empirical NACF of the frame error traces we use the method proposed in [1, 14]. Particularly, we minimize the error of approximation γ by varying the value of coefficient λ according to

$$\gamma = \frac{1}{i_0} \sum_{i=1}^{i_0} \left(\frac{K^{[W^{[E]}]}(i) - \lambda^i}{K^{[W^{[E]}]}(i)} \right), \qquad K = 1, 2, \dots \tag{5}$$

where i is the lag, i_0 is the lag up to which the NACF have to be approximated, and $K^{[W^{[E]}]}(i)$ is the value of empirical NACF for lag i. Note that NACF is used instead of ACF in (5). It is easy to see that $\lambda = K^{[E]}(1)$, $\gamma = 0$, for $i_0 = 1$. Values of λ obtained for frame error traces for different i_0 are shown in Table 1.

It is known that the transition probability matrix of irreducible aperiodic two-state Markov chain posses a single non-unit eigenvalue. In what follows, λ is used as this eigenvalue. One should note that more than a single coefficient λ can be used to approximate empirical NASF. However, with increasing of the number of coefficients approximating empirical NACF, the number of eigenvalues increases, and so does the state space of the modulating Markov chain. When K coefficients are used, the number of states of the modulating Markov chain, N, is

Table 1. Approximation of empirical NACFs by λ for different i_0

Trace/i_0	2Mbps		2Mbps		5.5Mbps		5.5Mbps	
	λ	γ	λ	γ	λ	γ	λ	γ
$i_0 = 1$	0.256	0	0.353	0	0.415	0	0.305	0
$i_0 = 2$	0.189	0.038	0.423	0.029	0.466	0.011	0.328	0.004
$i_0 = 3$	0.097	0.762	0.477	0.086	0.524	0.089	0.368	0.112
$i_0 = 4$	0.098	0.819	0.516	0.118	0.569	0.148	0.388	0.269
$i_0 = 5$	0.098	0.856	0.545	0.149	0.603	0.218	0.397	0.395

between[2] $(K+1)$ and 2^K. However, it is always wise to keep the complexity of the model as low as possible. Therefore, the state space of the modulating Markov chain should be minimized. From this point of view, usage of a single geometrical term provides the best trade-off between the accuracy of the approximation and the simplicity of the model.

4.3 Approximation by Mean Process

The construction of Markov modulated process from statistical data involves the inverse eigenvalue problem. It is known that the general solution of this problem does not exist. However, it is possible to solve it when some limitations on eigenvalues and resulting process are set. Our limitation is that the non-unit eigenvalue should be located in $(0, 1]$ fraction of $0X$ axis. Note that $-1 \leq \lambda \leq 1$ is fulfilled since all eigenvalues of transition probability matrix of irreducible aperiodic Markov chain are located in $[-1, 1]$ fraction of $0X$ axis. Finally, $0 < \lambda \leq 1$ must be fulfilled by the solution of the inverse eigenvalue problem.

Let $\{W(n), n = 0, 1, \dots\}$, $W(n) \in \{0, 1\}$, be the D-BMP process modeling a frame error trace, and $\{S(n), n = 0, 1, \dots\}$, $S(n) \in \{1, 2\}$ be its modulating Markov chain. Let $\{G(n), n = 0, 1, \dots\}$, $G(n) \in [0, 1]$ be the mean process of $\{W(n), n = 0, 1, \dots\}$.

Stochastic properties of $\{G(n), n = 0, 1, \dots\}$ are *completely characterized* by a triplet $(E[W], \phi, \lambda)$, where $E[W]$ is the mean of the process, ϕ is the variance, and λ is the non-unit eigenvalue of the modulating Markov chain. Particular values of $(E[W], \phi, \lambda)$ can be related to parameters of $\{G(n), n = 0, 1, \dots\}$ using the following equations

$$\begin{cases} E[W] = \frac{\alpha G_2 + \beta G_1}{\alpha + \beta} \\ \lambda = 1 - \alpha - \beta \\ \phi = \alpha\beta \left(\frac{G_1 - G_2}{\alpha + \beta} \right)^2 \end{cases}, \tag{6}$$

where G_1 and G_2 are means in states 1 and 2 respectively, α and β are probabilities of transition from state 1 to state 2 and from state 2 to state 1 respectively.

Observing (6) one may note that in order to *completely parameterize* the mean process of SD-BMP, we must provide four parameters $(G_1, G_2, \alpha, \beta)$. If we choose G_1 as a free variable with constraint $G_1 < E[W^{[E]}]$ to satisfy $0 < \lambda \leq 1$, we can determine G_2, α, and β from the next set of equations

$$\begin{cases} G_2 = \frac{\phi^{[E]}}{E[W^{[E]}] - G_1} + G_1 \\ \alpha = \frac{(1-\lambda)(E[W] - G_1)}{G_2 - G_1} \\ \beta = \frac{(1-\lambda)(G_2 - E[W])}{G_2 - G_1} \end{cases}, \tag{7}$$

where $\phi^{[E]}$ is the variance of the process, $E[W^{[E]}]$ is the mean of the process, λ is the coefficient determined at the previous step and used as a non-unit eigenvalue.

[2] Particular value of N depends on the solution of the inverse eigenvalue problem.

From the first equation of (7) one may conclude that there should be an infinite number of processes matching $(E[W^{[E]}, \phi^{[E]}, \lambda)$. However, there is an additional restriction on the choice of G_1. Let us now identify a distinctive feature of the proposed matching method that uniquely identifies the process we are looking for and simplifies the fitting procedure. Consider the first equation in (7) and rewrite it using $\phi^{[E]} = E[W^{[E]}] - (E[W^{[E]}])^2$

$$G_2 = \frac{E[W^{[E]}] - E[W^{[E]}]G_1}{E[W^{[E]}] - G_1}. \tag{8}$$

To represent the frame error trace, SD-BMP $\{W(n), n = 0, 1, \dots\}$ must be defined on the state space $\{W(n) \in \{0, 1\}\}$. Thus, the value of G_2 must be equal or less than 1 for any state of $\{S(n), n = 0, 1, \dots\}$. To identify what values of G_1 must be chosen to satisfy $G_i \leq 1$, $i = 1, 2$, consider (8) with extreme cases, $G_1 \to E[W^{[E]}]$ and $G_1 \to 0$. We get

$$\begin{aligned} &\lim_{G_1 \to E[W^{[E]}]} \frac{E[W^{[E]}] - E[W^{[E]}]G_1}{E[W^{[E]}] - G_1} = \infty, \\ &\lim_{G_1 \to 0} \frac{E[W^{[E]}] - E[W^{[E]}]G_1}{E[W^{[E]}] - G_1} = 1, \end{aligned} \tag{9}$$

Observing (9) one may note that $G_1 = 0$, $G_2 = 1$, gives us the only process exactly matching $(E[W^{[E]}, \phi^{[E]}, \lambda)$. Hence, the only parameters we have to determine to match the mean process of empirical frame error trace are α and β

$$\begin{cases} \alpha = (1 - \lambda)E[W] \\ \beta = (1 - \lambda)(1 - E[W]) \end{cases}. \tag{10}$$

4.4 Approximation of the Histogram

To assure that the histogram and NACF are matched we should assign the probability function of frame error to each state of the four state modulating Markov chain such that the whole probability function matches the histogram of relative frequencies of bit errors. For probability functions in every state of the modulating Markov chain the following equations must hold:

$$\begin{cases} \sum_{i=1}^{2} f_j(i)i = G_j, & j = 1, 2, \\ \sum_{i=1}^{2} f_j(i) = 1, & j = 1, 2, \end{cases} \tag{11}$$

where $f_j(i)$, $i = 0, 1$, are probabilities of correct and incorrect frame reception in state j, G_j is the mean in the state j. The last equation in (11) is just normalizing condition that must hold for every discrete probability function.

Observe that the histogram of relative frequencies of frame error trace has only two bins corresponding to correct and incorrect bit receptions. Since $f_j(i)i =$

0, $i = 0$, $j = 1, 2$, and $f_j(i) = f_j(i)$, $i = 1$, $j = 1, 2$, the first equations in (11) reduce to

$$f_j(1) = G_j, \qquad j = 1, 2. \tag{12}$$

Since we satisfied $G_j \leq 1$, $f_j(i)$, $i = 0, 1$, $j = 1, 2$, must satisfy the normalizing conditions. From the second equations in (11) we obtain $f_j(0)$, $j = 1, 2$, as follows

$$f_j(0) = 1 - f_j(1), \qquad j = 1, 2. \tag{13}$$

5 Algorithm and Practical Implications

5.1 Modeling Algorithm

The step-by-step algorithm is as follows:

1. **START**
2. compute $f^{[E]}(i)$, $i = 0, 1$, $K^{[E]}(i)$, $i = 0, 1, \ldots, i_0$, $E[W^{[E]}]$;
3. approximate $K^{[E]}(i)$ using λ according to (5);
4. choose $G_1 = 0$, $G_2 = 1$;
5. compute α and β according to (10);
6. compute $f_j(1)$, $j = 1, 2$, according to (12);
7. compute $f_j(0)$, $j = 1, 2$, according to (13).
8. **END**

We made available the C code of the model as well as pre-compiled binaries for Linux and Windows operating systems at http://www.cs.tut.fi/~moltchan.

5.2 Practical Implementation

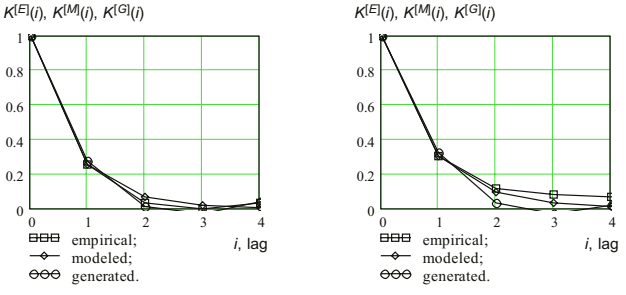
Computational requirements and complexity of the algorithm are low. Thanks to binary nature of frame error traces we have only two histogram bins, $f^{[E]}(0)$ and $f^{[E]}(1)$ corresponding to correct and incorrect frame reception. Therefore, the mean of the empirical trace is expressed using just a single parameter and the same holds for means in states of D-BMP. Indeed, $E[W^{[E]}] = \sum_{i=0}^{1} f^{[E]}(i)i = f^{[E]}(1)1 = f^{[E]}(1)$ and $G_j = \sum_{i=0}^{1} f_j(i)i = f_j(1)1 = f_j(1)$, $j = 1, 2$. Taking into account this property of binary traces we avoid the random search algorithm usually required to approximate the histogram of relative frequencies.

6 Modeling Results

We applied our algorithm to available 2Mbps and 5.5Mbps rate traces and found that it accurately matches both the histogram of relative frequencies and empirical NACF. We show comparison for 2Mbps and 5.5Mbps rate traces.

Table 2. Comparison of statistical characteristics

Trace	2Mbps rate trace			5.5Mbps rate trace		
Parameter	Emp.	Mod.	Gen.	Emp.	Mod.	Gen.
Mean	0.096	0.096	0.093	0.117	0.117	0.112



(a) 2Mbps rate trace (b) 5.5Mbps rate trace

Fig. 5. NACFs of empirical, generated traces and NACF of the model

Comparison of statistical characteristics of empirical data and modeled ones is shown in Table 2. Recall, that the mean value (together with NACF) provides sufficient information regarding distributional properties of the covariance stationary binary stochastic process. One may see that the mean of the model exactly matches mean of the empirical trace. Mean of the generates trace deviates from mean of the model. This is due to statistical fluctuations.

NACFs of empirical and generated traces and NASF of the model are shown in Fig. 5. Good approximation of the empirical data takes place up to lags $2 \sim 3$. Then, the NACF of the model underestimates NACF of empirical trace. We believe that these differences can be neglected since the empirical NACF for $i > 2 \sim 3$ is not far from the confidential interval of NACF estimation.

7 Conclusions

We developed simple and computationally efficient model of IEEE 802.11b frame error traces that is capable to capture important statistical characteristics of frame error traces. The associated algorithm allows to explicitly take into account distributional and autocorrelational properties of empirical data. We showed that among the class of Markov modulated processes with two-states of the modulating Markov chain there is unique model that provides best approximation of covariance-stationary binary stochastic process. Such a model results in extremely simple parameterization algorithm.

We validated our model against empirical frame error traces of IEEE 802.11b wireless access technology. The proposed approach is not only applicable to frame error traces but can be used to model bit error traces of wireless channels as long as the empirical NACF can be approximated by geometrically distributed component. We believe that the proposed algorithm can also be used to model frame and bit error traces of other wireless access technologies. For this purpose we made available the C code of the model as well as pre-compiled binaries for Linux and Windows operating systems at http://www.cs.tut.fi/~moltchan.

References

1. A. Lombardo, G. Morabito, and G. Schembra. An accurate and treatable Markov model of MPEG video traffic. In *Proc. of IEEE INFOCOM*, pages 217–224, 1998.
2. T. Rappaport. *Wireless communications: principles and practice*. Communications engineering and emerging technologies. Prentice Hall, 2nd edition, 2002.
3. B. Sklar. Rayleigh fading channels in mobile digital communication systems part I: characterization. *IEEE Comm. Mag.*, pages 90–100, July 1997.
4. H. Bai and M. Atiquzzaman. Error modeling schemes for fading channels in wireless communications: a survey. *Wireless Networks*, 5(2):2–9, 4th Quarter 2003.
5. M. Zorzi, R. Rao, and L. Milstein. ARQ error control for fading mobile radio channels. *IEEE Trans. on Veh. Tech.*, 46(2):445–455, May 1997.
6. M. Zorzi and R. Rao. The effect of correlated errors on the performance of TCP. *IEEE Comm. Let.*, 1(5):127–129, September 1997.
7. M. Zorzi, A. Chockalingam, and R. Rao. Throughput analysis of TCP on channels with memory. *IEEE JSAC*, 18(7):1289–1300, July 1999.
8. S. Khayam and H. Radha. Markov-based modeling of wireless local area networks. In *ACM MSWiM*, pages 100–107, San-Diego, US, September 2003.
9. G. Nguyen, R. Katz, and B. Noble. A trace-based approach for modeling wireless channel behavior. In *Proc. Winter Simu;lation Conf.*, pages 597–604, 1996.
10. S. Karande, S. Khayam, M. Krappel, and H. Radha. Analysis and modeling of errors at the 802.11b link layer. In *Proc. IEEE ICME*, July 2003.
11. S.A. Khayam. IEEE 802.11b traces. Technical report, Michigan State University, http://www.egr.msu.edu/waves/people/Ali_files/bit_trace_802_11b.zip, Accessed 11.11.2004.
12. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Standard, IEEE Std. 802.11-1997, 1997.
13. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4GHz band. Standard, IEEE Std. 802.11b-1999, 1999.
14. D. Moltchanov, Y. Koucheryavy, and J. Harju. The model of single smoothed MPEG traffic source based on the D-BMAP arrival process with limited state space. In *Proc. of ICACT*, pages 55–60, Phoenix Park, R. Korea, January 2003.

Efficient Multicast Trees with Local Knowledge on Wireless Ad Hoc Networks*

Tansel Kaya[1,2], Philip J. Lin[3], Guevara Noubir[1], and Wei Qian[1]

[1] College of Computer and Information Science,
Northeastern University, Boston, MA, USA
{tansel, noubir, qwlli}@ccs.neu.edu
[2] Flarion Technologies Inc., Bedminster, NJ, USA
t.kaya@flarion.com
[3] Draper Laboratory, Cambridge, MA, USA
plin@draper.com

Abstract. In this paper we address the problem of establishing a cost efficient multicast tree among a group of stationary nodes in a multi-hop wireless network. The flooding of broadcast discovery messages is a major limitation to the scalability of most ad hoc protocols. To avoid massive flooding, we limit the reach of broadcast discovery messages, and consider the case were joining nodes can only learn limited information about the multicast group topology from neighbors within a fixed number of hops. We propose two algorithms that satisfy this constraint. We analyze the worst case cost of the established trees and prove that the first algorithm builds a minimal cost spanning tree, while the second builds a sub-optimal tree with a worst-case approximation ratio of O(log n/loglog n). The advantage of the second algorithm is that the communication requirement for a node to join the multicast tree is smaller. We simulate and compare the proposed algorithms. Finally, we discuss the implementation issues and scenarios for using each one of them. We also describe our secure multicast application that builds on top of the proposed protocols.

1 Introduction

Multicast is an important communication paradigm since it allows efficient data delivery from a source to multiple receivers. Multiple applications can benefit from the efficient construction of a low cost multicast tree that spans all group members. This is especially true for wireless multi-hop networks where radio frequency bandwidth is a scarce resource. The construction of an efficient multicast tree can also benefit applications in sensor networks where a group of disseminated nodes have to gather, merge, and deliver sensed data to some central node. In this case, the data is traveling from the leaves to the root. A significant amount of research has already been done on multicast tree construction addressing both wired and wireless networks and from

* Work supported by Draper Laboratory IR&D grant under contract #523120. This work was done while Tansel Kaya was a graduate student at CCIS, Northeastern University.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 246–257, 2005.

theoretical and practical perspectives. In this paper, we focus on the issue of building a low cost multicast tree where the joining node is only allowed to discover limited information about the current multicast tree. This information is obtained from nodes that are within some limited number of hops and that are already members of the considered group. The simplest algorithm in this category would consist of having the joining node broadcast a hop-limited request to discover its closest neighbor already in the group and then attach to it. This algorithm is known in the literature of the theory community as the vertex greedy algorithm [1].

We consider a set of nodes that create a connected multi-hop wireless network. We assume that the nodes are static (e.g., sensor network) and do not make use of power control to adapt their range. The resulting connectivity graph is usually referred to in the literature as Unit Disc Graph (UDG). This assumption makes sense for low-cost sensor networks with only on/off power amplifiers. We also do not make use of the broadcast advantage in this first problem (a node in the tree sends two packets to its children even if they are both within range). The last assumption makes more sense when the group members form a small set of sparsely distributed nodes. The reason for this assumption derives from the application that led to the study of this problem, which is secure multicasting over ad hoc networks [4]. In this application authorized nodes create a multicast overlay network where each node establishes a secure channel with its children and therefore does not benefit from the wireless broadcast advantage. Our goal is to construct a low cost multicast tree that spans all the group members. The cost of an edge between two group members is the number of hops of the shortest path. We consider a special case of the re-arrangeable online Steiner tree problem [3]. This problem consists of nodes joining the group in sequence, where we are allowed to partially rearrange the previously constructed tree. Specifically, we would like to reduce the joining node's communication cost to discover its neighbors and attach to the network. This is a very important constraint because it allows reducing the broadcast messages to a small number of hops around the joining node.

Finding the minimum cost Steiner tree that connects all group members is well known to be a NP-complete problem [2]. We only consider the graph induced by the group members and aim at efficiently constructing a tree that has low cost and spans all the group members. The difference between a Steiner tree and a spanning tree is that in the Steiner tree some nodes that are not group members can be used in the tree as fork nodes (have more then one children). The spanning tree considers only the induced graph where the vertices are the group members. It is also known that the cost of the minimum spanning tree (MST) is at most twice the cost of the minimum Steiner tree. Therefore, we aim at building a low-cost spanning tree and try to do so while reducing the information that needs to be known by the joining nodes. This limits the cost of broadcast. Computing an MST, when the whole network topology is known, is computationally easy (e.g., using Prim-Dijkstra algorithm). However, it is more difficult when only limited information about the network topology is known. Furthermore, we consider the scenario of online algorithms. In the online version of the Steiner tree problem, the nodes appear one at a time, and at the end of step i where node v_i was introduced, the online algorithm must construct a tree T_i that contains nodes $v_1 \ldots v_i$. The new node v_i can be connected to any point of the connected tree.

The Steiner tree problem has been further classified by considering removals and the possibility of rearrangement. If at each step the online algorithm is confined to adjusting the links of the introduced or removed node, then this type of problem is referred to as non-rearrangeable.

From a theoretical perspective, the Steiner tree problem has been extensively studied in arbitrary metric spaces. Imase and Waxman have analyzed a simple greedy online algorithm called the vertex greedy algorithm (VGA) and have shown that it has a competitive ratio of $O(\log n)$ in any metric space for the online Steiner tree problem [3]. Alon and Azar proved, for Euclidean spaces, that any deterministic or randomized online algorithm has a lower competitive ratio bound of $\Omega(\log n/\log \log n)$ [1]. In this paper, we analyze the performance of our algorithms by the competitive ratio measure introduced by Sleator and Tarjan [5]. The competitive ratio is defined as the worst case, over all possible sets V_T (group members), of the ratio between the total cost of the tree constructed by the online algorithm and the minimal Steiner tree for the set V_T. Previous research assumes a complete knowledge of the network topology while we consider limited knowledge, but allow some limited re-arrangement.

Contributions: We propose two algorithms to build and rearrange the multicast tree when a new node joins. The first algorithm builds an optimal tree, but requires the group members already in the tree to know the length of the longest edge in the existing tree. The second algorithm builds a sub-optimal tree, but only requires each group member to know the length of the longest edge on its path to the root of the tree. We show that this second algorithm can have a worst-case approximation ratio of $O(\log n/\log \log n)$. However, our simulation results show that this second algorithm is usually within 25% of the optimum, and performs much better when the density of group members is high. From this theoretical analysis we derived a multicast routing protocols integrated with security mechanisms for access control, compromised nodes revocation capability, and data integrity provision over ad hoc networks. We have implemented the integrated protocols in our testbed [4].

2 Globally Longest Logical Edge Algorithm (GLLE)

The greedy algorithm described in [1] is a simple solution to the problem of attaching nodes to a multicast tree. For each join request, the algorithm attaches the new node using the shortest path to an existing node of the tree without making any rearrangements of existing links. For metric spaces consisting of only the member nodes and the shortest distances between them, a variant called the vertex greedy algorithm (VGA) is used. In this variant, the new node can only be connected to a member node. Connections to intermediate nodes of the tree are not allowed. Although the vertex greedy algorithm is simple and robust, it is unsuitable for long-term settings where the construction of an optimal spanning tree is desired. The cost of the tree constructed by the VGA can have a $O(\log n)$ ratio to that of the optimal tree.

We present an optimal algorithm for constructing a MST, which requires all member nodes to be updated about the longest edge in the tree before an insertion is performed. We define the globally longest logical edge (*GLLE*) as the largest num-

ber of hops between any two member nodes of the multicast tree. A joining node would discover all group members within *GLLE* hops distance. It forms cycles by establishing edges with these nodes and topologically sorts all fork nodes, breaking the longest edge in each cycle following this topological order. The algorithm finally commits to the selected link. First we present a sample scenario. Figure 1.a shows the corresponding graph for a sample ad-hoc wireless network, where there is an edge between two nodes if they are mutually in range of each other. Members of the multicast tree are numbered and are shown in gray. Figure 1.b shows the induced graph, where edges are computed using the shortest paths between member nodes and the joining node.

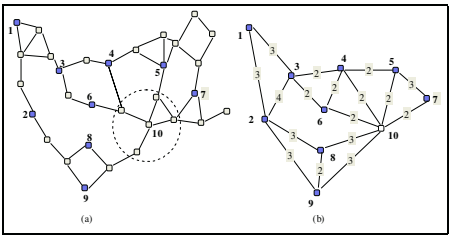


Fig. 1. (a) A sample network with member nodes in gray (b) The corresponding induced graph of member nodes

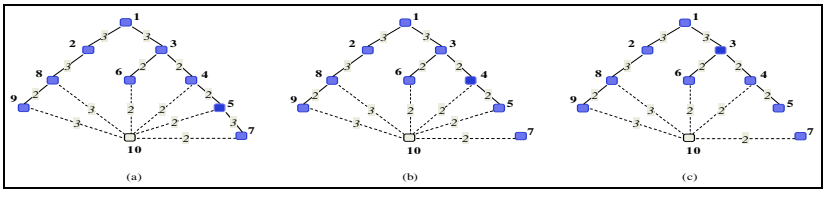


Fig. 2. Node insertion of the GLLE algorithm, steps a-c

The operation of the GLLE algorithm for node 10 is shown in Figures 2 and 3. Part (a) represents the tree generated by the algorithm through nodes 1 to 9, where node 1 is the group source and the globally longest edge is 3. The algorithm starts with node 10 discovering members of the tree within 3 hops (*GLLE* = 3) distance and establishing virtual edges, shown in dotted lines. Next the algorithm topologically sorts member nodes according to the number of incident paths, starting with node 10 and traversing them such that a node is always traversed later than its children.

In steps (a)-(f), the nodes 5, 4, 3, 8 and 1 are traversed and cycles are broken by deleting the longest edge in the cycle. In case of ties, virtual edges are removed first. In the last step, the logical edges between node 10 and nodes 6 and 7 are committed.

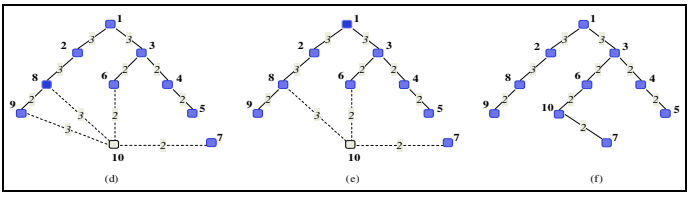


Fig. 3. Node insertion of the GLLE algorithm, steps d-f

Here we give a detailed description of the algorithm. Let V_G denote the set of member nodes and L denote the globally longest logical edge of the tree. $d(v_i, v_j)$ is defined as the distance in hops between any two nodes. Let v_n be the joining node at step n and S_L the set of nodes in V_G where distance to v_n in hops is less than or equal to L.

```
GLLE(V, E, V_G, v_n)
   if  v_n = groupSource then
      L←0; V_G←{v_n};
   else
   /* Determine members up to L hops away */
   if  |S_L| • 1 then
      Attach to v_min the closest member within V_G
      L ← max(L, d(v_min, v_n)) ;
      V_G← V_G ∪{v_n}
   else
      G← (V' = ancestors(S_L), E': v_j = ancestor(v_i) ⇒ (v_i, v_j)
      ∈ E')
      V_T ← TopologicalSort(G)
      for (v_t = first(V_T) to last(V_T)) do
         /* Execute algorithm of Lemma 1 to update E*/
         while (|Paths from v_t to v_n| ≥ 2) do
            Let P the path from v_t to v_n with lowest longest
            edge
            for each path P' from v_t to v_n and P' • P
               Remove longest edge of P' from E
               Reverse direction of edges below removed edge
      Attach to v_min = P ∩ S_L on the last remaining path
      L←max(E); V_G← V_G ∪{v_n};
```

3 Proof of Correctness and Optimality

We claim that the above algorithm produces a minimum spanning tree over the induced graph. We prove the optimality of the above algorithm using two intermediate steps. First we show that our cycle elimination step results in the optimal cost for disjoint paths. Then we argue that topological sorting always results in the optimal tree in the presence of non-disjoint paths and last we prove that the set of discovered nodes is both necessary and sufficient.

Theorem 1: Given a set of nodes V from a connected metric space, if a terminal set $V_T \subset V$ is presented to the GLLE algorithm one element at a time, it constructs a minimum spanning tree connecting V_T.

Since the minimum spanning tree over the graph induced by the group members has a cost at most twice the cost of the minimum Steiner tree, then the GLLE algorithm constructs a tree with a cost at most twice the minimum Steiner tree over V_T.

Lemma 1: Given a Directed Acyclic Graph (DAG) $D = (V_D, E_D)$ with a single source s, a single sink t and having all paths from s to t disjoint, a MST $T=(V_T, E_T)$ rooted at the sink is obtained by preserving the path P_m with the weight of the heaviest edge $w_{max}(P_m)$ smallest among all paths, removing the respective heaviest edges from other paths P_k and reversing all edges on these paths up to the removed heaviest edge.

Proof: A DAG D with disjoint paths from s to t is converted into a spanning tree *iff* one disjoint path is fully included in the spanning tree to connect the source and one and only one edge is removed from each disjoint path to remove cycles, while keeping all intermediate nodes connected to T (See Figure 4.). The cost of a spanning tree can be computed as the sum of all edges of D less the cost of the removed edges. This value is minimized by removing heaviest edges from paths that are not fully included.

Assume that the tree T containing path P_m is not optimal and there exists another tree T' containing path P_l, which has a lower total cost. This leads to a contradiction.

$$T' = |D| - \sum_{k=1,k\neq l}^{n} \max(\, w(s, v_{k,1}), \ldots, w(v_{k,n}, t))$$

$$\wedge\, T = |D| - \sum_{k=1,k\neq m}^{n} \max(\, w(s, v_{k,1}), \ldots, w(v_{k,n}, t)\,)$$

By definition $w_{max}(P_m)$ is smallest among all paths, so the cost of T is smaller than T'.

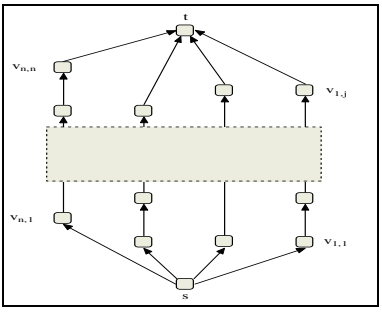


Fig. 4. Single Source-Sink DAG with Disjoint Paths

$$\max(w(s, v_{m,1}), \ldots, w(v_{m,n}, t)) \leq \max(w(s, v_{l,1}), \ldots, w(v_{l,n}, t)) \Rightarrow T \leq T' \;\square$$

After proving that the cycle elimination step is correct for disjoint paths, we have to show that it also holds for a set of non-disjoint paths.

Lemma 2: Let $P_{inc}(v_x)$ be the number of incident paths through a vertex v_x. Given a DAG $D = (V_D, E_D)$ with a single source-sink pair and intermediate nodes with out-

degree one, another MST, T' is obtained by topologically sorting[1] all nodes v_x by the number of incident paths $P_{inc}(v_x)$ and applying the algorithm described in Lemma 1 to the sub-graphs $D' = (V_{D'}, E_{D'})$ in topological order of $P_{inc}(v_x)$, where the sub-graphs have s as a source and v_x as the sink with $P_{inc}(v_x) \geq 2$.

Proof: We must show that the algorithm described in Lemma 2 correctly partitions the graph into sub-graphs satisfying the preconditions described in Lemma 1 and this sequence of partitions produces a MST. The proof proceeds by induction on sub-graphs. Any pair of paths in the given DAG share common edges *iff* these paths intersect at a vertex. The common edges of paths occur in order and start from an intersection node. For each of these intersection nodes there could also be disjoint paths of the form $P_i=\{s, \ldots, v_x\}$. So for a DAG with n+1 non-disjoint paths, there exist n+1 nodes, where pairs of non-disjoint paths intersect.

In the base case, we consider the DAG formed by a node v_n and the source. The resulting T' is trivially a MST. In the inductive step, we assume the optimality and correctness of the case with n non-disjoint paths and analyze the case of n+1 non-disjoint paths with arbitrary number of disjoint paths for both cases. Since any sub-tree of a MST T is another MST T' composed of a proper subset of nodes, the problem shows the optimal substructure property.

For n+1 non-disjoint paths, there exists a node v_n at which the n^{th} and $n+1^{th}$ paths intersect. A sub-graph D^i is rooted at the i^{th} member of the topological list v_i and consists of disjoint paths $P_1 \ldots P_n$ s.t. $P_1 = \{s, v_i, \ldots, v_x\}$, …, $P_n = \{s, v_j, \ldots, v_x\}$. By definition of the topological sorting function, the node v_n is traversed last with D^n as a subtree of v_n. Since D^n is a tree, there can be a single path from s to v_n through D^n. Hence D^n can be considered as a disjoint path and evaluated along with all other disjoint paths leading from s to v_n. By Lemma 1, the described algorithm produces a MST rooted at v_n. □

In the last part of the proof, we prove that the DAG constructed from the set of discovered nodes and their ancestors is required and sufficient for optimality. Here $E_{T,n}$ is defined as the set of all edges from a non-member vertex v_n to vertices in tree, V_T with a weight less than or equal to the longest edge in E_T (called GLLE).

Lemma 3: Given a non-member vertex v_n and a MST $T = (V_D, E_D)$ rooted at t, another MST T' is obtained after constructing the DAG $D = (V_D + v_n , E_D + E_{T,n})$ and applying the algorithm described in Lemma 2. If there is no node satisfying this requirement, $E_{T,n}$ consists of the minimal weight edge between v_n and a vertex $v_m \in V_T$.

Proof: We have to show that the graph $G = (V_D + v_n, E_D + E_{T,n})$ satisfies the requirements described in Lemma 2. Further we must prove that for the vertex set $V_D + v_n$ the augmented set of edges $E_D + E_{T,n}$ contains a MST.

The graph G can be converted into a DAG by selecting v_n as a source and the root of T v_r as the sink and converting all undirected edges into directed edges such that for all intermediate nodes the out-degree is one.

---

[1] The choice of $P_{inc}(v_x)$ as the total ordering function is not mandatory. Only a partial ordering is necessary: $v_x > v_y$ if there exits a path from v_y to v_x.

Suppose there exists a node v_k, within the path P_k, whose distance to v_n is larger than the maximum edge weight in the tree. It is clear that there is no edge in P_k with a weight w larger than the weight of the edge (v_k, v_n) or that edge would have been the maximum. Including this edge (v_k, v_n) does not reduce the total cost of T', since one cannot remove a higher weight edge and it does not extend the tree to a new node. This implies $(v_k, v_n) \notin T'$. The edge (v_k, v_n) can be used to construct a spanning tree on $V_D + v_n$, only in the case when there is no node of T, whose distance to v_n is smaller than or equal to the maximum edge weight.

Similarly, assume there exists an edge between a pair of disjoint paths which has smaller weight than all edges in either of the paths. T could have a lower cost if it included this edge. This implies that T is not an MST, which is a contradiction. □

4 Locally Longest Logical Edge Algorithm (LLLE)

The GLLE algorithm we presented requires each node to be updated about the longest edge of the entire tree, which is costly because the tree changes at each join. A more practical assumption is that each node knows the longest edge on its path to the source, which we refer to as a locally longest logical edge (LLLE). In the LLLE algorithm, we determine the closest member node and obtain its LLLE information. We keep increasing our range if this value is larger than our current range, and stop when the LLLE value cannot be increased based on the information obtained from the contacted nodes.

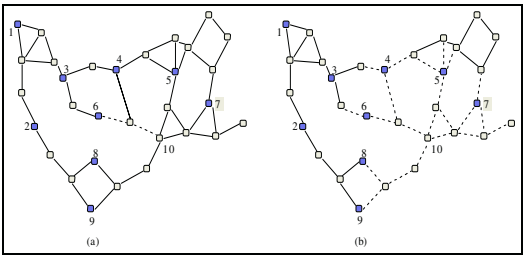


Fig. 5. Node Discovery and Insertion with LLLE

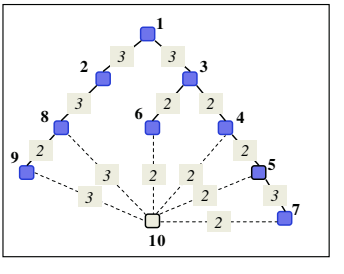


Fig. 6. The insertion step for

Figures 5 and 6 show the tree constructed over the nodes through 1 to 9 using the LLLE algorithm and the insertion of node 10. Node 10 first contacts the closest node (node 6) and obtains its LLLE value, which is 3. It extends its broadcast hops to 3, obtains new LLLE values and stops since all discovered nodes provide an LLLE lower or equal to 3. The choice of the first closest neighbor is not important since all LLLE values from neighbors within the final value will eventually be gathered. We can see in Figure 6 that for this instance of the problem the algorithm produces the same tree if the nodes are inserted in the same order. Except for the update of the global longest logical edge information, the cycle elimination and link commitment steps are the same as the GLLE algorithm. They are described in Section 2 and can be followed from Figures 2 and 3.

5 Competitive Ratio

The LLLE algorithm alleviates the need to store and update the GLLE information, but it results in sub-optimal multicast trees. Figure 7 presents an adversarial scenario where the second algorithm yields the worst case competitive ratio $O(\log n/\log\log n)$. Suppose we are given a network of nodes arranged in a (i+1)*(j+1) matrix, where $i = k^m$, $j = \sum_{l=0}^{m-1} k^l + 1$ and k, m are positive integers. Using the second algorithm, we first introduce all the nodes in the first (top) row one at a time into the multicast tree from left to right. This results in a chain of member nodes, on which any two successive member nodes are one hop away. The remaining member nodes are chosen from m carefully selected columns in order to constitute the worst-case scenario.

The last node in column 1 is inserted first, and it joins to the same column's first node with a distance of i hops. The next column is chosen as $i/k+1$, where we pick k member nodes in such a manner that any two close member nodes in the column are i/k hops away, and the node from the smallest row joins first. The third column is $i/k+1 + i/k^2+1$ with k^2 equally spaced new member nodes, and so on. (See Figure 7 for an example with $k = 2$ and $m = 4$.) The total cost for constructing such a multicast tree is given as $(m+1)\times k^m + \frac{k^m - 1}{k - 1} + m$. We also compute the total cost for the corresponding optimized multicast tree, $k^m + m\times k^{m-1} + 2\times\frac{k^m - 1}{k - 1} + m$. (See Figure 7.b for the optimized multicast tree for the scenario in Figure 7.)

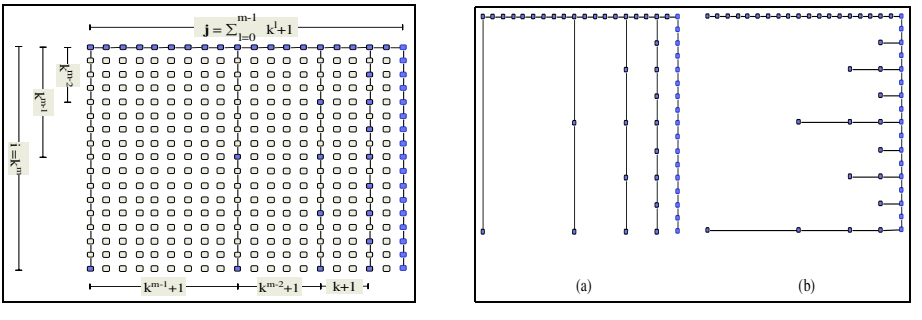


Fig. 7. Scenario with k=2, m=4. Constructed tree has a cost ratio $O(\log n)$ versus the OPT

If we take $m = k$ such that the total number of nodes $n = O(k^{2m})$, the competitive ratio becomes $O(m)$ which can be lower bounded by $O(\log n/\log\log n)$.

6 Implementation

In this section, we briefly describe how both of the algorithms can be implemented on an ad hoc wireless network using message exchanges. For simplicity, we assume that

a source node is present and starts first. For a node other than the source, the algorithms start by discovering the closest member node through a series of broadcasts and obtain the respective longest logical edge value. If the obtained value is not larger than the hop distance to the reached node, then the node immediately joins. Otherwise, it extends its range and collects replies from all reachable nodes. For the LLLE algorithm, this step continues as long as the range can be extended. If multiple nodes reply to its request, the cycle elimination step is used. All replying nodes send their path to the source, along with hop distances.

The joining node runs the algorithm based on the received information and determines the affected nodes. Affected nodes are sent a message to reverse selected edges. After this step is completed, the longest edge values are updated by an update message sent to the root which is propagated to all nodes. This information can be piggybacked with multicast data. In order to prevent any race conditions resulting from concurrent joins, a locking mechanism can be used, where the permission to run the cycle elimination step is obtained from the last node in the topological ordering.

7 Comparison of Algorithms

We simulated the above defined algorithms within a 1000 by 1000 simulation area using uniformly distributed wireless nodes with range 200. We determined the number of total nodes and the number of member nodes using two linear density functions over the simulation area and obtained our sample space by repeating each simulation 120 times. In the following figures, the calculated numbers of total nodes for the given area and the member node densities over the total number of nodes have been given. We collected data on cost ratio, degree, number of contacted nodes and obtained minimum, maximum and mean values. Our additional experiments with larger simulation areas, smaller wireless ranges, and higher number of total and member nodes, were consistent with the results obtained here.

The ratio of optimality has been generated by computing the minimal spanning tree over the induced graph of member nodes. We observe that the GLLE algorithm performs optimally as expected and the LLLE algorithm converges to optimality very quickly as the member node density increases. For higher numbers of total nodes, the rate of convergence also increases. VGA always performs worse than both algorithms, but we do not observe a ratio of optimality above 1.5 in any experiment.

The average number of contacted nodes shows how many nodes on average have been discovered at each step of the construction of the multicast tree. This value is always 1 for the VGA. For the GLLE and LLLE algorithms, it can be given as a linear function of total number of nodes, whose slope includes the member node density. As expected, the LLLE algorithm contacts a fewer number of nodes.

When we analyze each algorithm, we see that the drawbacks of the VGA include non-optimality and high average degree. However, since the average optimality ratio is below 1.5, and because of its simplicity VGA is a good candidate for mobile applications. The GLLE algorithm is more complex to implement due to the requirement that all nodes be updated about the LLE information before any new node can be added. On the other hand, the GLLE algorithm is optimal. It also has smaller expected degree than the other two algorithms.

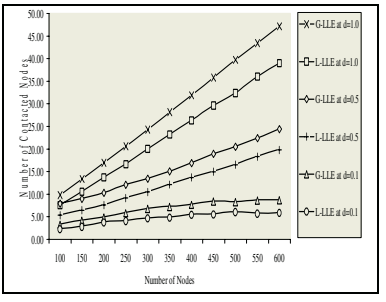


Fig. 8. Comparison of Average Multicast Tree Cost with different densities

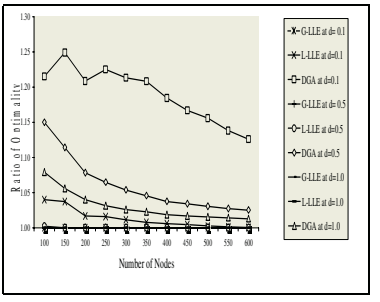


Fig. 9. Comparison of average number of contacted nodes for different densities

Compared with the two other algorithms, the LLLE algorithm provides near optimal cost without the need for globally updating the LLE information. It requires contacting less nodes than the GLLE algorithm. It has a reduced communication cost compared with the GLLE algorithm and both maximum and average tree costs converge to optimality very quickly. While the LLLE algorithm is reduced to the VGA in the worst case, the average behavior is very close to GLLE.

8 Application

Our research on efficient multicast for ad hoc networks was initially driven by a secure location tracking and monitoring of mobile nodes interconnected by a MANET [2]. Because of its simplicity and robustness and considering the simulation results of the greedy vertex algorithm, it was chosen to be implemented in our demonstration application to create multicast tree between mobile nodes. The LLLE algorithm is being implemented between static sensing nodes to build more efficient long-term multicast trees. The prototyping testbed is composed of a set of PDAs and laptops, equipped with a wireless interface (IEEE802.11) and location acquisition interface (CF-GPS) (See **Fig. 10**). The nodes are running *linux*.

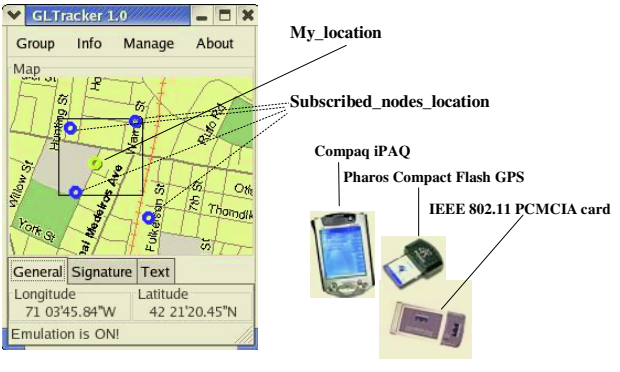


Fig. 10. GUI of the application and components of the MANET testbed nodes

References

1. N. Alon and Y. Azar, “On-line Steiner trees in the Euclidean plane”, Proceedings of the eighth annual symposium on Computational geometry, 1992.
2. M. R. Garey and D. S. Johnson. “Computers and Intractability”, Freeman, 1979.
3. M. Imase and B. Waxman, “Dynamic Steiner Tree Problem”, SIAM J. Disc. Math. 1991.
4. T. Kaya, G. Lin, G. Noubir, A. Yilmaz, “Secure Multicast Groups on Ad-Hoc Networks”, ACM workshop on Security of Ad Hoc and Sensor Networks (SASN ‘03).
5. D. Sleator and R. Tarjan, “Amortized efficiency of list update and paging rules”, CACM 1985.

Limiting Control Overheads Based on Link Stability for Improved Performance in Mobile Ad Hoc Networks

Hwee Xian Tan[1,2] and Winston K.G. Seah[2,1]

[1] Department of Computer Science, School of Computing,
National University of Singapore,
3 Science Drive 2, Singapore 117543
[2] Institute for Infocomm Research,
Agency for Science Technology and Research,
21 Heng Mui Keng Terrace, Singapore 119613
{stuhxt, winston}@i2r.a-star.edu.sg

Abstract. The widespread use of Mobile Ad Hoc Networks (MANETs) in many fields of applications has led to the continuous development of routing protocols which can perform well when deployed under different scenarios, as well as offer better Quality of Service (QoS) support. In this paper, we focus on how link stability is utilized as a metric to provide accurate feedback on the network. We then introduce two mechanisms – L-REQ (Limited forwarding of Route Requests) and L-REP (Limited initiation of Route Replies by intermediate nodes) – which make use of link stability to dynamically adapt the characteristic behaviour of existing protocols to achieve better network performance. The two adaptive schemes are then applied to the Ad Hoc On Demand Distance Vector (AODV) routing protocol as proof of concept.

1 Introduction

MANETs are increasingly gaining pervasiveness in many fields, particularly in the military and rescue operations. This is attributed to the nature of MANETs – they do not require the setup of any existing infrastructure and can be easily deployed under physically hostile terrains.

Several routing protocols have been developed for use in MANETs, including the following: i) reactive protocols such as DSR (Dynamic Source Routing) and AODV (Ad Hoc On Demand Distance Vector) which compute routes on demand; ii) proactive protocols such as OLSR (Optimized Link State Routing) and TBRPF (Topology Dissemination Based On Reverse-Path Forwarding) which store pre-computed paths to reachable destinations; and iii) hybrid protocols such as ZRP (Zone Routing Protocol) and CBRP (Cluster Based Routing Protocol), which make use of the benefits of both reactive and proactive routing schemes. Each of these routing protocols work well under the scenarios which they were designed for: proactive protocols perform well in relatively static networks while their reactive counterparts outperform the proactive ones under highly dynamic network conditions.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 258–268, 2005.

In recent times, there has also been a growing trend towards the development of adaptive protocols, which include the ARM (Adapting to Route Demand and Mobility) protocol, ASAP (Adaptive Reservation and Pre-Allocation Protocol), SHARP (Sharp Hybrid Adaptive Routing Protocol) and ADV (Adaptive Distance Vector) protocol. These adaptive protocols recognize that, owing to the indeterministic nature of network conditions in MANETs, conventional routing protocols may not work well under different network scenarios. Hence, they adjust network parameters dynamically to achieve better network performance, often by making use of one or more mobility metrics that can provide information on the network [1].

As such, it is of utmost importance to identify metrics that can accurately reflect the communication potential in a MANET [2]. In this paper, we have identified link stability as a metric that can be used to provide feedback on the network characteristic. Link stability itself is essential for the formation of stable networks, which enjoy high reliability and better QoS support. With network stability, data packets can be delivered more successfully, resulting in improved network performance.

In networks with unstable links, link breakages tend to occur frequently. Due to the breakage of existing routes, more control packets have to be propagated in the network for route maintenance. In the case of reactive protocols, these include Route Request (RREQ), Route Reply (RREP) and Route Error (RERR) packets, since new routes have to be established more frequently, and nodes that make use of broken linkages will also have to be notified. Consequently, there is greater contention for bandwidth, leading to overall deterioration in network performance.

Therefore, adaptive routing protocols should ensure that stable links are established in preference over unstable links during route formation. Here, our work mainly focuses on how link stability can be determined, as well as demonstrate how it can be used as a metric to dynamically adapt any arbitrary reactive routing protocol that makes use of periodic beacons, to achieve better network performance. We then implement our proposed schemes on AODV, a well-known reactive protocol, and use simulations to compare some performance measures.

The remainder of this paper is organized as follows: Section 2 discusses related work and motivation. Section 3 describes how link stability is determined. Section 4 provides a detailed overview of how link stability is being incorporated as a metric in our adaptive scheme. Simulation results and analysis are presented in Section 5. In Section 6, we conclude with directions for future work.

2 Related Work and Motivation

Much of the reported work in the literature have used estimated distance or estimated route lifetime to measure link stability. This requires the use of GPS (Global Positioning System) to gain knowledge about the speed and/or location of the nodes, which can then be used to determine the stability of the link. Other previously studied methods of estimating link stability include using signal strength and Signal to Noise Ratio (SNR). Some of these schemes are briefly discussed below.

The Signal Strength based Adaptive Routing (SSA) [3] protocol performs route discovery on demand by selecting longer-lived routes based on signal strength and location stability. The average signal strength at which packets are exchanged between hosts is used to determine the strength of the channel and the location stability is used to choose the channel which has a longer time existence.

Krco and Dupcinov in [4] discuss the problems caused by the underlying MAC (Medium Access Control) protocol, which causes different transmission ranges to be set up for unicast and broadcast packets. The SNR value is hence used to determine if the quality of the interconnecting channel is good enough to carry broadcast and unicast messages without dependency on the underlying MAC protocol.

The Minimum Displacement Update Routing (MDUR) [5] is an updating strategy that controls the rate at which route updates are being sent into the network, based on the frequency of location change of a node by a pre-specified distance (using GPS). This reduces periodic route updates by restricting the update transmission to nodes which have not been updated for a minimum threshold time, or nodes which experience/create significant topology change.

Adaptive Location Aided Routing from Mines (ALARM) [6] describes a hybrid routing protocol that combines the LAR (Location Aided Routing) protocol with a directed flooding method. When the link duration for the node in the route is longer than a specified threshold, the data packets are then forwarded along the route; otherwise, the node will initiate a directed flood of the data packet towards the destination.

[7] introduces a new metric for determining the link stability of routes, so as to select longer-lived routes during route discovery. The Route Fragility Coefficient (RFC) estimates the rate at which a given route expands or contracts. With the selection of less dynamic routes, the route lifetime is often longer, resulting in improved throughput and reduced routing protocol overhead.

MOBIC [8] is a lowest relative mobility clustering algorithm that uses a mobility metric as a basis for cluster formation. The relative mobility metric is computed by measuring the received signal strength detected at the receiving node. The aggregate local mobility value is then calculated and used to choose the preferred cluster heads.

While we acknowledge that a considerable amount of research has been done on link stability, there are some issues of concern which we hope to address in this paper. These issues include the following:

- Direct measure of distance may not be a good basis for gauging the stability of the link because it is subject to approximations. Two nodes in close proximity may not be able to transmit data packets efficiently due to low power levels or interferences from the noise in the environment – these two factors lead to low transmission ranges, which could result in higher packet loss rate.
- Different MAC protocols have varying methods of handling unicast and broadcast packets. Some MAC protocols transmit broadcast packets (such as Hello messages) at lower bit rates than unicast data packets, by making use of a more robust modulation scheme which results in higher transmission ranges.
- Nodes in an ad hoc network usually have limited power and processing capabilities. As such, algorithms to compute the link stability should not be too complex, and any additional information that needs to be maintained should be minimized so as not to incur higher network overhead.

Most of the related work presented above used link stability to control flooding of data packets in highly mobile networks, or as a metric for the clustering of nodes in hierarchical routing strategies. However, the main problem faced by routing protocols in very dynamic conditions is that links may be broken soon after route establishment. This leads to excessive control messages and data packets inside the network, which results in higher bandwidth contention and consequently, reduced performance. We hence propose schemes to reduce network overhead using link stability as a metric.

3 Determining Link Stability

Link stability is a measure of the reliability of the link between any two nodes in a network. It depends on a number of factors, such as: distance between the nodes, signal strength emitted by the transmitting node, sensitivity of the receiving node, antenna gains of both the transmitter and the receiver, environmental conditions, etc.

We use the successive transmission power of packets received by nodes to determine link stability. This is done by measuring the signal strength of packets received by nodes in consecutive time periods, and then calculating the relative signal strength. Here, we have chosen to use the relative signal strength and not the direct measure of signal strength because fading in wireless environments can lead to fluctuations in the received signal strengths. The Ground Reflection (Two-Ray) model is used in our simulations, which considers both the direct path and ground reflected propagation path between the transmitter and the receiver.

$$P_r = P_t \times \frac{h_t^2 \times h_r^2}{d^4} \times G_t \times G_r \tag{1}$$

where P_r = received power; P_t = transmitted power; G_t = antenna gain at the transmitter; G_r = antenna gain at the receiver; h_t = height of the transmitter antenna; and h_r = height of the receiver antenna.

3.1 Preliminaries

We define the relative signal strength between any two arbitrary nodes as:

$$Rel_{ss}[x], = 10 \lg(\frac{P_x}{P_x'}) \tag{2}$$

where P_x = signal strength received in current time period; and P_x' = signal strength received in previous time period.

When two nodes are moving away, or when there is increased interference between them, the signal strength that is being sensed will decrease. This leads to a negative value for $Rel_{ss}[x]$, where x refers to a particular neighbour of the node calculating the relative signal strength. The relative signal strength is a good indicator of link stability, because a higher value of $Rel_{ss}[x]$ indicates that the signal strength being received by two adjacent nodes is increasing. Although this can be due to nodes moving closer to each other, or decreased noise/interferences surrounding

these nodes, this is nevertheless an indication that the link between the nodes is stable. Hence, links that are established between these nodes tend to last longer, leading to less frequent breakages.

3.2 Implementation

In typical MANET routing protocols, periodic beacons are broadcasted between neighbouring nodes to provide local connectivity information. In AODV, these periodic beacons are known as Hello messages and are used by nodes that are part of active routes. Every HELLO_INTERVAL milliseconds, the node will check if it has sent a broadcast within the last HELLO_INTERVAL; otherwise, it may broadcast a Hello message to its neighbours. This value is set to 1000 milliseconds in RFC 3561.

We make use of the signal strength information which is already available in the periodic Hello messages to calculate the relative signal strength, $Rel_{ss}[x]$ between adjacent nodes (which are assumed to transmit at constant power). Hence, these values of $Rel_{ss}[x]$ are updated every 1000 ms between active nodes. The structure of each neighbour table is also modified accordingly to store the values of $Rel_{ss}[x]$ and P_x.

4 Adaptive Algorithm with Link Stability

Our adaptive algorithm, which uses link stability to reduce the network overhead, comprises of the following two components:

- L-REQ – limited forwarding of RREQ packets to highly mobile nodes
- L-REP – limited initiation of RREP packets by intermediate nodes

4.1 L-REQ

During the transfer of data packets, routes have to be established beforehand. In the event that such routes do not already exist between the source and destination nodes, the former broadcasts a RREQ (Route Request) packet to its neighbouring nodes, which is then propagated into the network via local broadcasts by the receiving nodes.

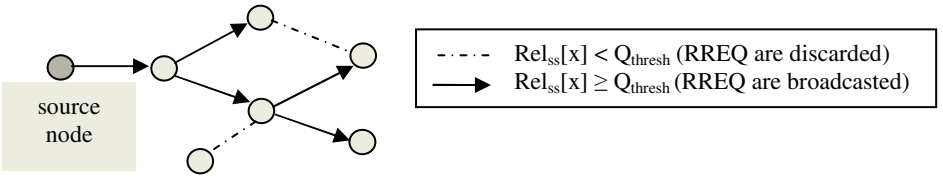


Fig. 1. L-REQ mechanism

The repetitive broadcasting of RREQ packets introduces a high number of control packets into the network. These control packets compete with the data packets for bandwidth, and may thus deteriorate network performance. Our L-REQ scheme (see

Fig. 1) works as follows: Since routes via unstable links are not preferred, each RREQ packet that is received by any particular node will only be broadcasted to neighbouring nodes if it is received from a node that has stable links. This will reduce the propagation of RREQ packets to highly dynamic nodes, since they are unlikely to maintain the route should one be established.

4.2 L-REP

When a node receives a new RREQ packet, it may respond with a unicast RREP if it is the destination or an intermediate node with a route that is fresh enough. However, RREP packets may be initiated by intermediate nodes that have already moved away from its previous neighbours. This can lead to the establishment of broken routes, resulting in more RERR packets being released into the network.

In our proposed L-REP scheme (see Fig. 2), if the intermediate node has a valid route to the destination, it may only unicast a RREP back to the source if the adjoining link is relatively stable. If the link between a pair of nodes has a low relative signal strength $Rel_{ss}[x]$, link breakages tend to occur more often and this will increase the amount of control overhead in the network. Hence, we restrict the probability of this happening by preventing RREPs that are otherwise initiated from intermediate nodes which form relatively unstable links with the neighbouring node in consideration.

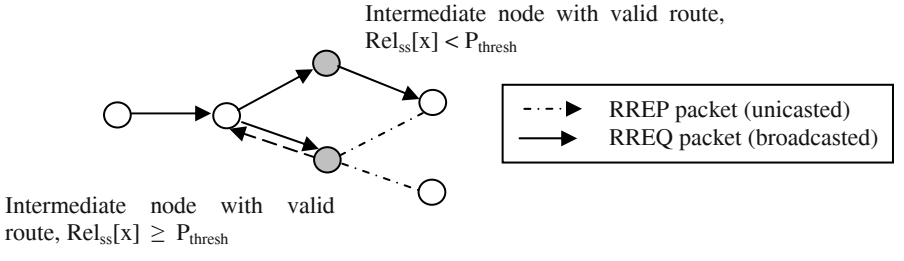


Fig. 2. L-REP mechanism

4.3 Adaptive Algorithm

Combining the two mechanisms together, our adaptive algorithm works as follows:

During the transfer of data packets, a route discovery process will be initiated when there is no existing route to the targeted destination. The source node broadcasts a RREQ packet to its neighbouring nodes, which then calculate the relative signal strength in order to determine the link stability of the adjoining link. If the link is very unstable, the RREQ is discarded immediately because unstable links are unlikely to last and may result in many broken links. If the link is fairly stable, the receiving node will broadcast the RREQ to its other neighbours.

When the RREQ is intercepted by an intermediate node that has a valid path to the targeted destination, it will compute its relative signal strength $Rel_{ss}[x]$ with the adjacent node with which a route may be established. A low value for $Rel_{ss}[x]$ indicates

that the link between the two nodes is fairly unstable, and it will not generate a RREP packet. Instead, it will continue to forward the RREQ packets to other nodes. The RREQ packet will eventually initiate a RREP packet from either the destination node or an intermediate node that has a high value of $Rel_{ss}[x]$ with the respective neighbouring node. RREP packets are then unicasted back to the source node.

Our two mechanisms aim to reduce the overall control overhead in the network by reducing the number of control packets that are being released. These control packets include RREQs, RREPs, RERRs as well as Hello messages that are used to provide local connectivity information. With lesser control overhead, there is less contention for bandwidth with the data packets, which leads to improved network performance.

5 Simulation Results and Analysis

We apply our proposed methods of adapting protocols (i.e. L-REQ and L-REP) on AODV-LR, an enhanced version of AODV [9] with local repair, and evaluate them using the following performance measures:

- Control overhead – total number of RREQ, RREP, RERR and Hello packets that are being propagated into the network.
- Packet delivery ratio – total number of data packets received as a fraction of the total number of packets originated from all the nodes in the network.
- End to end delay – time taken to transmit a packet from source to destination.
- Throughput – total number of successfully delivered data (in kilobytes).

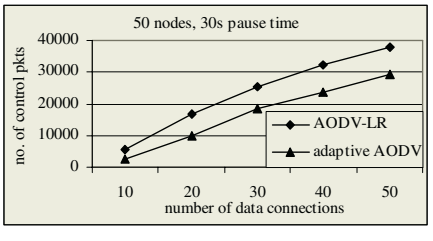


Fig. 3. No. of control pkts vs data load

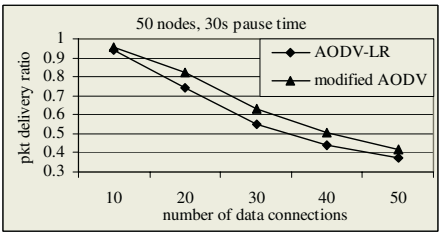


Fig. 4. Packet delivery ratio vs data load

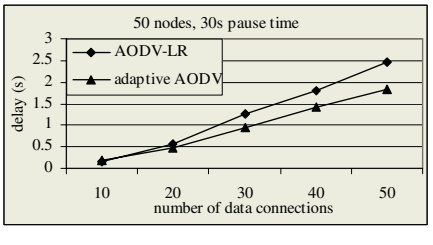


Fig. 5. Delay vs data load

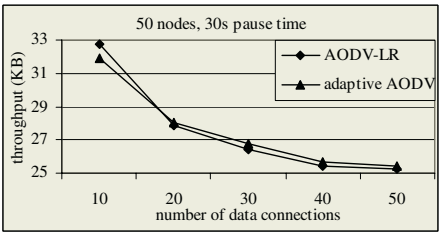


Fig. 6. Throughput vs data load

Simulations are run on GloMoSim [10], a simulation platform for networks. Each scenario is run for 300 seconds with different seeds, and the measurements are averaged to reduce the randomness of the mobility patterns. We use the Random Waypoint mobility model, where nodes move towards randomly selected destinations with speeds between $10ms^{-1}$ to $20ms^{-1}$ and rest there for a specified pause time. The terrain size is set to 2000×2000 metres. The CBR (Constant Bit Rate) generator is used to emulate data traffic between randomly selected nodes. At time intervals of 100ms, the selected nodes transmit 512 bytes of data for specified time intervals.

Figures 3-6 show the comparisons between AODV-LR and the adaptive AODV enhanced with the L-REQ and L-REP schemes. Nodes are uniformly placed and the pause time is set to 30s. Fig. 3 shows a marked decrease in the total number of control packets, which comprises of RREQ, RREP, RERR and Hello messages. This is expected, since the total number of RREQ packets (and hence RREP packets) that are being propagated into the network is greatly reduced, with the use of a threshold, Q_{thresh}. The use of P_{thresh} in the L-REP scheme also reduces the number of link breakages and thus the number of RERR messages. We also discover that our adaptive schemes have indirectly reduced the number of Hello packets being transmitted by nodes. This is because the restriction of propagation of control packets between unstable links reduces the number of nodes which receive RREQs from highly mobile nodes and become part of active routes which are prone to link breakages.

With a significant decrease in network overhead caused by control packets, there is less congestion in the network. This leads to a higher packet delivery ratio since there is now lesser contention for bandwidth between data and control packets, as shown in Fig. 4. We observe shorter end to end delay and increased throughput in Fig. 5 and Fig. 6 respectively. With the formation of more stable links within routes, the probability of link breakages decreases. This reduces the frequency of route repairs and route requests and hence reduces the end to end delay during the transfer of data packets. As such, the overall throughput will also increase because more data packets are delivered within the specified time intervals. There is however, a slighter longer delay and lower throughput for 10 data connections because there is already very little congestion in such networks. Hence, restriction of control packets for the formation of routes may lead to longer times needed for route establishment.

We also compared the performance of AODV-LR and the adaptive AODV using the Random Waypoint mobility model with 0s pause time, which emulates continuous random motion of nodes in the network. Similar to the previous set of results with 30s pause time, there are significant improvements in network performance.

Our next set of simulation explores the effects of our adaptive schemes on large scale networks. Different number of nodes are uniformly placed and simulated under the Random Waypoint mobility model with a pause time of 30s. 10 data connections are established at varying time intervals, each one transmitting 512 bytes of CBR traffic at periodic intervals of 100ms. The comparative results are shown in Fig. 7-10.

In Fig. 7, there is a clear reduction in the number of control packets that are being released into the network. There is also higher packet delivery ratio for the varying numer of nodes in the network, as shown in Fig. 8. This improvement is more pronounced in dense networks with more than 100 nodes, because comparatively fewer nodes are involved in the sending of control packets via unstable links.

There are also significant improvements in the total end to end delay and the network throughput as shown in Fig. 9 and Fig. 10 respectively. However, there is slightly higher end to end delay and lower throughput for networks with less than 100 nodes, because relatively more nodes are involved in the data connections, and limiting RREQs and RREPs under these situations can lead to longer times needed for route establishment. More data packets may also be dropped during the longer wait, which can result in lesser throughput.

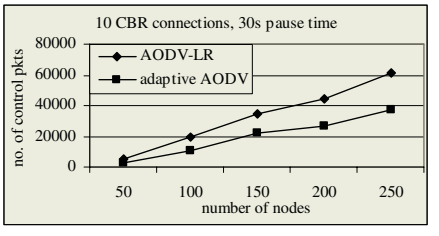


Fig. 7. No. of control pkts vs network size

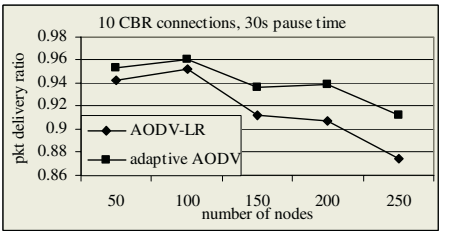


Fig. 8. Packet delivery ratio vs network size

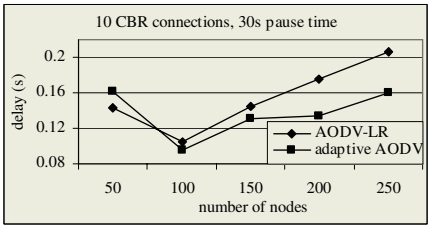


Fig. 9. Delay vs network size

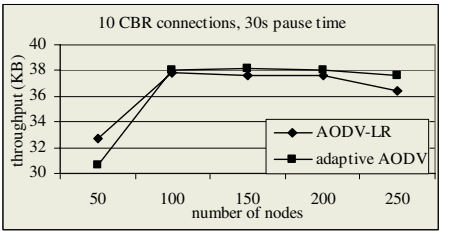


Fig. 10. Throughput vs network size

In all our simulations, we have set Q_{thresh} = -0.25 and P_{thresh} = -0.5, which are optimal values obtained through extensive simulations. These values allow for slight deviations in the signal strength being received by neighbouring nodes, which could be caused by fading, interference, or node mobility. Having negative thresholds allow links to be maintained even when the stability of the link decreases slightly during route establishment. In addition, P_{thresh} allows for a greater tolerance in the deterioration of link stability because more preference is given to existing, valid routes which have already been established, as this reduces the control overhead incurred by propagating more RREQ packets in the network during the route discovery process.

6 Conclusion and Future Work

Link stability is a metric that can provide precise feedback on the behavioral characteristics of a network, such as the mobility pattern of nodes and the estimated route lifetime between nodes. We have proposed a method to measure link stability using the received signal strengths between neighbouring nodes at periodic time intervals. This eliminates the need for additional infrastructure such as GPS and also takes into the account the effects of noise interference and other factors such as the power levels of the transmitting and receiving nodes.

We have suggested two mechanisms to adapt any arbitrary reactive routing protocols using link stability as a determining metric. L-REQ attempts to control the flooding of RREQs in the network, which is a persistent problem faced by many existing ad hoc routing protocols. With the use of Q_{thresh}, RREQs are not broadcasted in cases where routes are likely to break frequently. L-REP addresses the problem caused by route replies originating from intermediate nodes with valid routes to the targeted destinations, but which may cause unstable links to be formed. It limits the formation of such unstable routes with the use of another threshold, P_{thresh}. Pairs of nodes which do not satisfy the criteria have to forward the RREQ packets instead of unicasting a RREP packet back to the source node.

We have implemented L-REQ and L-REP on top of AODV-LR, an enhanced version of the AODV routing protocol with local repair, and simulation results have highlighted that our adaptive schemes can improve network performance in terms of control overhead, packet delivery ratio, end to end delay as well as overall throughput. As very little modification to the protocol architecture has been made, our adaptive scheme is also able to interoperate with the existing unmodified protocol.

We are currently investigating the effects of our adaptive schemes on other types of mobility models, such as the Reference Point Group Mobility (RPGM) model, which simulates group movements are based on the path traveled by a logical centre. Our continued research area includes developing other adaptive schemes based on network characteristics such as traffic characteristics and patterns, etc.

References

1. H. X. Tan and W. K. G. Seah, Dynamically Adapting Mobile Ad Hoc Routing Protocols to Improve Scalability, Proceedings of the *IASTED International Conference on Communication Systems and Networks* (CSN 2004), Marbella, Spain, Sep 1-3, 2004.
2. J. Boleng, W. Navidi and T. Camp, Metrics to Enable Adaptive Protocols for Mobile Ad Hoc Networks, Proceedings of the *International Conference on Wireless Networks* (ICWN 2002), Las Vegas, Nevada, USA, Jun 24-27, 2002.
3. R. Dube, C. D. Rais, K. Y. Wang and S. K. Tripathi, Signal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks, *IEEE Personal Communications*, Vol. 4, No. 1, pp. 36-45, 1997.
4. S. Krco and M. Dupcinov, Improved Neighbour Detection Algorithm for AODV Routing Protocol, *IEEE Communications Letters*, Vol. 7, No. 12, Dec 2003.

5. M. Abolhasan and T. A. Wysocki, Displacement-based Route Update Strategies for Proactive Routing Protocols in Mobile Ad Hoc Networks, *Workshop on the Internet, Telecommunications and Signal Processing* (WITSP 2003), Coolangatta, Gold Coast, Australia, Dec 8-11, 2003.
6. J. Boleng and T. Camp, Adaptive Location Aided Mobile Ad Hoc Network Routing, Proceedings of the *23rd IEEE International Performance, Computing and Communications Conference* (IPCCC 2004), Phoenix, Arizona, USA, Apr 15-17, 2004.
7. O.Tickoo, S. Raghunath and S. Kalyanaraman, Route Fragility: A Novel Metric for Route Selection in Mobile Ad Hoc Networks, Proceedings of the *IEEE International Conference On Networks* (ICON 2003), Sydney, Australia, 2003.
8. P. Basu, N. Khan and T. D. C. Little, A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks, Workshop on Wireless Networks and Mobile Computing (in IEEE ICDCS 2001), Phoenix, Arizona, USA, Apr 2001.
9. S. J. Lee, E. M. Belding-Royer and C. E. Perkins, Scalability Study of the Ad Hoc On-Demand Distance-Vector Routing Protocol, *International Journal of Network Management,* Vol. 13, No. 2, Mar 2003.
10. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia and M. Gerla, GloMoSim: A Scalable Network Simulation Environment, *UCLA Computer Science Department Technical Report 990027,* 1999.

Performance Analysis of Secure Multipath Routing Protocols for Mobile Ad Hoc Networks

Rosa Mavropodi, Panayiotis Kotzanikolaou, and Christos Douligeris

University of Piraeus,
Department of Informatics,
80 Karaoli & Dimitriou, Piraeus 185 34, Greece
{rosa, pkotzani, cdoulig}@unipi.gr

Abstract. In the Mobile Ad Hoc Network (MANET) paradigm, multipath routing protocols were initially proposed due to QoS needs, since they do not require initiation of the route discovery process after each link disconnection. Moreover, research on MANET routing security has shown that multipath routing provides increased resilience against security attacks of collaborating malicious nodes. Towards this direction, several secure multipath routing protocols have been recently proposed in the literature, which indeed provide such increased security protection for critical applications. However, embedding security mechanisms always imposes extra burden to the route discovery process. In this paper, we evaluate the performance of the existing secure multipath routing protocols for MANET through extensive simulations in various traffic scenarios.

1 Introduction

Multipath routing protocols were initially proposed to ensure QoS in mobile ad hoc networks. Maintenance of multiple routes towards a destination, prevents initiation of a new path discovery from the source node, each time there is a link failure. Furthermore, the existance of multiple paths may prevent node congestion, since it balances the traffic load through alternative routes. Examples of ad hoc multipath routing protocols are given in [9, 7, 10, 11, 5]. The route discovery may stop when a sufficient number of paths is discovered (*e.g.* [11]) or when all possible paths are detected (*e.g.* [3]). The protocols of the second case, are also known as complete. Multipath routing protocols can be *node-disjoint* (*e.g.* [11]) or *link-disjoint* (*e.g.* [6]) if a node (or a link) cannot participate in more than one path between two end nodes.

Apart from the multipath routing protocols that aim to increase efficiency, several multipath routing protocols have been proposed,recently, in order to provide additional security services. More specifically, the secure multipath routing protocols of [8, 2, 4] were designed in order to resist Denial of Service (DoS) attacks of collaborating malicious nodes, which single path protocols fail to encounter. Indeed, with single path routing protocols it is trivial for an adversary

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 269–278, 2005.

to launch a DoS attack, even if security measures are taken. A malicious node controlled by the adversary may participate passively in the routing path between two end nodes and may behave as a legitimate intermediate node. The malicious node can stop the communication at any time it seems most advantageous to the adversary. Although communication may be cryptographically protected, network characteristics (such as variation in traffic) or external factors may be used by the adversary in order to identify the proper time to disrupt communication. Even though the end nodes may initiate a new route request after the DoS attack, the time required to establish the new path may be critical. A dedicated and skilful adversary may thus identify the most critical nodes and disable their single routing paths, by compromising a small fraction of nodes.

Multipath routing protocols can be resilient to DoS attacks and may protect network availability from faulty or malicious nodes [1]. Indeed, if there exist k node-disjoint paths between two end nodes, the adversary should compromise at least k nodes - and more particularly at least one node in each path - in order to control their communication. A secure multipath routing protocol must be node-disjoint. Otherwise, a malicious node would be allowed to participate and consequently control more than one path. Thus, a single malicious node may manipulate the routing protocol and in this way it may compromise all the available routes between two end nodes.

In order to achieve resilience to DoS attacks, a multipath routing protocol should be properly enhanced with cryptographic means, which will guarantee the integrity of a routing path and the authenticity of the participating nodes. Towards this direction, three secure multipath routing protocols have been recently proposed; the Secure Routing Protocol (SRP) [8], the multipath routing protocol of [2] and the Secure Multipath Routing protocol (SecMR) [4]. The secure multipath routing protocols of [8, 2, 4] may guarantee at a certain level the availability of the communication against DoS attacks of a bounded number k of collaborating malicious nodes, by employing $k+1$ node-disjoint routing paths between two communicating nodes. However, the cryptographic protection in the route discovery of the secure multipath routing protocols will naturally increase the control overhead and until now, the efficiency of the secure multipath routing protocols for ad hoc networks has not been estimated.

In this paper, we evaluate the performance of the currently proposed secure multipath routing protocols of [8, 2, 4] by simulating their behavior in various traffic scenarios. In section 2, we briefly describe the examined routing protocols. In section 3, we present the simulation results, while in section 4 we discuss possible enhancements and we conclude this paper.

2 A Description of the Examined Secure Multipath Routing Protocol

In this section we briefly describe the route discovery process of the examined secure multipath routing protocols, along with some comments on the security properties of each protocol.

2.1 The SRP Protocol

SRP [8] was initially developed having in mind general security considerations of ad hoc networks. The basic considerations of the SRP protocol are integrity protection of the routing paths and authentication of the end nodes. The route discovery of the SRP can be used to discover multiple node-disjoint paths.

Before the propagation of a route request query, the source node assigns to it unique identifiers, in order to avoid replay attacks. When an intermediate node receives a route request, it checks whether it has already processed a query originating from the particular source node with the same identifiers, in order to drop it. Otherwise, it adds itself to the routing path and it forwards the request. In this way, an intermediate node can only participate in a single path between two end nodes and the paths that will be discovered will be node-disjoint. When the target node receives a route request query, the node checks the authenticity of the request be using a symmetric encryption key - a security association - which the two end nodes are supposed to share prior to the request. The route reply query will also be protected with the same security association, in order to protect the integrity of the routing paths. The protocol finds a number of node-disjoint routing paths between the source and the destination, which can be used for multipath communication.

The SRP protocol is very efficient since it restricts security checks at the end nodes only and it uses efficient symmetric key encryption. A problem of this protocol is that it does not authenticate the intermediate nodes which may lead to several impersonation attacks and in this way reduce the resilience of the protocol to DoS attacks [4]. For example, a malicious intermediate node may participate with fake identities to several paths, rendering multipath routing insecure. Furthermore, the protocol is not complete in the discovery of the existing node-disjoint multiple paths, *i.e.* although the paths discovered are node-disjoint the protocol may not discover all the existing node-disjoint paths between the two end nodes, depending on the propagation conditions of the query.

2.2 The Secure Multipath Routing Protocol of Burmester and Van Le

The secure multipath routing protocol of [2] is based on the Ford-Fulkerson MaxFlow algorithm. The propagation of a route request query assures that a query will reach any intermediate node within a pre-defined maximum hop distance. During the route request propagation, a node that receives a route query message for the first time, appends its neighborhood information along with a signature and re-broadcasts the message along with all the previously received query information. When the request query message reaches the destination, the destination node uses the received information in order to estimate the current connectivity of the intermediate nodes that answered the request query. In this way, the destination node can construct the complete set of the existing node-disjoint paths.

This protocol satisfies all the security requirements of multipath routing, since it authenticates all participating nodes, while it also protects the integrity of the routing paths. Furthermore, it satisfies completeness, *i.e.* it discovers all existing paths bounded with a TTL or maximum hop field. However, the propagation of the route request query is not efficient in terms of computation and space costs. The message size of a route request may increase to intolerable levels, since it contains information regarding the connectivity of all previous nodes. Furthermore, the use of digital signatures by the intermediate nodes of each route request message costs both in delay and processing power and may not be affordable for typical equipment.

2.3 The SecMR Protocol

In order to reduce the cost of node authentication, the SecMR [4] protocol works in two phases. The first phase is the neighboring authentication phase which is repeated in periodic time intervals. During this phase, nodes in range are mutually authenticated through digital signatures. Each node n_i constructs a set N_i that contains the identifiers of its authenticated neighbors. The neighborhood set is then used in the second phase of the protocol, which involves the route discovery. The advantage of using a separate authentication phase is that the number of signatures and verifications performed by each node is bounded in each authentication period and does not depend on the number of paths that the node will participate in for a given authentication period.

A route request message in the SecMR protocol contains three independent lists of nodes, in order to reduce the cost of the route discovery. The *RouteList* is the list of the intermediate nodes participating in a routing path. The $NexHop$ list contains the possible next participants of a particular route query. Finally, the *ExcludeList* holds the nodes that are not allowed to participate in the particular instance of the route request query.

An intermediate node n_i receiving a query will process the query, provided that: i) it is listed in the $NextHop$ list, ii) it does not already belong to the *RouteList* and iii) it is not listed in the *ExcludeList*. Processing the request involves updating the lists included in the query. The updated *RouteList*, is constructed by appending its identifier to the received *RouteList*. The updated *ExcludeList* is generated by appending the rest of the nodes included in the received $NextHop$ list, into the received *ExcludeList* (duplicates are removed). Finally, the updated $NextHop$ list is generated as the list of the neighbors N_i of the node n_i that executes the route query (again, node identifiers already participating in another list are removed). Now, the node n_i updates the query thread with the new lists and broadcasts it.

The use of the *ExcludeList* and the $NextHop$ list is a key element for the efficient propagation of a route request. The $NextHop$ list restricts the query to propagate only through mutually authenticated nodes. The use of the *ExcludeList* dynamically generates non-cyclic "threads" of the request in an optimized way. By dynamically generating threads of a request, the algorithm eventually discovers all the existing node-disjoint paths for a pre-defined max-

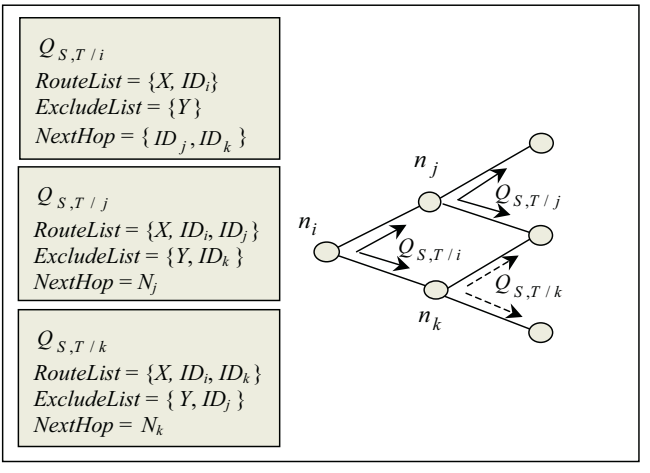


Fig. 1. Threading of a route request query

imum hop distance and only a limited number of redundant paths. To clarify the threading of a request query, consider the following scenario. Let n_i be an intermediate node that broadcasts the request $Q_{S,T}$ for the source and target nodes S and T respectively, to its neighbors n_j, n_k, after it has processed it (see figure 1). In order to distinguish the various threads of the request query, we denote as $Q_{S,T/i}$ the thread that is processed by node n_i. Thus, the thread $Q_{S,T/i}$ will contain the lists: $RouteList = \{X, ID_i\}$, $ExcludeList = \{Y\}$ and $NextHop = \{ID_j, ID_k\}$, where X and Y denote sequences of node identifiers.

Both nodes n_j and n_k will process the request (supposing that $ID_j, ID_k \notin X, Y$). Node n_j will add its identifier to the $RouteList$, add the identifier of n_k to the $ExcludeList$ and update the $NextHop$ list with its own neighborhood. Thus, the updated threads of the request query will become $Q_{S,T/j}$, $Q_{S,T/k}$, containing the updated lists shown in figure 1. Each of these threads will propagate towards T, with the limitation that the thread $Q_{S,T/j}$ is not allowed to pass from the node n_k and vice versa. This forces the query to move only to more distant nodes of S towards T. The threads that return backwards tend to decline in a short time, when they reach a node closer to S that has been previously excluded.

At the end of the route discovery, the target and the source nodes will use a symmetric key contained in the route request message, in order to verify the integrity of the discovered paths.

3 Efficiency Analysis

Our study involves a comparison of the route request query between the SRP protocol [8], the complete multipath routing protocol of [2] and the SecMR pro-

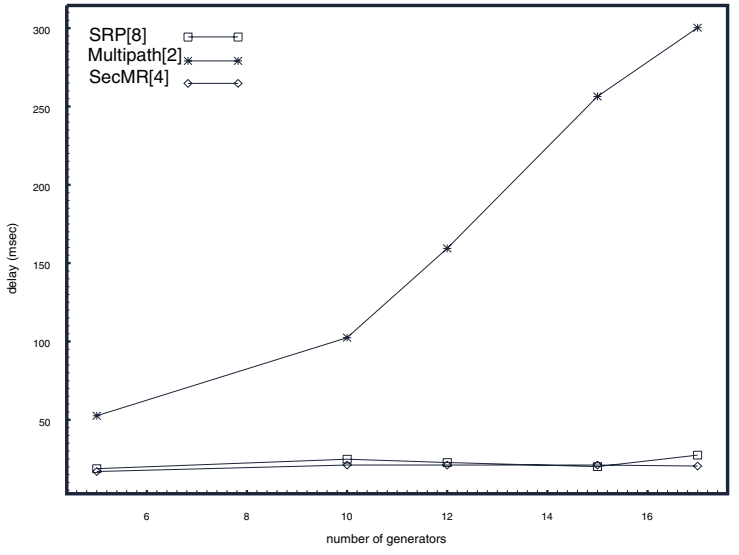


Fig. 2. Total data packet delay for data packet interarrival time 1 sec

tocol [4]. We implemented the simulator within the NS-2 library. Our simulation modelled a network of 50 hosts placed randomly within a $670 \times 670m^2$ area. Each node has approximately 5 hops as neighbors. Each node has a radio propagation range of 150 meters and channel capacity was 2 Mb/s. The minimum and maximum speed is set to 0 and 20 m/s, respectively. This setup leads to a relatively dense network distribution with medium to high mobility and with medium mean connectivity. The size of the data payload was 512. Each run executed for 600 sec of simulation time. We used the IEEE 802.11 Distributed Coordination Function (DCF) as the medium access control protocol. The traffic generators were developed to simulate constant bit rate sources. The sources and the destinations are randomly selected with uniform probabilities. The destination of the traffic wait for 5 seconds until it assumes that all possible paths have been found. We generated various traffic scenarios by using different number of sources and scalar intrarrival data packet's time.

A free space propagation model with a threshold cutoff was used in our experiments. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, *i.e.*, radio capture.

Figure 2 shows the average delay of the received data packets for data interarrival time equal to 1 sec. We can observe from the results that both SRP [8] and SecMR [4] outperform the multipath protocol of [2] especially when the number of data generators increases, which depicts high traffic conditions. In both SRP and SecMR the number of generated messages during the route discovery process are kept in sufficient low levels while the ones of Multipath[2] tent to flood the network. This is because in the multipath protocol of [2], each intermediate node forwards all the route requests that reaches it for a given source, destination

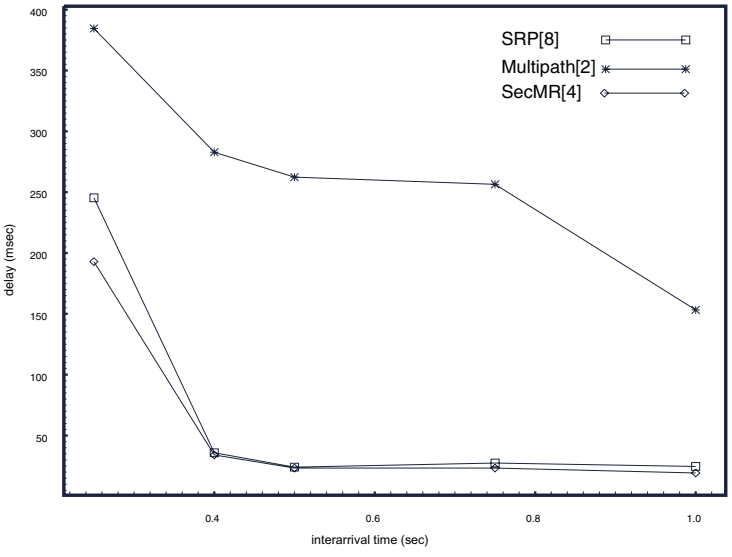


Fig. 3. Total data packet delay for 17 data sources

and sequence number, while SRP forwards only the first and SecMR performs a selective forward with the use of the exclude list. This flooding of the network results in higher delay in the data packet delivery. On the other hand, as the node's movement is rather high, the discovered paths of SRP are insufficient thus resulting in degradation of its performance. Figure 3, which presents the average delay of the received data packets to a network with multiple data sources with scalar interarrival time, strengthens the above observations. Indeed, as shown in figure 3, the SecMR protocol handles high traffic conditions better.

Figures 4 and 5 present the average total time that route request messages travel through the network. Figure 4 presents the average total time that route request messages travel through the network versus the number of data generators, for interrival time of 1 sec. Again, an increase in traffic leads to a proportional increase of the time that the route request messages are alive. In the secure multipath protocol of [2] a route request travels for a longer time than in the other two protocols, as the request is being forwarded to all nodes in range, many of which will not be included into one of the discovered paths. The route request of the SRP propagetes the request towards the destination faster than the other protocols, since it rejects any variant of a specific request. The route request of the SecMR has slightly longer living time than SRP. This is reasonable as it attempts to ensure descovery of the complete set of existing node-disjoint paths. Furthermore, the SecMR makes sure that all its neighboring nodes have contributed to the route discovery, either by participating to the *RouteList* (*i.e.* to a routing path) or by avoiding to re-process the same thread of the query (*i.e.* by participating into the *ExcludeList* of the query thread). This is also obvious

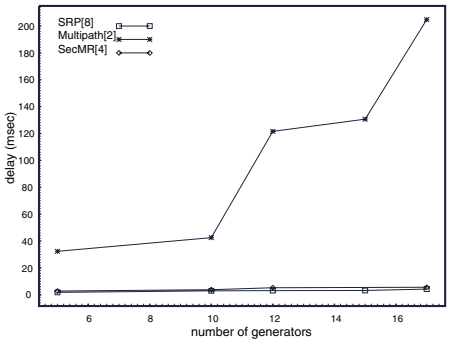


Fig. 4. Average living time of route request messages for data packet interarrival time 1 sec

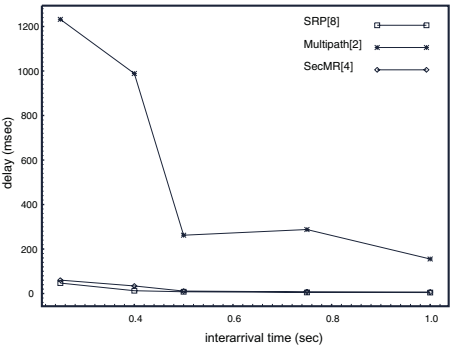


Fig. 5. Average living time of route request messages for 17 data sources and scalar data packet interarrival times

in figure 5, which presents the average total time that a request travels through the network, per send rate, for multiple data sources.

Figures 6 and 7 illustrate the average time taken for a request message to reach the destination. In dense traffic conditions, while in SRP and SecMR the required time is comparable to the average time the request stays alive in the network (as illustrated in figures 4 and 5 respectively), in the case of the protocol of [2] the request stays alive in the network almost 8 times more after the first request query thread has reached its destination. This means that the redundant request messages will exist in the network for a long time, causing the network to experience high delays. Finally, figures 8 and 9 illustrate the average throughput, per interrival time, of data and route control messages respectively for 17 data sources. Multipath serves less data packets then SecMR and SRP (figure 8) in contrast to the number of the routing control messages (figure 9).

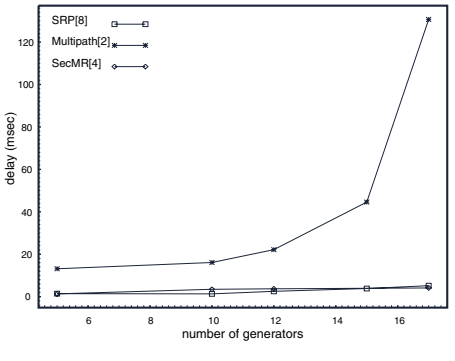


Fig. 6. Route Discovery delay for data packet interrival time 1 sec

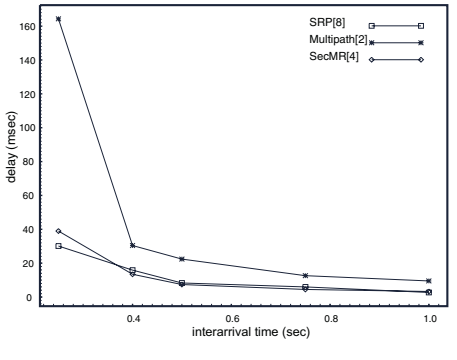


Fig. 7. Route Discovery delay for 17 data sources

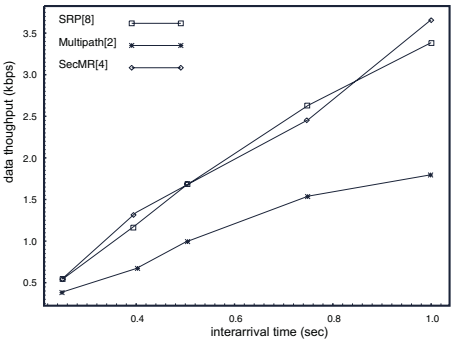


Fig. 8. Average throughput of data packets for 17 data sources

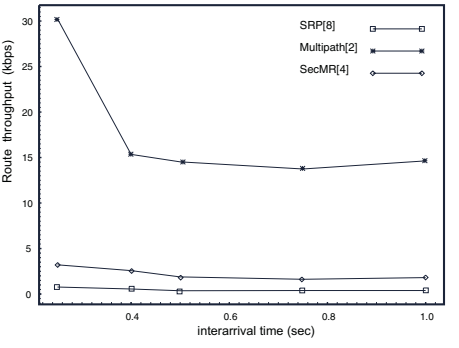


Fig. 9. Average throughput of route control packets

4 Discussion and Future Work

The simulation results provide significant evidence about the efficiency of the route request propagation of the examined secure multipath routing protocols. However, it should be mentioned that these ratings reflect the examined network design scenario, with relatively dense network distribution, medium to high mobility and medium mean node connectivity. In this scenario, SRP performs better than the other two protocols, SecMR follows in short distance, while the protocol of [2] seems to be the heavier.

From a security point of view, the ranking is reversed. The protocol of [2] achieves all the required security properties, to provide maximum resilience against DoS attacks of collaborating malicious nodes. It provides completeness in the route discovery process and it explicitly authenticates all the intermediate nodes in each routing path. The SecMR protocol also achieves completeness and it provides implicit authentication of the intermediate nodes, since node authentication is performed once for a discrete time period. Finally, the SRP protocol does not provide the complete set of node-disjoint paths, and it provides only end-to-end authentication.

Based on the above observations and the simulation results, the protocol of [2] can be considered suitable for security critical ad hoc network applications, but its applicability can only be considered in networks with low mobility and relatively low density. In situations with high mobility and high node density, the protocol would saturate the network, since it would lead to long route request messages which would exist in the network for long time.

The SecMR protocol seems most appropriate for ad hoc networks that require high security protection and they present medium to high mobility and medium node density. Indeed, in such situations the SecMR protocol has comparable efficiency with the SRP, while it offers increased security level. Moreover, as the node mobility increases, the SecMR shows better performance than the SRP. This is due to the fact that the SRP discovers less paths than the other two

protocols and this forces the protocol to re-initiate route requests in shorter time that the other protocols, when nodes move and links are broken.

Finally, the SRP seems a suitable choice for several network configurations with increased node density. This is caused by the fact that the route request propagation avoids discovery of all the possible routes that each node could participate and in this way it converges faster. This however leads to a non-complete route discovery [4] and reduces the security resilience of the protocol to distributed DoS attacks. Thus, the SRP seems suitable for applications with medium security risks.

Regarding possible extensions of our work, we consider examining the behavior of the secure multipath routing protocols in various network configurations and arrival patterns. Furthermore, we consider examination of the behavior of the route reply and route maintenance algorithms of the examined protocols.

References

1. M. Burmester and Y. Desmedt, *Secure communication in an unknown network using certificates*, Advances in Cryptology - Asiacrypt 99, Lecture Notes in Computer Science Vol. 1716, Springer, 1999, pp. 274–287.
2. M. Burmester and T. van Le, *Secure multipath communication in mobile ad hoc networks*, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004) (Las Vegas), IEEE, April 2004.
3. Gunyoung Koh, Duyoung Oh, and Heekyoung Woo, *A graph-based approach to compute multiple paths in mobile ad hoc networks*, Lecture Notes in Computer Science Vol.2713, Springer, 2003, pp. 3201–3205.
4. P. Kotzanikolaou, R. Mavropodi, and C. Douligeris, *Secure multipath routing for mobile ad hoc networks*, Proceedings of the WONSS05 Conference (St. Moritz, Switzerland), IEEE, January 19-21 2005, pp. 89–96.
5. Sung-Ju Lee and Mario Gerla, *Split multipath routing with maximally disjoint paths in ad hoc networks*, Proceedings of ICC 2001 (Helsinki, Finland), IEEE, June 2001, pp. 3201–3205.
6. Mahesh K. Marina and Samir R. Das, *Ad hoc on-demand multipath distance vector routing*, ACM SIGMOBILE Mobile Computing and Communications Review **6** (2002), no. 3.
7. A. Nasipuri and S.R. Das, *On-demand multipath routing for mobile ad hoc networks*, Proceedings of IEEE INFOCOM99, 1999, pp. 64–70.
8. P. Papadimitratos and Z. Haas, *Secure routing for mobile ad hoc networks*, In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) (TX, San Antonio), January 2002.
9. Anand Prabhu Subramanian, A. J. Anto, Janani Vasudevan, and P. Narayanasamy, *Multipath power sensitive routing protocol for mobile ad hoc networks*, Lecture Notes in Computer Science Vol.2928, Springer, 2003, pp. 171–183.
10. A. Tsirigos and Z.J. Haas, *Multipath routing in the presence of frequent topological changes*, IEEE Communications Magazine **39** (2001), no. 11, 132–138.
11. Jie Wu, *An extended dynamic source routing scheme in ad hoc wireless networks*, Telecommunication Systems **22** (2003), no. 1-4, 61–75.

Lecture Notes in Computer Science: Packet Error Rate Analysis of IEEE 802.15.4 Under IEEE 802.11b Interference

Soo Young Shin[1], Sunghyun Choi[1], Hong Seong Park[2], and Wook Hyun Kwon[1]

[1] School of Electrical Engineering & Computer Science,
Seoul National University, Seoul, Korea
{wdragon, whkwon}@cisl.snu.ac.kr, shchoi@snu.ac.kr
[2] Dept. of Electrical and Computer Eng.,
Kangwon National University, Chuncheon, Kangwon-Do, Korea
hspark@kangwon.ac.kr

Abstract. In this paper, the packet error rate (PER) of IEEE 802.15.4 low rate wireless personal area network (WPAN) under the interference of IEEE 802.11b wireless local area network (WLAN) is analyzed. The PER is obtained from the bit error rate (BER) and the collision time. The BER is obtained from signal to noise and interference ratio. The power spectral density of the IEEE 802.11b is considered in order to determine in-band interference power of the IEEE 802.11b to the IEEE 802.15.4. The simulation results are shown to validate the numerical analysis.

1 Introduction

Recently, a low rate wireless personal area network (LR-WPANs), IEEE 802.15.4, has been standardized[1],[2]. The goal of the IEEE 802.15.4 is to provide a standard, which has the characteristics of ultra-low complexity, low-cost and extremely low-power for wireless connectivity among inexpensive, fixed, and portable devices such as sensor networks and home networks. To provide the global availability, the IEEE 802.15.4 devices use the 2.4GHz industrial scientific and medical (ISM) unlicensed band. Because this ISM band is commonly used for the low cost radios such as IEEE 802.11b (WLAN)[3] and IEEE 802.15.1 (Bluetooth)[4], an unrestricted access to the ISM band exposes the IEEE 802.15.4 devices to a high level of interference. Since the IEEE 802.15.4 and the IEEE 802.11b have been designed for different purposes, they can be coexisted within the communication range of each other. For example, the IEEE 802.15.4 network is used for a sensor and control network and the IEEE 802.11b network is used for a audio/video (A/V) network within a home. When a notebook is capable of supporting these two standards, the coexistence distance may be smaller than 1 m. Therefore, the coexistence performance of the IEEE 802.15.4 and the IEEE 802.11 needs to be evaluated.

Some related reseaches study the coexistence problem between the IEEE 802.15.4 and the 802.11b[5],[6],[7]. In [5], the packet error rate (PER) of the IEEE

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 279–288, 2005.

802.15.4 under the IEEE 802.11b and IEEE 802.15.1 is obtained by experiments only. In [6], the impact of an IEEE 802.15.4 network on the IEEE 802.11b devices is analyzed. However, the PER of the IEEE 802.15.4 packets is not considered. In [7], the PER of IEEE 802.15.4 under the interference of IEEE 802.11b is evaluated using simulation. To the best knowledge of the authors, the analysis of the PER of the IEEE 802.15.4 under the interference of the IEEE 802.11b has not been reported yet in the literature.

In this paper, the PER of the IEEE 802.15 4 under the interference of the IEEE 802.11b is analyzed using the bit error rate (BER) and the collision time. The BER is obtained from signal to interference and noise ratio (SINR). The collision time is defined as the time that an IEEE 802.15.4 packet experiences the interference by packets of the IEEE 802.11b. For accurate analysis, in-band interference power ratio of the IEEE 802.11b is obtained from the power spectral density of the IEEE 802.11b and the frequency offset. The frequency offset can be defined as the difference between the center frequencies of the IEEE 802.15.4 and the IEEE 802.11b. The analytic results are compared with the simulation results.

This paper is organized as follows. Section 2 briefly overviews the IEEE 802.15.4. In Section 3, the BER of the IEEE 802.15.4 under the IEEE 802.11b is evaluated. Section 4 describes the interference model of the IEEE 802.15.4 and the IEEE 802.11b. The PER is obtained in Section 4. In Section 5, comparisons between analytic and simulation results are shown. Finally, this paper concludes in Section 6.

2 IEEE 802.15.4 Overview

A new IEEE standard, 802.15.4, defines both the physical layer (PHY) and medium access control (MAC) sublayer specifications for low-rate wireless personal area networks (LR-WPANs), which support simple devices that consume minimal power and typically operate in the personal operating space (POS) of 10 m or less. Two types of topologies are supported in the IEEE 802.15.4: a one-hop star or a multi-hop peer-to-peer topology. However, the logical structure of the peer-to-peer topology is defined by the network layer. Currently, the ZigBee Alliance is working on the network and upper layers [8].

2.1 Operation in the ISM Bands and at Various Data Rates

The IEEE 802.15.4 defines two PHY layers, the 2.4 GHz and 868/915 MHz band PHYs. The unlicenced industrial scientific medical (ISM) 2.4 GHz band is available worldwide, while the ISM 868 MHz and 915 MHz bands are available in Europe and North America respectively. A total of 27 channels with three different data rates are defined for the IEEE 802.15.4: 16 channels with a data rate of 250 kbps at the 2.4 GHz band, 10 channels with a data rate of 40 kbps at the 915 MHz band, and 1 channel with a data rate of 20 kbps at the 868 MHz band.

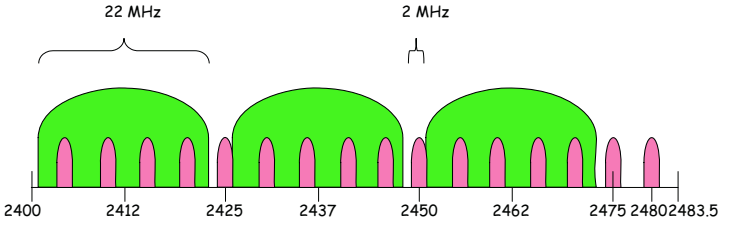


Fig. 1. IEEE 802.11b and IEEE 802.15.4 Channel Selection

The relationship between the IEEE 802.11b (non-overlapping sets) and the IEEE 802.15.4 channels at the 2.4 GHz is illustrated in Figure 1.

To prevent the interference between the IEEE 802.15.4 and the IEEE 802.11b, the standard of the IEEE 802.15.4 recommends to use the channels that fall in the guard bands between two adjacent the three IEEE 802.11b channels or above these channels. While the energy in this guard space will not be zero, it will be lower than the energy within the channels; and operating an IEEE 802.15.4 network on one of these channels will minimize interference between systems. However, if there will be more IEEE 802.15.4 networks, these four channels are not enough.

2.2 Different Data Transmission Methods and Low Power Consumption

An IEEE 802.15.4 network can work in either beacon-enabled mode or non-beacon-enabled mode. In beacon-enabled mode, a coordinator broadcasts beacons periodically to synchronize the attached devices. In non-beacon-enabled mode, a coordinator does not broadcast beacons periodically, but may unicast a beacon to a device that is soliciting beacons.

A superframe structure is used in beacon-enabled mode. The format of the superframe is determined by the coordinator. A superframe consists of an active part and an optional inactive part, and is bounded by the beacons. The length of a superframe (i.e., beacon interval, BI) and the length of its active part (i.e., superframe duration, SD) are determined by the beacon order (BO) and superframe order (SO), respectively. The active part of a superframe is divided into aNumSuperframeSlots (with the default value of 16) equal-sized slots, and a beacon frame is transmitted at the first slot of each superframe.

The active part can be further classified into two periods, a contention access period (CAP) and an optional contention-free period (CFP). The optional CFP may accommodate up to seven guaranteed time slots (GTSs) to provide the data with quality of service (QOS), 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. A slotted CSMA-CA mechanism is used for channel access during the CAP. All contention-based transactions shall be completed before the CFP begins. Moreover, all

transactions using GTSs shall be done before the time of the next GTS or the end of the CFP.

3 Bit Error Rate Evaluation of IEEE 802.15.4 Under IEEE 802.11b

The PHY of the IEEE 802.15.4 at 2.4 GHz uses offset quadrature phase shift keying (OQPSK) modulation with half-sine pulse shaping, which is equivalent to MSK[9]. Denote the E_b/N_o be the ratio of the average energy per information bit to the noise power spectral density at the receiver input, assuming an additive white Gaussian noise (AWGN) channel. Then the bit error rate (BER), P_B, can be expressed as

$$P_B = Q\left(\sqrt{\frac{2\gamma E_b}{N_o}}\right), \quad Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du \tag{1}$$

where $\gamma \simeq 0.85$[9].

In this paper, the indoor propagation model is assumed, and then, the path loss between transmitter and receiver can be expressed as:

$$L_p(d) = \begin{cases} 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) & , \ d \le d_0 \\ 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) + 10n\log_{10}\frac{d}{d_0} & , \ d > d_0 \end{cases} \tag{2}$$

where d and d_0 are the distance between the transmitter and receiver and length of line-of-sight (LOS), respectively, and λ is c/f_c, where c is the light velocity and f_c is the carrier frequency. Once the transmitter power is fixed like $P_{T,x}$, then the received power is obtained as $P_{R,x} = P_{T,x} \cdot 10^{-\frac{L_p(d)}{10}}$ where x is either IEEE 802.15.4 or IEEE 802.11b.

The bandwidth of the IEEE 802.11b is 22 MHz, which is much larger than that of the IEEE 802.15.4, 2 MHz. So the signal of the IEEE 802.11b, interferer, can be modeled as bandlimited AWGN to the IEEE 802.15.4 signal, user[10]. Then, the SINR can be determined by

$$SINR = \frac{P_c}{P_{N_o} + P_i} + ProcGain \tag{3}$$

where P_c, P_{N_o}, and P_i denote the power of the desired signal, the noise power, and interferer power, respectively. The $ProcGain$ is the spreading gain of IEEE 802.15.4. By replacing Eb/No in Eq. (1) with SINR in Eq. (3), the BER of the IEEE 802.15.4 under the IEEE 802.11b can be obtained.

Because the bandwidth of the IEEE 802.11b is 11 times that of the IEEE 802.15.4, in-band interference power of the interferer to the user is usually calculated as $P_{R,IEEE802.11b}/11$. However, the power spectral density of the IEEE 802.11b is not uniformly distributed across 22 MHz as illustrated in Figure 2 [11].

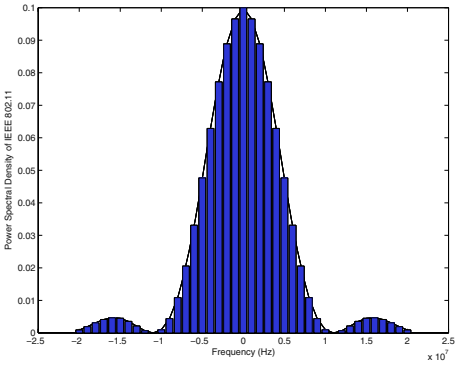


Fig. 2. Power Spectral Density of the IEEE 802.11b

Since the power is concentrated around the center frequency, the in-band power of the interferer to user is dependent on the frequency offset between the center frequencies of the user and interferer. For example, if the center frequency of the IEEE 802.15.4 is 2416 MHz and that of the IEEE 802.11b is 2418 MHz, then the center frequency offset is 2 MHz. In that case, the in-band interference power to user is about 17 % of the total power of the IEEE 802.11b.

4 Packet Error Rate Analysis of the IEEE 802.15.4 and the IEEE 802.11b

In this paper, IEEE 802.15.4 users are assumed to be transparent to IEEE 801.11b users, and vice versa. In other words, they transmit the packets without consideration of the channel state whether busy or not to make the worst case interference environments. If both standards use the carrier detection method (CCA mode 2) to determine the channel state rather than the energy detection (CCA mode 1), the transparency can be assumed without loss of generality.

Then, the interference model can be illustrated as shown in Figure 3. In Figure 3, T_X, L_X, and U_X denote the inter-arrival time, packet duration, and average random backoff time, respectively, where the subscript X is either Z for the IEEE 802.15.4 and W for the IEEE 802.11b. The other parameters are listed in Table 1. The T_C is the collision time.

Both the IEEE 802.11b and the IEEE 802.15.4 use carrier sense multiple access with collision avoidance (CSMA/CA) for medium access control. In the both protocols, nodes must perform a backoff process before transmitting a packet. However, in the IEEE 802.15.4, a channel is sensed only during the clear channel assessment (CCA) period, which occurs after finishing a backoff countdown. Accordingly, the backoff countdown occurs even during a busy channel period. The contention window of the IEEE 802.15.4 is doubled only when the channel is determined to be busy during the CCA period. On the other hand, the con-

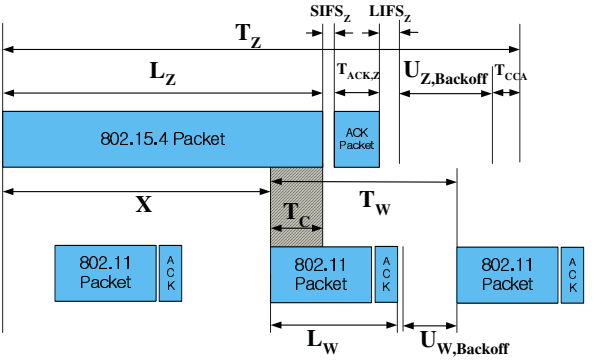


Fig. 3. Interference Model between IEEE 802.15.4 and IEEE 802.11b

Table 1. Parameters of the Interference Model

T_Z	inter-arrival time between two IEEE 802.15.4 packets
L_Z	duration of IEEE 802.15.4 packet
$SIFS_Z$	short IFS of IEEE 802.15.4
$LIFS_Z$	large IFS of IEEE 802.15.4
$T_{ACK,Z}$	duration of IEEE 802.15.4 ACK packet
U_Z	average backoff time of IEEE 802.15.4
T_W	inter-arrival time between two IEEE 802.11b packets
L_W	duration of IEEE 802.11b packet
$SIFS_W$	short IFS of IEEE 802.11b
$DIFS$	DCF IFS of IEEE 802.11b
$T_{ACK,W}$	duration of IEEE 802.11b ACK packet
U_W	average backoff time of IEEE 802.11b

tention window size is reset even for a retransmission after a unsuccessful packet transmission.

Because of the transparency, the transmissions of the IEEE 802.15.4 and IEEE 802.11b are independent. Since the both protocols transmit packets without consideration of the channel state, the contention window is not changed by the busy channel. The transmission of the IEEE 802.11b packet can be assumed as error-free because the narrow band and low-power 802.15.4 signal is not bad enough to affect the transmission of the 802.11b packets. So, there is no increase of the contention window of the IEEE 802.11b. Therefore, in both protocols, the backoff time is randomly chosen within the minimum contention window, i.e, CW_{min}.

Then, the inter-arrival times, T_W, T_Z can be obtained as:

$$T_Z = L_Z + T_{CCA} + SIFS_Z + T_{ACK,Z} + U_Z \tag{4}$$

and

$$T_W = L_W + SIFS_W + T_{ACK,W} + DIFS + U_W \tag{5}$$

where T_{CCA} denote the CCA time of the IEEE 802.15.4 and $U_X = CW_{min,X}/2$.

Assume that the time offset x is assumed uniformly distributed in $[0, T_Z)$, then, the collision time, T_C can be obtained as :

$$T_C(x) = \begin{cases} L_Z - 2\left(T_W - L_W\right) - x + nT_W, \\ \quad if \quad nT_W \le x \le L_Z - 2T_W + nT_W \\ 2L_W\,, \\ \quad if \quad L_Z - 2T_W + nT_W < x \le T_W - L_W + nT_W \\ 3L_W - T_W + x - nT_W, \\ \quad if \quad T_W - L_W + nT_W < x \le L_Z - (T_W + L_W) + nT_W \\ L_Z - 2\left(T_W - L_W\right), \\ \quad if \quad L_Z - (T_W + L_W) + nT_W < x \le \min\left(T_W + nT_W, T_Z\right) \end{cases} \tag{6}$$

where $n = 0, 1, 2, 3, 4$.

Now, the packet error rate (PER) is easily obtained from the BER and the collision time, T_C. For simplicity, acknowledgement (ACK) packets of both IEEE 802.11 and IEEE 802.15.4 are not considered. Let's denote the P_B and P_B^I be the BER without and with interference, respectively. If the bit duration of the IEEE 802.15.4 is b, then the PER, P_P, is expressed as

$$P_P = 1 - \left(1 - (1 - P_B)^{L_Z - \lceil T_C/b \rceil}\right)\left(1 - \left(1 - P_B^I\right)^{\lceil T_C/b \rceil}\right). \tag{7}$$

5 Comparative Evaluation

For simulation, the slotted CSMA/CA of the IEEE 802.15.4 model is developed using OPNET. The complementary code keying (CCK) modulation with 11 Mbps is used for the IEEE 802.11b. The payload size of the IEEE 802.15.4 is 105 bytes, and that of the IEEE 802.11 is 1500 bytes. The length of LOS, d_0, is 8 m and the path loss exponent, i.e. n, is 3.3. The transmitter power of IEEE 802.15.4 is 1 mW and that of IEEE 802.11b is 30 mW. The simulation scenario is shown in Figure 4.

For simplicity, only the IEEE 802.15.4 End_device and IEEE 802.11b WLAN_1 transmit data packets. The other nodes send only the ACK packets for the corresponding data packets. The distance between two IEEE 802.15.4 devices and that of the two IEEE 802.11b devices are fixed to 1 m. The distance between IEEE 802.15.4 Coordinator and the IEEE 802.11b WLAN_1 is d, which is variable.

Figure 5 shows the PER of the IEEE 802.15.4 under the interference of the IEEE 802.11b with 0 frequency offset. The distance between Coordinator and WLAN_1, d, varies from 1m to 10m.

In Figure 5, the "without PSD" term means that the in-band interference power is calculated as $P_{R,IEEE802.11b}/11$ where $P_{R,IEEE802.11b}$ is the received signal power of the IEEE 802.11b. On the other hand, the "with PSD" term

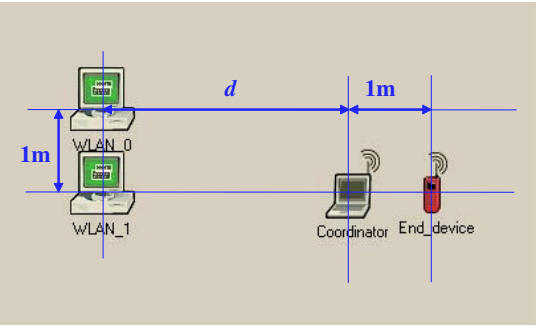


Fig. 4. Simulation Model between IEEE 802.15.4 and IEEE 802.11b

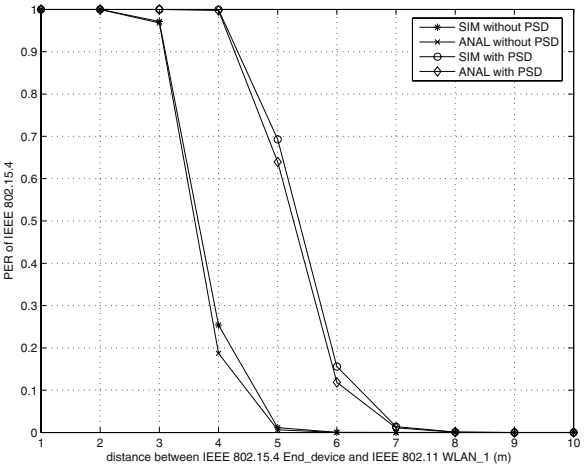


Fig. 5. PER of the IEEE 802.15.4 with/without considering the power spectral density of the IEEE 802.11b

means that the in-band interference power is obtained from the power spectral density of the IEEE 802.11b. Since the power of the IEEE 802.11b is concentrated at the center frequency as shown in Figure 2, the in-band interference power with 0 frequency offset is larger than $P_{R,IEEE802.11b}/11$. Therefore, the PER of the "with PSD" is larger than that of the "without PSD". Note that when the distance between the IEEE 802.15.4 Coordinator and IEEE 802.11b WLAN_1 is longer than 8m, the packet error rate of the IEEE 802.15.4 is smaller than 10^{-6} in both simulations.

In Figure 6, the in-band power of the IEEE 802.11b for the IEEE 802.15.4 varies with the center frequency offset between the IEEE 802.11b and IEEE 802.15.4. Since the SINR varies according to the in-band interference power, the PER varies as illustrated. If the in-band power is uniformly distributed as $P_{R,IEEE802.11b}/11$, the PER is obtained as a horizontal line, i.e, ANAL_WO, in

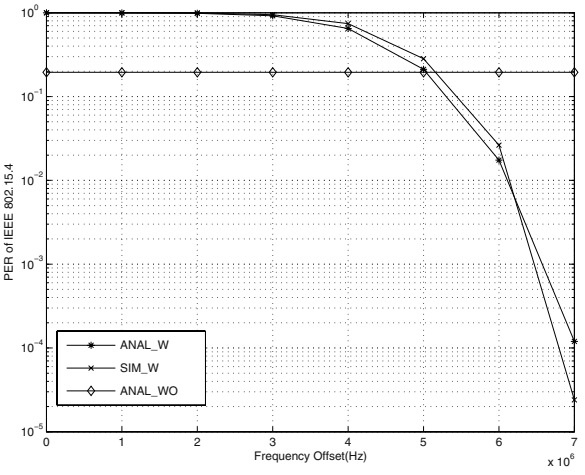


Fig. 6. PER of the IEEE 802.15.4 with the different frequency offsets to the IEEE 802.11b when d is fixed to 4m

the figure, which means that the PER is independent to the frequency offset. Note that if the frequency offset is larger than the 7 MHz, the interference of the IEEE 802.11b does not affect the PER of the IEEE 802.15.4.

6 Conclusion

In this paper, the packet error rate (PER) of IEEE 802.15.4 under the interference of IEEE 802.11b is analyzed. The PER is obtained from the bit error rate (BER) and the collision time. The BER of IEEE 802.15.4 is obtained from the offset quadrature phase shift keying (OQPSK) modulation. The collision time is calculated under assumption that the packet transmissions of the IEEE 802.15.4 and the IEEE 802.11b are independent. Because the bandwidth of IEEE 802.11b is larger than that of IEEE 802.15.4, the in band interference power of IEEE 802.11b is considered as the additive white gaussian noise (AWGN) for the IEEE 802.15.4. For an accurate calculation, the in-band interference power ratio of the IEEE 802.11b is considered with different frequency offsets between IEEE 802.15.4 and 802.11b. To obtain the ratio, the power spectral density of the IEEE 802.11b is considered. The simulation results are shown to prove the analysis.

If the distance between the IEEE 802.15.4 and 802.11b is longer than 8 m, the interference of the IEEE 802.11b is almost negligible to the performance of the IEEE 802.15.4, i.e., the packet error rate is smaller than 10^{-6}. If the frequency offset is larger than 7 MHz, the interference effect of the IEEE 802.11b is negligible to the performance of the IEEE 802.15.4. Therefore, three additional channels of the IEEE 802.15.4 such as 2420 MHz, 2445 MHz, and 2470 MHz can be used for the coexistence channels under the interference of the IEEE 802.11b.

The result of this paper can suggest the coexistence criteria for the IEEE 802.15.4 and IEEE 802.11b and be useful for designing and implementing networks using both IEEE 802.15.4 and IEEE 802.11b.

References

1. IEEE Std.802.15.4: IEEE Standard for Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) (2003)
2. Zheng, J., M.J., Lee: Will IEEE 802.15.4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard,IEEE Communications Magazine, Vol. 42 (2004) 140–146
3. IEEE Std.802.11: IEEE Standard for Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specificaton (1997)
4. IEEE Std.802.15.1: IEEE Standard for Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) (2004)
5. Sikora, A.: Coexistence of IEEE802.15.4 (ZigBee) with IEEE802.11 (WLAN), Bluetooth, and Microwave Ovens in 2.4 GHz ISM-Band, http://www.baloerrach.de/stzedn/ (2004)
6. Howitt, I., Gutierrez, J.A.: IEEE 802.15.4 low rate - wireless personal area network coexistence issues, in Proc. of IEEE Wireless Communications and Networking Conference (WCNC), Vol. 3 (2003) 1481–1486
7. Golmie, N., Cypher, D., Rebala, O. : Performance Analysis of Low Rate Wireless Technologies for Medical Applications, will appear in Computer and Communication‘s special issue on WPANs (2004)
8. Zigbee Alliance: Zigbee specification v1.0, http://www.zigbee.org/ (2004)
9. Rappaport, T.S.: Wireless Communications, Prentice Hall (1996)
10. Ziemer, R. E., Peterson, R. L. Peterson, Borth, D. E.: Introduction to Spread Spectrum Communications, Prentice Hall (1995)
11. Lee, J.S., Miller, L.E.: CDMA Engineering Handbook, Artech House (1998)

On the Service Differentiation Capabilities of EY-NPMA and 802.11 DCF*

Orestis Tsigkas, Fotini-Niovi Pavlidou, and Gerasimos Dimitriadis

Department of Electrical and Computer Engineering,
Aristotle University of Thessaloniki
{torestis, niovi, gedimitr}@auth.gr

Abstract. Most existing WLAN access mechanisms cannot provide QoS assurances. Even those that are QoS aware can only provide relative service differentiation. Based on EY-NPMA, the HIPERLAN Medium Access Control algorithm, we propose a dynamic priority medium access scheme to provide time-bounded services. By approximating an ideal Earliest Deadline First (EDF) scheduler, the proposed scheme can offer delay and delay jitter assurances while achieving high medium utilization. Furthermore, we compare our scheme with a mechanism that enhances the IEEE 802.11 MAC protocol with QoS support. Simulation studies document and confirm the positive characteristics of the proposed mechanism.

1 Introduction

Holding the promise of making ubiquitous mobile access to IP-based applications and services a reality, wireless networks have gained popularity at an unprecedented rate over the last few years. Concurrent with the expansion of wireless networks is a high demand for real-time applications with very stringent and diverse QoS requirements. Providing QoS requires the network to guarantee hard bounds on a set of measurable prespecified attributes, such as delay, bandwidth, probability of packet loss, and delay variance (jitter). However, the unstable nature of WLANS and their different characteristics compared to those of their wired counterparts, have a direct impact on their ability to guarantee bounds on these QoS metrics.

When delay-sensitive traffic is to be supported by the network, the optimal choice is to use the Earliest Deadline First (EDF) service discipline [1]. The EDF scheduler is a dynamic priority scheduler where the priority of each packet is given by its arrival time plus the delay budget associated with the flow that the packet belongs to. The scheduler selects the packet with the smallest deadline for transmission on the link. It has been proven that for any packet arrival process

* This work was supported in part by the Hellenic State Scholarships Foundation (I.K.Y) and by the General Secreteriat of Research and Technology in the framework of the G.S.R.T. project: Irakleitos.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 289–299, 2005.

where a deadline can be associated with each packet, the EDF policy is optimal in terms of minimizing the maximum difference between the deadline of a packet and the time it is actually transmitted on the link [2].

EY-NPMA [7], the HIPERLAN MAC protocol, is a dynamic priority scheme, which provides hierarchical independence of performance by means of channel access priority. However, its ability to track an ideal EDF scheduler, and thus provide service differentiation degrades severely as traffic load increases and the number of contending nodes grows. This is mainly due to the fact that EY-NPMA supports only 5 priority levels. Based on EY-NPMA, we propose a dynamic priority Medium Access Control protocol to support time-bounded services in wireless networks. We modify the channel access scheme of EY-NPMA to support a high number of priority levels. Simulation results show that our scheme can approximate an ideal EDF scheduler to the largest possible extent while achieving high medium utilization. Furthermore, we compare our scheme with a mechanism that enhances the IEEE 802.11 MAC protocol with QoS support and assess the ability of each scheme to provide service differentiation while achieving high throughput.

The rest of the paper is structured as follows. In section 2 we provide an overview of distributed QoS capable medium access algorithms and we review a mechanism proposed to enhance the IEEE 802.11 MAC prorocol with QoS support. Section 3 reviews EY-NPMA, the MAC protocol for HIPERLAN. Section 4 presents the objectives and the design of the proposed protocol. Section 5 deals with the simulation results of the proposed schemes, while section 6 concludes the paper.

2 Related Work

In this section we review some of the existing approaches to provide service differentiation at the distributed wireless MAC layer. The common feature of these distributed medium access algorithms is their attempt to provide QoS support by implementing a priority scheduler, thus allowing faster access to the channel to traffic classes with higher priority.

[6][11][10] propose modifications to the IEEE 802.11 Distributed Coordinated Function (DCF) to incorporate differentiated service by supporting two or more priority levels. In [8] the authors propose a MAC protocol that provides multiple priority levels and adopts the black-burst mechanism [9] to guarantee that higher-priority packets will be always transmitted earlier than lower-priority packets. Packets with the same priority are then transmitted in a round robin manner.

All of the above schemes attempt to provide distributed service differentiation by assigning traffic to fixed priority classes. Better service differentiation can be provided if the priority level of packets contending for access to the wireless medium is updated in a dynamic manner. This allows packets with loose QoS requirements obtain better service than they would in a static priority scheduler without sacrificing the tight QoS guarantees that may be provided to other flows.

2.1 Priority Broadcast for DCF

In [3] a distributed priority scheme is proposed which takes advantage of the broadcast nature of the wireless medium to approximate an ideal deadline based schedule. It is shown that this scheme can achieve a closer approximation to an ideal deadline based schedule than IEEE 802.11.

The authors propose to piggyback the priority index of a Head-of-Line packet onto existing handshake messages of the 802.11 DCF. If the RTS suffers no collisions, then all nodes in the broadcast region hear the RTS and add an entry in their local scheduling table. When the receiving node grants a CTS, it also appends the priority in the CTS frame. This allows the hidden nodes which are unable to hear the RTS, to add an entry in their scheduling tables upon hearing the CTS. Upon the successful completion of the packet transmission, each node removes the current packet from its scheduling table. Moreover, when transmitting a packet, each node also piggybacks its HOL packet priority. The priority is also copied in the ACK frame to allow hidden terminals to hear the HOL priority index. Neighbors monitor these transmissions and add another entry in their scheduling table.

Nodes keep a table of these times in order to assess the relative priority of their own Head-of-Line packet. Specifically, given a node's j local scheduling table $\boldsymbol{S}_j$ and its rank r_j in its local scheduling table, the following equation is used in [3] to calculate the backoff interval,

$$f_l(\boldsymbol{S}_j) = \begin{cases} \text{Uniform}[0, CW_{min} - 1], & r_j = 1, \\ CW_{min} + \text{Uniform}[0, CW_{min} - 1], & r_j > 1. \end{cases} \quad (1)$$

The proposed backoff policy prevents nodes which are not ranked one in their scheduling table from contending in the first CW_{min} slots, thereby reducing contention for the top ranked nodes. The performance of the above policy improves when the scheduling table contains a higher fraction of the backlogged nodes' HOL indexes.

3 EY-NPMA

EY-NPMA, stands for Elimination-Yield Non-Pre-emptive Priority Multiple Access. Elimination-Yield describes the contention resolution scheme, while NPMA refers to the principle of the HIPERLAN medium access mechanism that provides hierarchical independence of performance by means of channel access priority. When a new packet arrives, its lifetime is set to a value that cannot exceed 500 ms. Depending on its residual lifetime, the packet is assigned one of the five priorities from 0 to 4, with 0 being the highest priority. Packets that cannot be delivered within the allocated lifetime are discarded. The synchronized channel access cycle comprises three phases: the prioritization, contention and transmission phase.

The prioritization phase ensures that only those data transmission attempts with the highest channel access priority will survive this phase. The contention

phase consists of two-subphases: elimination phase and yield phase. During the elimination phase a contending node transmits a channel access burst, whose length in slots is random between 1 and a predefined maximum, according to a truncated geometric distribution and then listens to the channel. If the channel is sensed as idle the node proceeds to the yield phase. During the yield phase, the contending nodes sense the channel for a random number of slots, and if the channel is sensed idle, they immediately enter the transmission phase by transmitting the packet stored in their buffer. All other stations sense the beginning of the transmission and refrain from transmitting. The parameters in the HIPERLAN standard were chosen so as to achieve a quasi-constant collision rate of 3.5% up to 256 simultaneous transmitting nodes. A performance study of EY-NPMA can be found in [12] and [13], where extended analytical and simulation results are presented. Further, it has been compared with DCF and EDCF in [4] and [5] respectively.

4 Proposed Protocol

Based on EY-NPMA, we propose a dynamic priority MAC protocol, DP-TB, to support time-bounded services in wireless networks. The proposed medium access scheme provides support to traffic with delay requirements by approximating an ideal EDF schedule to the largest extent possible. The proposed scheme preserves all three phases of the synchronized access cycle of the EY-NPMA scheme; yet, it features a different structure for the prioritization phase. Instead of a maximum of 5 prioritization slots, we propose a scheme that uses at most N slots for this phase. The prioritization phase, in the proposed DP-TB scheme, is further sub-divided in n sub-phases, where sub-phase j consists of at most p_j slots, such that $\sum_{i=1}^{n} p_i = N$. We do not fix N and n to constant values, but rather let them be parameters of the system. Depending on the choice of N and n there is a trade-off between the extent to which the ideal EDF scheduler can be approximated to and the throughput that can be achieved.

EY-NPMA uses N prioritization slots to support N priority levels. By subdividing the prioritization phase in n sub-phases we can provide a maximum of $P = \prod_{i=1}^{n} p_i$ priority levels, with 0 being the highest and $P - 1$ the lowest. The lifetime of a packet that has just arrived is set to a value that cannot exceed 500 ms. We divide the interval of 500 ms into P time intervals, each of which has a duration of $t_p = 0.5/P$ sec. Then the priority index PI of a packet with residual lifetime RL can be computed as:

$$PI = \{k : k \cdot t_p \leq RL < (k+1) \cdot t_p\} = \left\lfloor \frac{RL}{t_p} \right\rfloor \tag{2}$$

Given the priority index of a packet, the algorithm below can be used to determine how many slots a node should sense the channel in each sub-phase in order to determine if it has the highest priority packet for transmission.

$$
\begin{aligned}
&for(i = 1; i < n; i++) \\
&\qquad \{ps_i = \left\lfloor \frac{PI}{\prod_{j=i+1}^{n} p_j} \right\rfloor \\
&\qquad PI = PI - ps_i \cdot \prod_{j=i+1}^{n} p_j\} \\
&ps_n = PI
\end{aligned}
$$

As soon as the set of parameters $\{ps_1, ..., ps_n\}$ has been computed a packet can contend for channel access in the prioritization phase. The prioritization phase of DP-TB works as follows. At the beginning of the first sub-phase, a station that has a packet ready for transmission senses the channel for as many as ps_1 slots. If the channel is idle for the whole sensing interval, the station transmits a burst of one slot and proceeds to the second sub-phase. Otherwise, the station exits contention and will have another chance for accessing the channel at the next cycle. In the same manner, during the second sub-phase the station senses the channel for ps_2 slots, and if the channel is sensed idle it transmits a burst slot. The procedure is repeated until the last sub-phase, where the node transmits a burst of random length, instead of just one burst slot. The length of this burst is between 1 and a predefined maximum number of slots. The contention phase in DP-TB works as in EY-NPMA. However, during yield, a station, instead of randomly choosing an interval to backoff, will compute the duration of the backoff interval as:

$$\text{Backoff\_Interval} = \left\lfloor \frac{RL - PI \cdot t_p}{t_p} \cdot (m_y + 1) \right\rfloor \tag{3}$$

where m_y is the maximum number of slots that a station may backoff during the yield phase. This ensures that if there is a successful transmission, the station that transmits is the one with the lowest residual lifetime among those who survived the elimination phase. The proposed medium access protocol, allows us to define a large number of priority levels by using a relatively small number of prioritization slots. The added overhead of the prioritization phase in DP-TB can be alleviated by the lower collision rates.

5 Simulation Experiments

The experiments conducted in this work aim at comparing the performance of the proposed medium access schemes and not the respective implementations as expressed in the standards. Towards this end, the capacity of the common medium was set to 23.5 Mbps and was considered to be ideal, that is the only reason behind erroneous reception was packet collision. Furthermore, all network stations were within one hop from each other, eliminating thus the appearance of hidden/exposed terminals. The working parameters of the proposed MAC

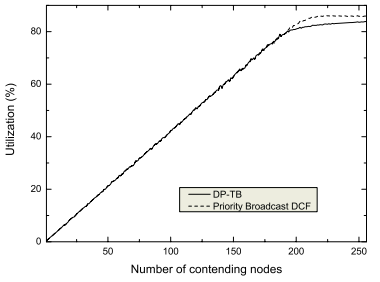
schemes were set to the values defined in their standards. Nodes generated traffic according to a Poisson process with mean rate 100 kbps. The performance metrics of interest were average medium utilization and probability of correct scheduling. The tool that was used for these experiments was customly coded by the authors in C++.

For DCF, a number of priority index assignment schemes is proposed in [3]. In our simulation experiments, we adopt the *Time To Live (TTL) allocation* scheme. In this scheme, a packet inserts its desired delay as its priority index. The authors in [3] propose the use of one byte to represent the priority tag. We rather assume that each packet is tagged with its delay budget which can be represented with infinite precision. DP-TB is designed to support 4840 priority levels. This is achieved by using at most 35 slots in the prioritization phase and subdividing it into 4 sub-phases, where the first two use at most 11 slots, the third 8 slots and the last one uses at most 5 slots. DP-TB uses at most 2 slots for the elimination and the yield phase.

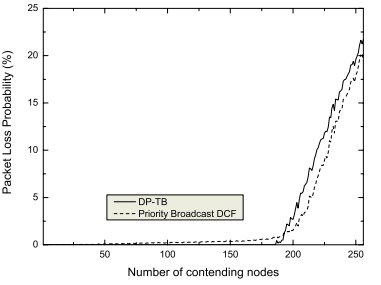
Two sets of experiments were performed. Each node is assumed to generate one flow. The performance metrics were examined for different node populations and packet sizes (128 and 2048 bytes). These sizes correspond to two extreme cases. For 2048 bytes packet size the factor that mainly affects the throughput of the protocol is the collision rate, while for 128 bytes packet size the dominant factor becomes the protocol's overhead.

In the first set of simulations, the delay requirements of flows are looser as they are distributed over a wider range. Each newly generated flow has a delay budget, which is uniformly distributed in the interval [0, 500ms]. Upon an arrival of a new packet the residual lifetime assigned to it is equal to the delay requirement of the flow that it belongs to. In Fig. 1(a) the mean medium utilization for 2048 bytes packet size is presented. For low and medium traffic load, DP-TB achieves slightly better medium utilization than Priority Broadcast DCF. Moreover, as the traffic load remains below 79%, DP-TB experiences no packet losses, while a small fraction of the contending packets are lost under Priority Broadcast DCF, as depicted in Fig. 1(b). Fig. 1(c) provides an explanation of the superiority of DP-TB under these traffic conditions. As it can be seen, DP-TB can approximate an ideal EDF scheduler to the largest extent possible. While traffic load is below 75%, DP-TB always makes the correct scheduling decision and schedules first the packets that their deadline is about to expire. The efficiency of DP-TB at approximating the ideal schedule is due to its ability to resolve the priorities of the contending packets. Fig. 1(d) shows that, under these traffic conditions, there are hardly any collisions, as the packet with the highest priority always captures the channel.

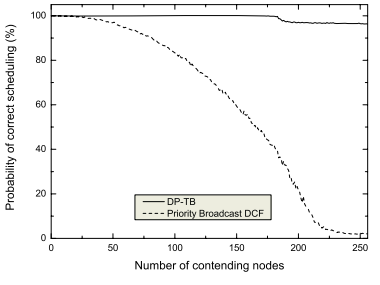
On the other hand, under Priority Broadcast DCF, the probability of correct scheduling decreases as the number of contending nodes increases. This results in a waste of resources (the delay budget assigned to each packet) as packets whose deadline is about to expire are pre-empted by packets that have enough delay budget to contend for channel access in forthcoming cycles. Moreover, even



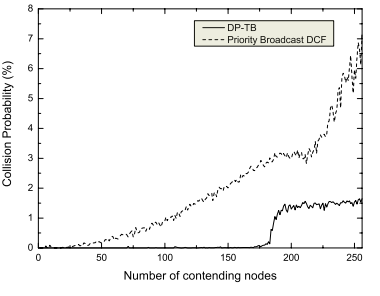
(a) Utilization vs. number of contending nodes



(b) Packet loss probability vs. number of contending nodes



(c) Probability of correct scheduling vs. number of contending nodes



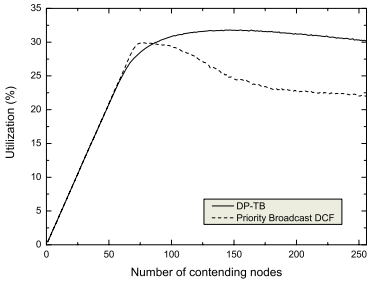
(d) Collision probability vs. number of contending nodes

Fig. 1. Performance metrics for 2048 bytes packet size

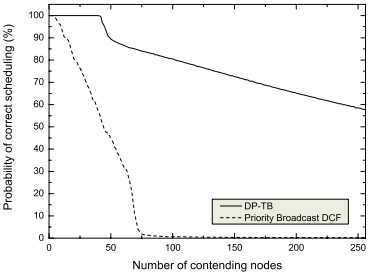
though Priority Broadcast DCF has lower overhead, it cannot achieve better medium utilization since it exhibits higher collision rate than DP-TB.

Under high traffic load, DP-TB makes the correct scheduling decision 96% of the time. However, the finite number of priority levels means that the probability of two packets having the same priority is nonzero, resulting in a slight increase in the collision rate. Priority Broadcast DCF suffers from a much higher collision rate; yet, the collisions are limited in the RTS/CTS exchange. Packet transmissions take place without collisions, allowing Priority Broadcast DCF to achieve slightly better medium utilization than DP-TB under high traffic load.

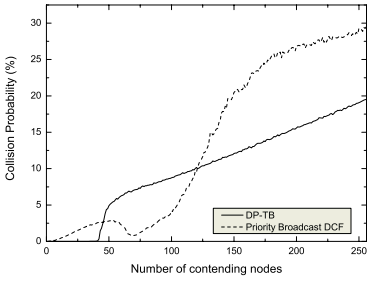
Fig. 2(a) shows the achieved medium utilization for 128 bytes packet length. DP-TB achieves higher throughput for a large number of contending nodes. However, it is outperformed by Priority Broadcast DCF when the number of contending nodes is in the range 60-85. In this range, Priority Broadcast DCF takes advantage of both its lower collision probability and its reduced overhead to achieve high medium utilization. The collision rate and the overhead of each protocol are depicted in Fig. 2(c) and Fig. 2(d) respectively.



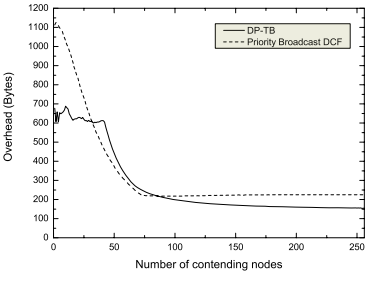
(a) Utilization vs. number of contending nodes



(b) Probability of correct scheduling vs. number of contending nodes



(c) Collision probability vs. number of contending nodes



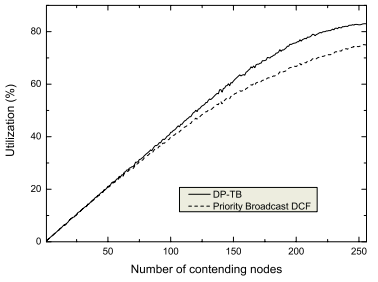
(d) Overhead vs. number of contending nodes

Fig. 2. Performance metrics for 128 bytes packet size

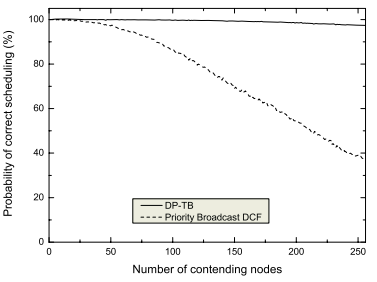
The efficiency of each proposed scheme at approximating an ideal EDF scheduler is depicted in Fig. 2(b). DP-TB achieves a closer approximation to the ideal deadline based schedule than Priority Broadcast DCF does. It is this inefficiency of Priority Broadcast DCF at making the correct scheduling decision that has an adverse impact on its throughput for a large number of contending nodes.

In the second set of experiments the QoS requirements are more stringent, since the delay budget of each contending flow is uniformly distributed in the interval [0, 10ms]. It should be noted that this is a more realistic scenario. Considering that wireless access is just another hop in a heterogeneous communication path that provides end-to-end delay guarantees, the delay budget of a flow at each node along the path will be small. The working parameters of each protocol were set to the values used in the first scenario.

Fig. 3-4 show that DP-TB outperforms Priority Broadcast DCF for any length of packet size. Not only does it achieve better medium utilization, but also it closely approximates the ideal deadline-based schedule. For 2048 bytes packet length, even when 256 nodes contend to gain channel access, the probability of

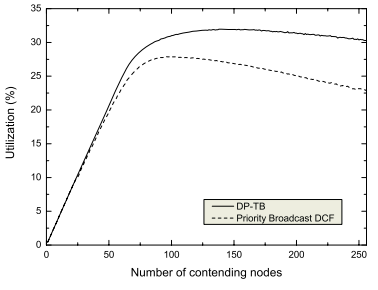


(a) Utilization vs. number of contending nodes

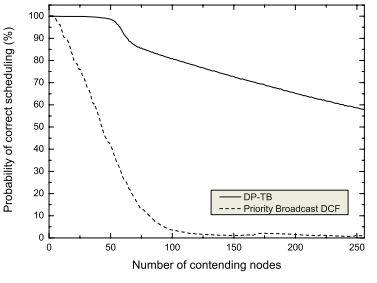


(b) Probability of correct scheduling vs. number of contending nodes

Fig. 3. Performance metrics for 2048 bytes packet size



(a) Utilization vs. number of contending nodes



(b) Probability of correct scheduling vs. number of contending nodes

Fig. 4. Performance metrics for 128 bytes packet size

correct scheduling for DP-TB is higher than 97%, while Priority Broadcast DCF starves to make the correct scheduling decision. The ability of DP-TB to track an ideal EDF scheduler, greatly depends on the number of the priority level it provides. The higher the number of the priority levels, the closer it can approximate the ideal schedule. By approximating an EDF scheduler to the largest possible extent, DP-TB minimizes the probability that a packet will be dropped due to lifetime expiration and, in this way the throughput of the protocol is increased. However, providing a very large number of priority levels is not beneficial. The efficiency of the protocol is adversely affected, since the introduced overhead increases as the number of priorities grows.

On the other hand, Priority Broadcast DCF attempts to approximate an ideal EDF scheduler by sharing information, which each node stores in its local scheduling table. It is evident that, the performance of the above policy improves

when the scheduling table contains a higher fraction of the backlogged nodes' HOL indexes. However, it may be the case that when a node starts transmitting a packet, it does not have another packet stored in its buffer. This will be common for low rate applications. Moreover, DCF allows a packet to gain channel access even when there is a packet with higher priority. In this scenario, where the QoS requirements are more stringent, the inefficiency of Priority Broadcast DCF at approximating an ideal EDF scheduler becomes evident.

6 Conclusions

In this work, we have proposed a distributed dynamic priority medium access scheme to support time-bounded services in wireless LANs. To better evaluate the performance of our scheme, we compared it with a mechanism that enables service differentiation in IEEE 802.11 DCF. The mechanisms behind our proposed protocol that allow it to achieve better performance rely on the approximation of an ideal EDF scheduler. By closely approximating an ideal EDF scheduler we minimize the maximum difference between the deadline of a packet and the time is actually transmitted on the wireless link. The good characteristics of the proposed scheme were confirmed via simulations, where significant gains in performance were witnessed.

References

1. D. Ferrari and D. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE Journal on Selected Areas in Communications, 8(3), pp. 368-379, Apr. 1990.
2. L. Georgiadis, R. Guerin, A. Parekh, Optimal Multiplexing on a single link: delay and buffer requirements, IEEE Transactions on Information Theory, 43(5), Sep. 1997
3. V. Kanodia, C. LI, A. Sabharwal, B. Sadeghi, and E. Knightly, Distributed Priority Scheduling and Medium Access in Ad Hoc Networks, Kluwer Wireless Networks, pp. 455-466, 2002.
4. G. Dimitriadis and F.-N. Pavlidou, Comparative performance evaluation of EDCF and EY-NPMA protocols, IEEE Comm. Lett., vol. 8, no. 1, pp. 42-44, Jan. 2004.
5. J. Weinmiller et al., Performance study of access control in wireless LANs - IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN, ACM/BALTZER Mobile Networks and Applications, vol. 2, no. 1, pp. 55-67, 1997.
6. A. Veres, A. T. Campbell, M. Barry, and L.-H. Sun, Supporting service differentiation in wirelees packet networks using distributed control, IEEE Journal on Selected Areas in Communications, 19(10), Oct. 2001.
7. ETSI, EN 300 652 v1.2.1: Broadband radio access networks : HIgh PERformance Local Area Network (HIPERLAN) Type I: Functional Specification, 1998.
8. J.-P. Sheu et al. A Priority MAC Protocol to Support Real-Time Traffic in AD Hoc Networks, Kluwer Wireless Networks 10, pp. 61-69, 2004.
9. J. L. Sobrinho, A. S. Krishnakumar, Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Networks, IEEE Journal on Selected Areas in Communications, 17(8), Aug. 1999.

10. D. J. Dneg and R. S. Chang, A priority Scheme for IEEE 802.11 DCF access method, IEICE Transactions on Communivation,, pp. 262-273, Jan. 1999.
11. I. Aad and C. Castellucia, Differentiation mechanisms for IEEE 802.11, IEEE INFOCOM, April 2001.
12. G. Dimitriadis and F.-N. Pavlidou, Two alternative schemes to EYNPMA for medium access in high bitrate wireless LANs, International Journal of Wireless Personal Communications, vol. 28, no. 2, pp. 121-142, Jan. 2004.
13. G. Anastasi et al., HIPERLAN/1 MAC protocol: Stability and performance analysis, IEEE J. on Selected Areas in Communications, vol. 18, no. 9, pp. 1787-1798, Sept. 2000.

Mitigating Interference Between IEEE 802.16 Systems Operating in License-Exempt Mode

Omar Ashagi, Seán Murphy, and Liam Murphy

Department of Computer Science,
University College Dublin,
Belfield Dublin 4, Ireland
{omar.ashagi, liam.murphy}@ucd.ie, sean.murphy@iname.com

Abstract. A rudimentary approach to mitigate interference issues in license-exempt 802.16 systems is presented. This approach operates by permitting each *Base Station* (BS), and associated *Subscriber Stations* (SSs) to remain inactive for a specified fraction of the time. Other systems can then transmit with a reduced likelihood of interference. A simulator was developed to determine how this system performs. The results show that the throughput of the system is very sensitive to the fraction of time each BS is active; the system throughput is maximised when each BS is active less than 40% of the time for the scenarios studied. The results demonstrate a discrepancy between uplink and downlink throughput which can be attributed to the greater amount of overheads in the uplink. Finally, the results show that broadcast information being transmitted periodically at full power has a significant detrimental impact on the system.

1 Introduction

Delivering broadband connectivity to every house and business is a challenging issue for broadband access providers. In many countries around the globe wired broadband connections like *Digital Subscribers Line* (DSL) or fibre-optics have been deployed, particularly in large urban areas. Wireless solutions can complement these wired technologies to deliver broadband access in less densely populated areas and developing regions.

While broadband access solutions have been available for some time now, standardised solutions have only recently become available and it is anticipated that they will have a significant impact on the market. IEEE 802.16 is a relatively new broadband wireless access standard but it is receiving considerable interest at present. It provides the setting for this work.

In general, the available wireless technologies can be divided into two mode of operation: licensed mode of operation, and license-exempt mode of operation. In licensed mode the spectrum is tightly controlled by a regulator, where licenses are issued to individual operators which provide exclusive access to some part of the frequency spectrum. By providing exclusive access to spectrum, this approach ensures that there is no interference between operators. In license-exempt mode

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 300–311, 2005.

spectrum is not assigned to any particular operator; the operators require no license to use this spectrum. However, some regulations may be applied for using these bands, such as limiting the transmit power and the coverage areas. The 2.4 GHz *Industrial, Scientific and Medical* (ISM) band and the *Unlicensed National Information Infrastructure* (U-NII) bands are examples of license-exempt bands.

As license-exempt spectrum is largely unregulated interference issues can arise. This interference can arise from: selfish use of the medium, the lack of cooperation between users, and the differences between systems characteristics and architectures. This interference affects the operation of wireless systems using license-exempt spectrum and can severely degrades their performance.

IEEE 802.16 has been designed such that it can operate in license-exempt spectrum. The IEEE 802.16 system consists of a *Base Station* (BS) and one or more *Subscriber Stations* (SSs) distributed over a geographical area with a radius of typically up to a few kilometres. In the case in which there are a limited number of channels available interference between different IEEE 802.16 systems can arise. In this paper we study the performance of a number of IEEE 802.16 systems operating in license-exempt mode of operation. More specifically, we wish to investigate the performance of an approach which can be used to mitigate the impact of such interference. This approach is based on introducing sleep intervals for each BSs. Initially, these sleep intervals are created randomly.

A rudimentary Java simulator was implemented to determine the system performance. The simulator was designed to simulate a number of 802.16 systems operating in license-exempt mode in the same geographical area on the same channel. Furthermore, the interference mitigation approach described below is simulated. The simulator can determine the amount of interference between these systems and system performance metrics such as the throughput per SS.

The reminder of the paper is organized as follows: in section 2 related work is discussed. Section 3 gives a brief introduction to the IEEE 802.16 standard. Section 4 describes the simulator and results obtained from using same. Finally, conclusions and future work is presented in section 5.

2 Related Work

There have been a few contributions to the literature on the performance evaluation of the 802.16 systems. The authors in [1], [2] investigated the performance of ETSI HiperMAN and IEEE 802.16a. Their results showed that the *Medium Access Control* (MAC) functions introduced an overhead of approximately 10%. Also, they have showed the efficiency gains that can be achieved by using the optional 802.16 packing and fragmentation features. In [6] the authors took a different perspective of 802.16 system performance and showed how different modulation and coding schemes have an impact on delay and throughput of the system.

In [7] an architecture for supporting *Quality of Service* (QaS) in 802.16 system was proposed. This architecture was based on priority scheduling and dynamic bandwidth allocation. No experiential work was presented to demonstrate the operation of this approach.

To the best of our knowledge no work has been published focusing on interference issues in IEEE 802.16 license-exempt systems. However, there have been contributions to the literature addressing interference problems for other radio systems, such as WLAN, HiperLAN/2 and Bluetooth. Several solutions have been proposed for such interference problems.

The authors in [3] proposed spectrum etiquette that requires a number of actions and rules, to facilitate the coexistence of wireless systems in unlicensed frequency bands. They examined this approach in three different radio systems, all of which support the *Listen Before Talk* (LBT) mechanism. Their results showed that using LBT mechanisms cooperation between systems can be achieved and interference significantly reduced. In [4] the authors looked into the interference problem between the IEEE 802.11a and the HiperLAN/2 systems which are operating in the 5.1 GHz band. They proposed a solution based on cooperation between these two systems and they concluded that this can be achieved by introducing minor changes to both standards.

While the above contributions are interesting and somewhat relevant in the context of this discussion, the architecture of the radio systems they have investigated differs from that of the IEEE 802.16. For this reason, the techniques that have been devised are not applicable here.

The IEEE 802.16 community is aware of the interference issue between the IEEE 802.16 license-exempt systems. For this reason, they have initiated a work activity focusing on mitigating interference in these systems. This work will lead to the development of the IEEE 802.16h standard. This work is still at an early stage, however; it is anticipated that the standard will be developed in the first quarter of 2007.

3 The IEEE 802.16 Standard

Broadband Wireless Access (BWA) technology has been around for a long time, but the lack of standards made the technology limited and expensive. The development of IEEE 802.16 standard is expected to result in significant changes to the costs of BWA systems due to economies of scale that can result from standardisation. This, in turn, is expected to stimulate significant growth in the BWA market.

The first 802.16 standard was published in April 2002. The standard defines the MAC and physical (PHY) layers, operating in licensed spectrum between 10 and 66 GHz. It requires *Line of Sight* (LOS) connectivity and supports up to 134 Mb/s of shared capacity per sector [5]. In April 2003, the IEEE 802.16a standard was published. It is an amendment to the IEEE 802.16 standard which provides additional PHYs for 2-11 GHz licensed and license-exempt operation and enhancements to the MAC to support a mesh topology. The standard sup-

ports *Non Line of Sight* (NLOS) connectivity and up to 70 Mb/s per sector. The IEEE 802.16-2004 [9] was released in October 2004. In essence, this integrates the original 802.16 standard and 802.16a amendment; it also provides some enhancements to improve the operation of indoor antennas in the 2-11 GHz band.

IEEE activity is still ongoing with much energy being devoted to adding mobility to 802.16 systems right now. The proposed IEEE 802.16e is an amendment to the standard which will provide support for mobility at vehicular speeds; it is due for completion in Summer 2005.

In the following subsection an overview of IEEE 802.16 is given. This is followed by a discussion of the 802.16 MAC which is particularly relevant to this work. This section ends with a short discussion of the standardised *Dynamic Frequency Selection* (DFS) mechanism which can be used to mitigate interference problems in some cases.

3.1 IEEE 802.16 System Overview

The IEEE 802.16 system consists of a BS and a number of SSs. It is a connection-oriented system with QoS support which is tightly controlled by the BS. The system typically operates in a point-to-multipoint fashion, although the standard does provide mesh network support.The uplink and downlink channels are *Time Division Duplexed* (TDD) or *Frequency Division Duplexed* (FDD). Four PHY layers are specified in the standard:

- WirelessMAN-SC: 10-66 GHz single carrier LOS required;
- WirelessMAN-SCa: 2-11GHz based on a single carrier, with NLOS support;
- WirelessMAN-OFDM: 2-11 GHz based on *Orthogonal Frequency Division Multiplexing* (OFDM) modulation, designed for NLOS operation [8];
- Wireless-MAN-OFDMA: 2-11 GHz based on OFDM modulation with support of *Orthogonal Frequency Division Multiple Access* (OFDMA) designed for NLOS operation.

The standard supports a number of modulation schemes such as QPSK, 16-QAM and 64-QAM.

For each PHY, a *physical slot* is defined. In the case of the OFDM PHYs, this corresponds to the transmission of a single OFDM symbol. A number of physical slots are grouped into so-called *mini-slots*: these are the smallest unit that can be used for resource allocation.

The standard makes specific stipulations regarding license-exempt operation. In license-exempt mode, TDD multiplexing should be used with a frame duration of 0.5ms, 1ms or 2ms [9] with a channel bandwidth of 20MHz.

3.2 MAC Overview

The IEEE 802.16 MAC is responsible for channel access and bandwidth allocation for different SSs. Medium access is controlled by the BS. In TDD operation, each frame consists of a *downlink* (DL) subframe and an *uplink* (UL) subframe as

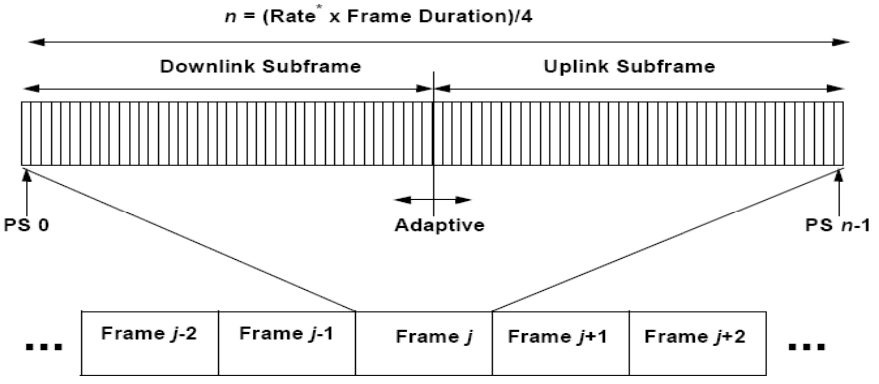


Fig. 1. OFDM Frame Structure with TDD

illustrated in figure 1. The downlink subframe is sent by the BS to SSs and consists of header information followed by data bursts transmitted to one or more SSs. More specifically, the DL *Protocol Data Unit* (PDU) contains a DL preamble used for synchronization, a *Frame Control Header* (FCH) and a number of DL data bursts. The first DL burst may contain information to be broadcast to all stations in the system such as a DL map, an UL map, a *Downlink Channel Descriptor* (DCD) and an *Uplink Channel Descriptor* (UCD). These messages are broadcast with full power to all of the SSs associated with a particular BS. The uplink consists of some time reserved for ranging and transmission of bandwidth requests, followed by the transmission of a number of UL PDUs by different SSs. Each UL PDU consists of a preamble and an UL burst.

IEEE 802.16 MAC supports various scheduling services classes for allocating bandwidth. These classes are defined as *Unsolicited Grant Service* (UGS), *Real-time Polling Service* (rtPS), *Non-real-time Polling Service* (nrtPS) and *Best Effort Service* (BES). There is also provision for an *Automatic Repeat Request* (ARQ) mechanism in the 802.16 MAC; this facilitates reliable data transfer.

3.3 DFS Overview

When using IEEE 802.16 in license-exempt operation, the U-NII license-exempt bands should be used. In these bands interference between users operating on the same channel can arise. The DFS mechanism was added to the 802.16 MAC to enable systems to avoid interference by automatically switching to an unused channel. The DFS mechanism is mandatory for license-exempt operation, where systems should detect and avoid primary channel users to avoid interference between them.

As illustrated in Figure 2(a) three channels are distributed between BSs A, B and C using DFS. In Figure 2(b) the number of BSs is increased to 5 resulting in more BSs than the available number of channels.In this case, DFS will be unable to distribute the available channels between the BSs. Thus, interference issues will arise unless some measures are taken to prevent them.

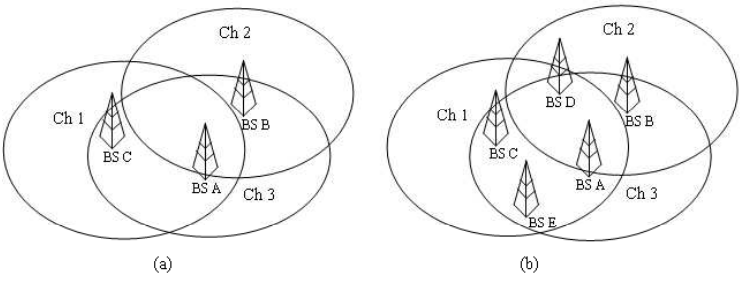


Fig. 2. Spectrum Distributions Using DFS

4 Simulation and Results

A simulator was developed throughout this work to model a number of different IEEE 802.16 systems operating in close proximity to each other on the same channel. In the following subsection an overview of the simulator is given followed by results and discussion subsection.

4.1 The IEEE 802.16 License-Exempt Mode Simulator

The simulator that was developed can model a number of different 802.16 systems operating in point-to-multipoint mode. Specific aspects of the 802.16 MAC and PHY are modelled such as scheduling, power control and interference are modelled in the simulator. To reduce the interference between the IEEE 802.16 systems, a rudimentary approach based on introducing random sleep times for each BS was used. The following assumptions were made while designing the simulator:

- The system is *symmetric* in the sense that the same amount of time is allocated to uplink transmission and downlink transmission;
- The system is *synchronised* in the sense that all BSs are assumed to operate from the same clock;
- The system is operating in saturated conditions, meaning that there is always data to be transmitted to and from each SS;
- All BSs use the same frame structure and operate on the same channel.

The scheduling mechanism is designed to distribute resources equally between the SSs in the system. In the simulator, it was assumed that each frame can accommodate a fixed number of SSs. The number of SSs that can be served for each of the permitted frame durations is shown in Table 1.

The TDD frame duration is divided between the downlink and the uplink subframes; the duration of the downlink preamble is 2 OFDM symbols and the FCH is transmitted using one OFDM symbol. In the uplink, the preamble is 1 OFDM symbol. There is a bandwidth contention period of 1 mini-slot and an initial ranging contention period of 4 mini-slots. The downlink and uplink

Table 1. Number of SSs per frame

Frame Duration	Downlink SSs	Uplink SSs
0.5ms	5	5
1ms	10	10
2ms	20	20

Table 2. Simulation Parameters

Parameter	Value
Wavelength	5.1238cm
Transmitter Gain	15dBm
Receiver Gain	15dBm
System Loss	1
Maximum Transmission Power	1mW
Bit Rate	46.08 Mb/s
OFDM Symbol Time	12.5μs
TTG and RTG	100μs
Mini-slot Duration	0.347μs
Channel BW	20MHz

preambles and the contention periods are called the TDD frame overhead; the parameters defined in Table 2 can be used to determine the amount of time consumed by this overhead. The downlink overhead consists of 3 OFDM symbols, 2 for preamble and 1 for FCH, where the uplink overhead consists of 2 contention periods of the same duration as 10 physical-slots, and 10 OFDM symbols divided between the SSs as uplink preambles.

The BS capability to become inactive for one or more frame durations is useful in the context of the license-exempt environment as it provides an opportunity for others on the same channel to transmit. If all BSs were to transmit continuously, then there could be very substantial interference for all users in the system, rendering it quite useless for all users. To avoid this, a probabilistic approach in which each BS remains inactive for some period of time is used. More specifically, each BS is configured with an *activity factor* which controls what fraction of the time the BS is active for. For each frame, the probability of the BS being active is equal to the activity factor.

The BSs schedule their traffic using a rudimentary scheduling approach, where the BS looks at how many SSs it has and what frame duration it uses. Then it finds the *Lowest Common Multiple* (LCM) between the number of SSs and the number of SSs per-frame according to Table 1. After that, the LCM is divided by the number of subscribers. The result is the number of different schedules required. The schedules are then constructed by placing the SSs consecutively into frames as shown in Figure 3.

The downlink and the uplink transmission powers are calculated at the system start-up using a power control algorithm, which is based on the well-known

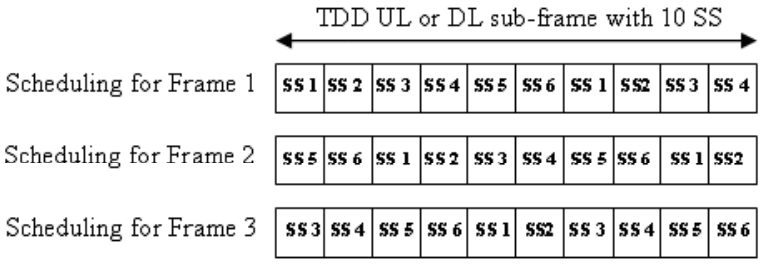


Fig. 3. Schedules for Six SSs

Friis power transmission equation. Determination of interference is also based on power calculations. This is done by comparing the receiver power of signals which are transmitted simultaneously. If the difference between these signals exceeds $> 5dB$, then the signal can be received correctly; otherwise interference is deemed to have taken place.In the simulations below, each BS is active of a defined period of time. A random process controls which specific frames a station is active for, So, while the fraction of time a BS is active is defined, the particular active and idle frames for a BS vary from simulation to simulations. It is worth nothing that an active time of 100% corresponds to the system being active all time; this is how the system would behave if there was no support for this sleep mode.

4.2 Results and Discussion

To study the performance of IEEE 802.16 licensed exempt systems, we used the simulation parameters specified in table 2. Random network topologies consisting of 4 BSs and 30 SSs per BS were used in these experiments. The network was located in a 100km^2 area as illustrated in Figure 4.

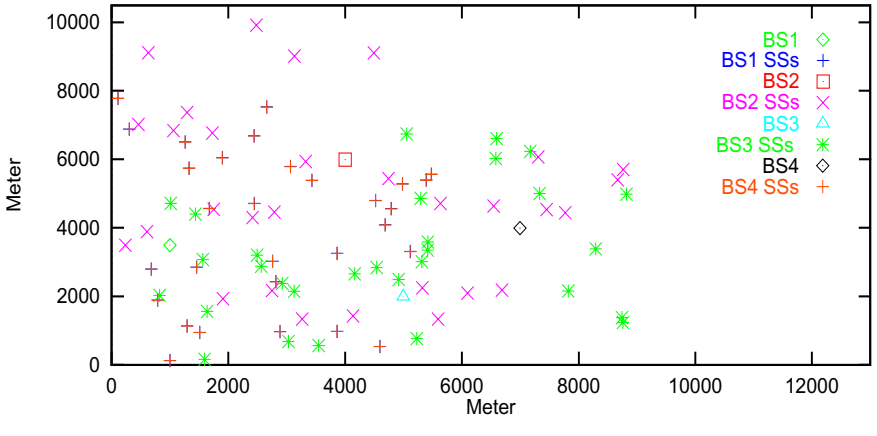


Fig. 4. SSs Throughput at 50% Active Time

Several simulations were performed each lasting 60 sec of simulation time. The results ware aggregated and analysed to provide graphical representation as shown in the following figures.

Figures 6 and 7 show the overall system throughput and the throughput per BS respectively. It can be seen from these figures that the performance of the system varies significantly with the activity factor. When the activity factor is low, the overall system throughput is quite low as each BS is inactive much of the time. When the activity factor is high, the overall throughput is also low, as there is much interference and few successful transmissions are made. There is a region around 25% to 40% during which time the throughput is maximised. At this point, every BS gets approximately 13% of the throughput that could be obtained if a BS had no interference issues. This can be compared to a scheme in which there is co-ordination between the BSs and the time could be divided such that each BS is active for 25% of the time. In this case, each BS could obtain 25% of that which it would obtain if it had exclusive access to the medium.

In Figure 8, the variation in the SSs throughput curve with the activity factor can be seen. This graph exhibits similar behaviour to that of the previous graphs - the throughput is low for low and high activity factors, and is highest for some intermediate values. It can also be seen from this graph that there is a considerable variation in the throughput achieved by each SS, as evident from the significant difference between the minimum and maximum. Figure 9 shows that the average system downlink and uplink throughput coincides with figure 6 and figure 7 and also shows the expected difference between the downlink and the uplink throughput due to the uplink overheads.

Figures 10 and 11 show the overall numbers of collisions on the system. From these results, it can be seen that there is a very linear relationship between the activity factor and the number of collisions in the system. Further, it can be seen that the collisions are divided pretty equally between all the BSs in the system. The numbers of collisions experienced by the SSs is shown in Figure 12. As with the previous graph, there is a quite linear increase in the mean number SS collisions with the activity factor. Also, as with the SS throughput, there is a significant variance in the amount of collisions that can be experienced by a SS. Figure 13 shows the average uplink and downlink collision rate. It is clear from the figure that most of the collisions occur in the downlink. It is worth noting, however, that in many cases a collision in the downlink can result in a SS missing an opportunity to transmit: if the SS does not receive the UL Map correctly, it does not know when to transmit and hence misses a transmission opportunity. The much greater number of collisions in the downlink can be attributed to the fact that some of the downlink information is transmitted at full power.

In Figure 14 the nodes in the system have been classified into those that obtain high throughput, medium throughput and low throughput. The results depicted in Figure 14 were generated using an activity factor of 50%. This classification is performed based on the difference from the overall mean throughput: nodes that obtain throughput of less than 50% of the mean throughput are deemed low throughput and nodes that obtain throughput of 50% greater than

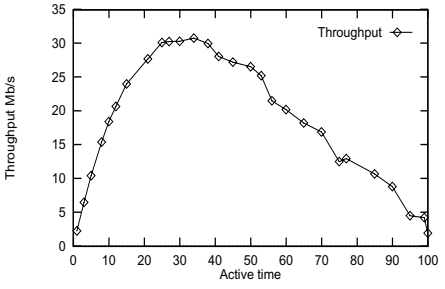


Fig. 6. Overall System Throughput

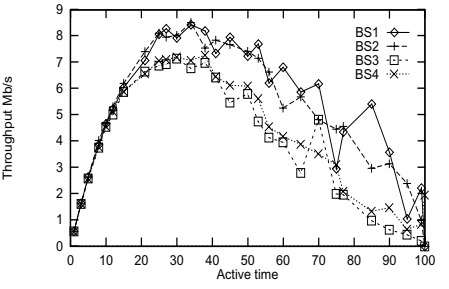


Fig. 7. Throughput per BS

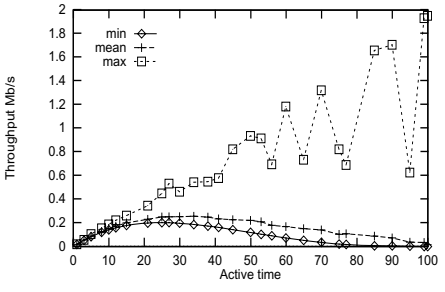


Fig. 8. Min, Ave, Max Throughput

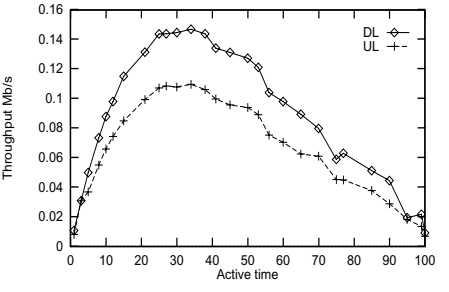


Fig. 9. Average DL and UL Throughput

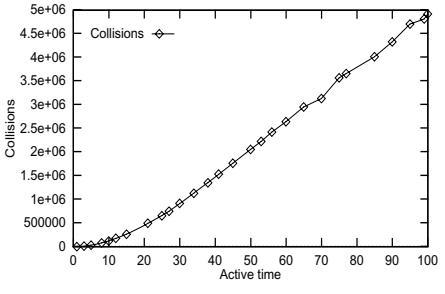


Fig. 10. Overall System Collisions

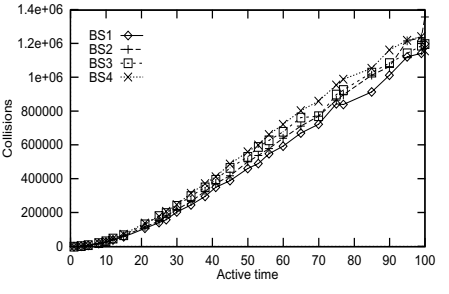


Fig. 11. Collision per BS

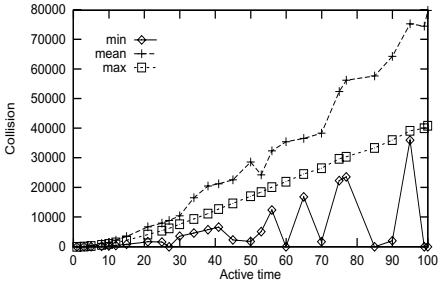


Fig. 12. Min, Ave, Max Collision

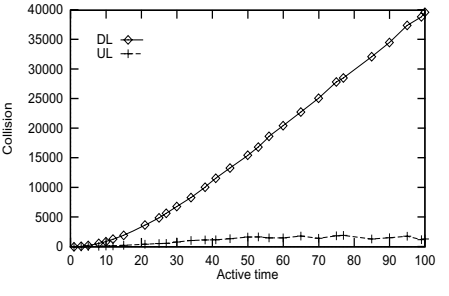


Fig. 13. Average DL and UL Collision

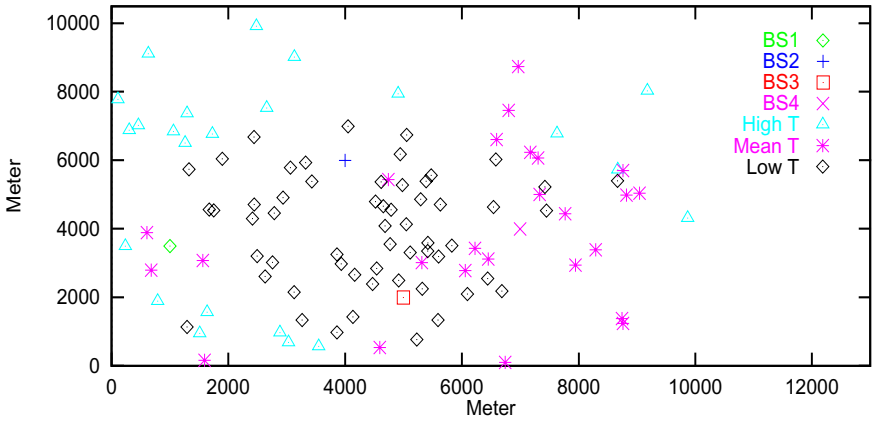


Fig. 14. SSs Throughput at 50% Active Time

the mean are considered high throughput. It is clear from the results that the nodes which are located in the centre of the area obtain lower throughput and those at the extremities obtain significantly higher throughput. This is not surprising as those at the centre are more likely to experience interference.

5 Conclusion

This work is an initial study of an approach to mitigate interferences issues that may arise in 802.16 systems operating in license-exempt operation. The approach is a natural one; it operates by enabling some BSs (and their associated SSs) to remain inactive, or asleep, for some periods of time, thereby permitting others in the same geographical area to use the limited available spectrum.

In this contribution, a rudimentary simulator which we have developed to simulate this scenario has been described. The simulator can be used to simulate a number of BSs and their associated SSs operating in the same geographical area. The simulator identifies when simultaneous transmissions from different entities in the system results in interference rendering the transmissions useless. These transmissions are dubbed collisions here.

The results show that the throughput of the system is greatly increased by limiting the amount of time a BS is active. For the case studied, the best system performance is obtained when each BS is active for quite a small fraction of the time ($< 40\%$). Another interesting finding in this work is that there is a significant discrepancy between the uplink and downlink performance: the downlink delivers better throughput due to the greater amount of overhead introduced by uplink overheads.

One issue that had a significant impact on the system performance was that of transmission of the broadcast information on the downlink. As all the BSs were synchronised and this information is transmitted at full power at the same

point in a frame, this was frequently the cause of collisions. One way to mitigate this may be to consider how the system performs in asynchronous operation.

Acknowledgments

The support of the Informatics Research initiative of Enterprise Ireland is gratefully acknowledged.

References

1. C. Hoymann, M. Putter, I. Forkel: Initial Performance Evaluation and Analysis of the global OFDM Metropolitan Area Network Standard IEEE 802.16a/ETSIHiperMAN. In Proc. European Wireless conf (2004)
2. C. Hoymann, M. Putter, I. Forkel: The HIPERMAN StandardPerformance and Analysis. In IST. Mobile and Wireless Communication Summit (2003)
3. S. Mangold, K. Challapali: Coexistence of Wireless Networks in Unlicensed Frequency Bands. Submitted to Wireless World Research Forum No.9. (2003)
4. S. Mangold, J. Habetha, S. Choi, C. Ngo: Co-existence and Inteworking of IEEE802.11a and ETSI BRAN HiperLAN/2 in MultiHop Scenario. In Proc. IEEE Workshop on Wireless Local Area Network (2001)
5. R. Marks, K. Stanwood, S. Wang: IEEE 802.16 Standard: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access. IEEE Communication Magazine (2002)
6. S. Ramachandran and C. Bostian, S. Midkiff: Performance Evaluation of IEEE 802.16 for Broadband Wireless Access. In Proc. OPENETWORK (2002)
7. G. Chu, D. Wang, S. Mei: A QoS Architecture for the MAC Protocol of IEEE 802.16 system. In Proc.In Proc. IEEE Circuits and Systems and West Sino Expositions Int. Conf (2002)
8. I. Koffman, V. Roman: Broadband Wireless Access Solution Based on OFDM Access in IEEE 802.16. IEEE Communication Magazine (2002)
9. IEEE 802.16-2004, IEEE Standard For Local and Metropolitan Area networks-Part 16: Air Interface for Fixed Broadband Wireless Access. (2004)

ECN Marking for Congestion Control in Multihop Wireless Networks

Vasilios A. Siris and Despina Triantafyllidou*

Institute of Computer Science (ICS),
Foundation for Research and Technology – Hellas (FORTH),
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
dtriant@ics.forth.gr

Abstract. In this paper we propose an approach to increase TCP's fairness in multihop wireless networks, using ECN as a congestion signalling mechanism. The novel idea we introduce is that the marking probability at a wireless node is a function of the aggregate utilization in the node's neighborhood, which is determined by the sum of the receiving rates of all nodes within its collision domain. A node's received rate can be communicated to neighboring nodes by piggy-backing it on control packets, such as CTS and RTS messages, or data packets. Simulation results demonstrate that our approach can improve TCP's fairness in a multihop wireless network compared to drop tail queueing, while achieving the same aggregate throughput. Moreover, the proposed approach yields smaller average packet delay and delay jitter compared to drop tail queueing.

1 Introduction

The efficient control and management of wireless network resources has been an area of active research. Enhancing TCP performance over a wireless ad hoc network is a challenge, due to the constraints posed by the limited capacity of the wireless spectrum and the medium access contention among the wireless nodes. On the other hand, the number of users accessing the Internet through IEEE 802.11 wireless links is growing dramatically. The above motivate the need for efficient and fair congestion control over such networks.

Multihop environments pose three specific issues related to TCP's performance. First, changing the position of nodes causes path interruptions, resulting in repeated TCP timeouts. Efficient retransmission techniques to overcome this problem have been proposed. The second problem has to do with the fact that while there exists a TCP window size for which a highest throughput is achieved, TCP tends to operate with a higher window size, leading to packet losses and reduced throughput [4]. The third problem is the unfairness introduced by such networks [6].

* Authors are also with the Dept. of Computer Science, University of Crete.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 312–322, 2005.

In this paper we focus on the unfairness problem, and propose an Explicit Congestion Notification (ECN) marking approach that correctly accounts for the level of congestion within different collision domains of a multihop network. The approach combines two key ideas: first, it uses ECN as an end-to-end signalling mechanism for conveying congestion information across a multihop wireless network; second, marking at a wireless node is performed using a modification of the load-based marking (LBM) algorithm proposed in [12]. This modification suggests that the marking probability is a function of the aggregate utilization within the node's collision domain.

Although the application of ECN to wireless networks is not new, e.g. see [8, 9], its application as a common signalling mechanism for conveying congestion information in wired and wireless networks, in a way that takes into account the particular characteristics of the underlying wireless technology was first proposed in [11], for the case of 3G networks based on Wideband CDMA. However, IEEE 802.11 networks differ from 3G WCDMA based cellular networks, hence the marking procedure for each should be different. In the case of a single hop wireless network, as discussed in [11], the marking probability for the wireless channel, which is a single shared resource, can be taken to be a function of the level of congestion, expressed by the aggregate utilization measured at the access point, which includes both the uplink and the downlink transmission throughput.

In the case of a multihop wireless network, unlike a single hop network, there is no access point acting as a gateway for all the packets travelling in the wireless network. Moreover, due to attenuation in the wireless medium, contention is location-dependent, hence the total utilization can no longer accurately reflect the level of congestion in the wireless network. For this reason we consider the utilization within a collision domain, which is given by the sum of the receiving rates of all nodes within the collision domain; a node's receiving rate can be communicated to all its neighbors within its collision domain by piggy-backing this information on control packets, such as CTS and RTS messages, or data packets. Such a scheme has the goal to broadcast location-dependent congestion information within a single collision domain, which is analogous to the use of the RTS/CTS mechanism for informing nodes of the presence of other neighboring nodes with which their transmission can collide. Indeed, as the experimental results demonstrate, our approach can address the unfairness that is observed in hidden and exposed terminal topologies. Each node, based on the sum of the received rates of all neighboring nodes within its collision domain, computes a marking probability using a load-based marking (LBM) algorithm [12]. Hence, our proposal combines this local (collision domain specific) broadcasting of congestion information with ECN marking, which provides the end-to-end communication and accumulation of congestion information.

The rest of the paper is structured as follows. In Section 2 we present our approach for load-based marking over multihop wireless networks. In Section 3 we present and discuss simulation results demonstrating that the proposed approach can improve TCP's fairness over a multihop wireless network. In Section 4 we

present a brief overview of related work, identifying where it differs from the work presented in this paper. Finally, in Section 5 we conclude the paper, identifying related ongoing and future work.

2 Congestion Signaling in a Multihop Wireless Network

In this section we describe our approach for measuring the load and signaling congestion in a multihop wireless network. The approach is motivated by the observation that in such networks the level of congestion is no longer an individual characteristic of a single resource, but a shared feature of all the nodes within the same collision domain. Each node, depending on the packets it sends, receives or senses within its range, can obtain a different view of the level of congestion in its surrounding area. In fact, it itself also contributes to congestion. The above location-dependent information must be taken into consideration, in order to effectively and fairly control the usage of the wireless channel by all nodes. Given the congestion information from all its neighbors, a node determines the probability with which it will mark packets at its output queue. Then, based on the aggregate congestion along the whole path its packets traverse, which is signaled using ECN marks, the node responds by adjusting its transmission window according to the TCP congestion control protocol.

According to the approach proposed in this paper, the first key idea to achieve the above goal is to implement ECN as the end-to-end signalling mechanism for communicating congestion information within the multihop wireless network. Secondly, marking is performed at each node using a load-based marking (LBM) algorithm [12], modified to consider the appropriate measure of the level of congestion in a multihop wireless network.

2.1 ECN as the End-to-End Signalling Mechanism

Explicit Congestion Notification (ECN) has been approved as an IETF proposed standard [10]. With ECN, congestion of a network link is explicitly signaled by having routers set the CE (Congestion Experienced) bit located in the IP header, rather than implicitly signaled through lost packets as is the case with TCP's current operation. ECN can thus provide an early warning of incipient congestion, before packets start to be dropped, thus avoiding their retransmission. Hence, ECN can, to a large extent, avoid packet drops and the corresponding overhead of retransmitting lost packets.

ECN is appropriate for wireless networks, since in wireless networks non-congestion related losses due to wireless channel corruption can occur with a non-negligible probability. However, ECN alone cannot solve the problem of TCP decreasing its throughput in the case of non-congestion related losses. To address this either TCP's reaction to losses must be modified, such as by identifying and differentiating between congestion and non-congestion related losses, or link-layer mechanisms, such as forward error correction and retransmission over the wireless link, should hide losses due to corruption from TCP.

Our proposal for using ECN goes one step further from addressing the issue of congestion and non-congestion related losses, which we assume are handled by IEEE 802.11 MAC's link-layer retransmission mechanism. Our approach proposes to use ECN to convey congestion information across a multihop wireless network. For wired networks, marking is performed at the output link of routers. On the other hand, for a multihop wireless network we propose to implement marking at the output queue of every wireless node, considering the aggregate utilization within the nodes collision domain. In the following section we describe how this aggregate utilization can be measured in a distributed fashion.

2.2 Load-Based Marking (LBM) in a Multihop Environment

In a multihop network, there is no pre-defined link between any two nodes. Instead, nodes share the wireless channel and compete for it in a distributed manner using the MAC protocol. Thus, congestion cannot be tracked to a specific node, but to the entire area around it. Hence, the aggregate utilization of this area should be taken as an indication of the level of congestion of the wireless resource. Moreover, since there is no single shared buffer that is used for the packets flowing in the same area, a RED (Random Early Detection)-like marking algorithm, where the packet marking probability is a function of an average queue length, cannot be applied.

Apparently, for each node we need to define this area over which the aggregate utilization should be estimated. Interference with a node may be caused both by one-hop neighbors and by two-hop neighbors. However, the set of nodes that interfere with each other also depends on the traffic destination. For example, consider the multihop network shown in Figure 1, where the transmission range for each node extends up to its immediate neighbor. Assume the case where node 2 transmits to node 1, and node 3 transmits to node 4. Even though 2 and 3 are immediate neighbors, they can both simultaneously perform their transmissions, since each is two hops away from the other's destination (this is the so-called exposed terminal scenario). Similarly, which two hop neighbors interfere also depends on the destination. Hence, if both nodes 2 and 4 transmit to 3, then they are in the same collision domain (this is the so-called hidden terminal scenario). On the other hand, if 2 transmits to 1 and 4 transmits to some other node on its right (not shown in the figure), then nodes 2 and 4 are not in the same collision domain.

Next we discuss how a node can compute the aggregate utilization within its collision domain. First note that summing the transmission rates of neighboring

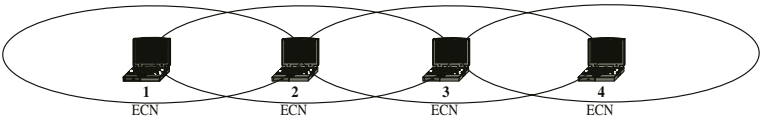


Fig. 1. All nodes apply ECN marking based on the congestion within their collision domain

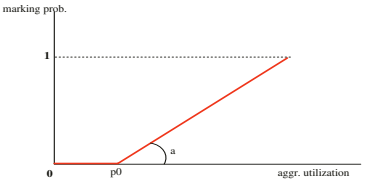


Fig. 2. LBM has three parameters: time interval t_{avg} over which aggregate utilization is measured, minimum utilization threshold ρ_0, and slope parameter α. Marking probability is a piecewise linear function of the average utilization

nodes is not correct, since in the scenario mentioned above where node 2 transmits to node 1, and node 3 transmits to node 4 (exposed terminal), because 2 and 3 do not interfere, node 2 should not consider node 3's transmission rate. Moreover, considering the transmission rate wouldn't account for contention due to two hop neighbors, for example in the case both nodes 2 and 4 transmit to node 3. Summing the utilization measured by neighboring nodes is for the same reason incorrect; indeed, such an approach has another subtle problem since it can result in considering some transmissions twice (once through the transmission rate and once through the receiving rate). Another option is to sum the received rates of all neighbors. In the case where the transmission range is the same for all nodes, and the transmission range is the same with the interference range, then this approach would indeed consider the received rates of only those neighbors that a node's transmission can reach, hence can interfere with other nodes transmitting to the same destination.

Based on the previous discussion, we propose that the marking probability is a function of the aggregate utilization over some time interval t_{avg}, measured by dividing the sum of the received throughput of all nodes within the same collision domain, with the wireless channel capacity. Because we are using TCP, t_{avg} should be at least a few RTTs, to allow the system to obtain an accurate estimate of the average rate. Also, it should be such that the system can track traffic changes, e.g. of number of users, therefore not too large.

The marking probability can have a piecewise linear dependence on the aggregate utilization, as illustrated in Figure 2. The marking probability is zero when the average utilization is less than ρ_0. For utilization values ρ larger than ρ_0, the marking probability is given by $\min\{\alpha(\rho - \rho_0), 1\}$. Hence, the extended load-based marking algorithm for multihop environments has the same three LBM parameters: the time interval t_{avg} over which the aggregate utilization is measured, the minimum utilization threshold ρ_0, and the slope parameter α.

3 Simulations, Results and Discussion

In this section we present and discuss the simulation results comparing the proposed marking approach with drop tail queueing. The results show that the

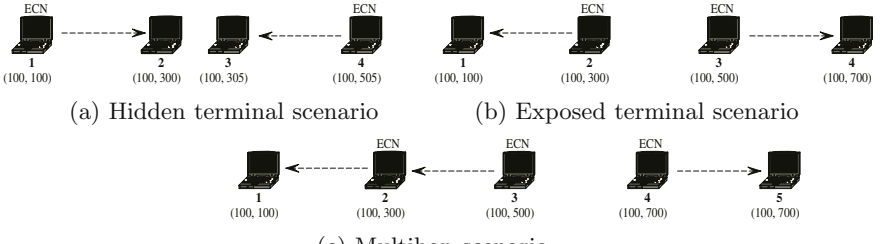
proposed approach can achieve higher fairness compared to drop tail queueing, while achieving the same utilization; as utilization we consider the ratio of the throughput -including the packet header overhead- and the wireless capacity, which is 11 Mbps. Furthermore, we show that the proposed approach results in smaller packet delay and delay jitter.

3.1 Experiment Scenarios

Experiments were conducted using the NS-2 simulator, with the topologies shown in Figure 3. The transmission range equals interference range, both being 250 m. The scenario in Figure 3(a) implements the hidden terminal case. Traffic flows from node 1 to node 2, and from node 4 to node 3. Marking is performed at the senders, i.e. nodes 1 and 4, and the marking probability at both of these nodes considers the sum of the receiving rate at nodes 2 and 3. The scenario in Figure 3(b) implements the exposed terminal case, with traffic flowing from node 2 to node 1, and from node 3 to node 4. In this case, node 2 calculates a marking probability based on the receiving rate of node 1, whereas node 3 calculates a marking probability based on the receiving rate of node 4. The scenario in Figure 3(c) is a multihop scenario. Traffic flows from node 3 to node 1 through node 2, and from node 4 to node 5. Each of the three sending nodes, i.e. nodes 3,2, and 4 performs marking based on the received rate of its one-hop receiver.

Ftp flows transfer files whose sizes follow a pareto distribution of average 500 KBytes and 5 MBytes, and shape parameter 1.5. The throughput for each flow is given by the ratio of the total received traffic -data and overhead- and the duration of the file transfer, taken as the interval between the time the first SYN packet is sent, and the time the last ACK is received. Each flow's start-time was randomly selected from the interval $[0, 5]$ seconds. We used TCP Reno. In all the experiments the measurement interval parameter of the LBM scheme is 500 ms. On the other hand, we consider different values for the minimum utilization threshold and the slope parameter.

The IEEE 802.11 MAC layer performs retransmissions of corrupted packets. To model the bursty (time correlated) errors in the channel, we consider a multi-state error model similar to that presented in [2], consisting of a three-state



(a) Hidden terminal scenario (b) Exposed terminal scenario

(c) Multihop scenario

Fig. 3. Simulation scenarios

Table 1. State transition matrix

Current state	Next state		
	G	B	VB
G	0.95	0.02	0.03
B	0.10	0.20	0.70
VB	0.20	0.03	0.77

discrete-time Markov chain. The time slot is equal to the slot of the 802.11b MAC sublayer, i.e. $20\,\mu s$. In the good state (G), transmission is error free. In state B and VB errors occur uniformly with probability p, and $5.5 \times p$, respectively. The probability of state transition is shown in Table 1. In our experiments we consider the case of error free transmission, and the case where errors occur according to the above model with an average error probability 1%, achieved for $p = 0.01$.

3.2 Fairness and Throughput

The graphs we present next show the average fairness, and the corresponding 95% confidence interval from 15 independent runs of the same experiment. As a measure of fairness we consider the widely used metric given in [3]:

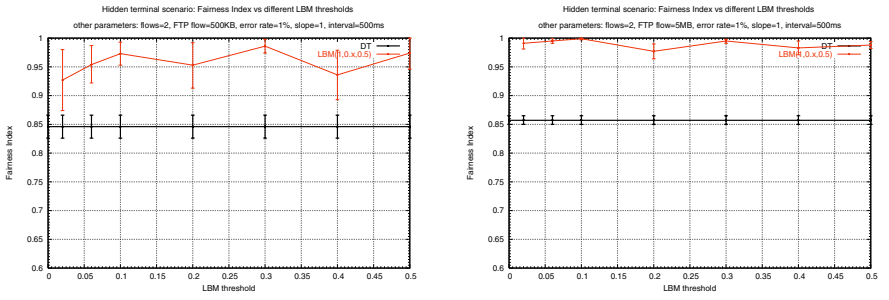
$$\text{Fairness Index} = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2} \ ,$$

where x_i is the rate of flow i and N the number of flows. The fairness index takes values in the interval (0,1], with a higher value indicating higher fairness.

Figures 4(a) and 4(b), for the hidden terminal scenario of Figure 3(a), show that the proposed LBM scheme achieves higher fairness compared to drop tail (DT) queueing, for both short and long ftp flows, and for a range of values of the LBM minimum utilization threshold ρ_0. Figure 5(a), which is for the exposed terminal scenario of Figure 3(b), also shows that fairness improves. Observe that with short ftp flows, the fairness improvement appears to be smaller for smaller values of the threshold parameter; this can be attributed to the fact that very small thresholds correspond to very early, and thus, inefficient marking.

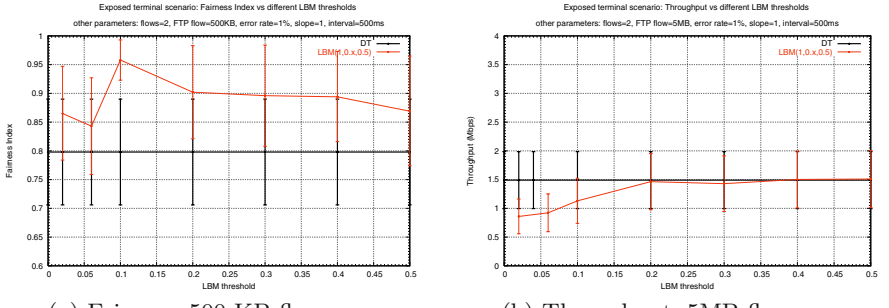
Comparison of Figure 4(a) with 4(b) shows that the improvements are larger for longer ftp flows. This is also the case in the exposed terminal scenario. Results also show that for long ftp flows the improvement is larger in the exposed terminal scenario; on the other hand, the improvement is similar in both scenarios, in the case of small ftp flows. To illustrate what the different values of fairness imply, note that a fairness index equal to 0.8 corresponds to the case where the two flows achieve a throughput of 2.09 Mbps and 0.75 Mbps. On the other hand, a fairness index equal to 0.97, corresponds to the case where the two flows achieve a throughput of 1.40 Mbps and 1.87 Mbps. For an error free channel, fairness is also improved with the proposed approach, compared to DT.

In all the above experiments, the aggregate throughput achieved by both the LBM scheme and drop tail queueing is the same, as shown in Figure 5(b); observe



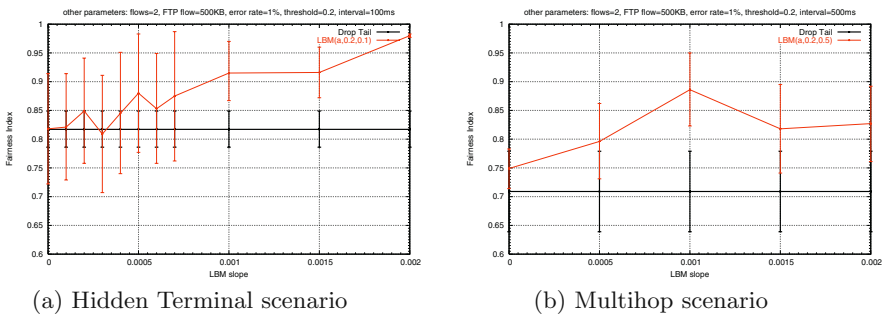
(a) 500 KB flows (b) 5 MB flows

Fig. 4. Fairness for the hidden terminal scenario, for different LBM minimum utilization thresholds ρ_0 and average error probability 1%



(a) Fairness, 500 KB flows (b) Throughput, 5MB flows

Fig. 5. Throughput and fairness for the exposed terminal scenario, and different LBM minimum utilization thresholds ρ_0, and average error probability 1%



(a) Hidden Terminal scenario (b) Multihop scenario

Fig. 6. Fairness for different LBM slope parameters, and for 500 KB flows and average error probability 1%

that the throughput decreases for small values of the utilization threshold, which is also the case with the fairness index, and which corresponds to a rather early marking. Hence, improved fairness is not achieved at the expense of decreased throughput, as is the case with the NRED scheme proposed in [6]. Conclusions are identical for the hidden terminal scenario. In Figures 6(a) and 6(b), LBM shows higher fairness compared to DT, for different slope parameters. Observe that the improvement decreases for smaller values of the slope parameter. This can be explained by the fact that a smaller slope yields a less aggressive marking.

3.3 Packet Delay and Delay Jitter

Table 2 shows the average and standard deviation of the packet delay over the wireless link. The packet delay is given from the time the packet is enqueued in the output queue of the TCP sender, until the corresponding ACK is received. Thus, it also includes queuing delay. Observe that LBM marking scheme achieves a smaller average delay and delay jitter compared to drop tail queueing, as indicated by the smaller values of the standard deviation.

Table 2. Average and standard deviation of packet delay over the wireless link. File size=1 MByte, loss prob.=1%, LBM: $\alpha = 1$, $\rho_0 = 0.1$, $t_{avg} = 500$ ms

Terminal scenario	Flow	DT		LBM	
		avg. delay(ms)	std.dev.	avg. delay(ms)	std.dev.
Hidden	1	97.6	77.7	13.77	13.1
Hidden	2	60.9	57.7	9.24	26.1
Exposed	1	41.3	26.5	22	19.8
Exposed	2	49.6	65.9	13.9	37.9

4 Related Work

In this section we present a brief overview of the related work. Schemes considering ECN marking fall within the end-to-end approaches for improving TCP's performance over wireless networks, and have been proposed for wireless networks in [8, 9, 5]. The new contribution of this paper is to propose an approach for computing the ECN marking probability that takes into account the location-dependent nature of contention in multihop wireless networks.

Other approaches consider using a RED-like mechanism [9] or some other level of congestion, such as a congestion price [1]; however, as we argue in this paper, a shared buffer does not exist in a wireless network, hence such marking algorithms do not reflect the underlying resource sharing model. The work in [6] explores the relationship between TCP unfairness and early network congestion in wireless multihop networks, suggesting a scheme called Neighborhood-RED (NRED) for marking, based on the aggregate (incoming and outgoing) queue size of a node's neighborhood. The underlying idea of this approach is that

improvement of spatial reuse can lead to better fairness. However, in this work fairness is achieved at the expense of decreased throughput; this is not the case with the approach proposed in this paper as our simulation results demonstrate.

The work in [4] considers a packet drop scheme (which the authors note can also be adapted to a mark probability scheme), where the drop probability is a piecewise linear function of the number of MAC layer retransmissions, which can signal network overload, when they exceed some minimum number. Such a scheme, called Link-RED, was shown to increase the performance in the case of multi-hop wireless networks. Finally, the work in [7] considers the general framework for congestion control schemes using a utility-based modelling approach. The proposed marking scheme is a concave function of the traffic arriving at a link, when this rate is larger than some minimum capacity value. Our approach differs in that the marking probability is a linear function of the aggregate utilization in a collision domain.

5 Conclusions

The main contribution of this paper is to use the aggregate receiving rate within a node's collision domain to compute an ECN marking probability. Simulations show that the proposed approach, operating with TCP congestion control, can achieve higher fairness compared to drop tail queuing, while achieving the same aggregate throughput. The approach, also, yields smaller packet delay and delay jitter.

We are currently performing experiments with more complex topologies, to investigate the influence of the LBM parameters on the performance. Such results can guide their proper selection, and provide directions on how they can be dynamically adjusted. Having focused on fairness, an interesting research topic is how marking can also be used to improve the aggregate throughput in a multihop wireless network.

References

1. S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin. *REM: Active Queue Management.* IEEE Network, pages 48-53, May/June 2001.
2. M. Bottigliengo, C. Casetti, C.F. Chiasserini, and M. Meo. *Short-term Fairness for TCP Flows in 802.11b WLANs.* Proc. of IEEE INFOCOM'04, 2004.
3. D. Chiu and R. Jain. *Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks.* Computer Networks and ISDN Systems, 17:1-14, June 1989.
4. Z. Fu, P. Zerfos, H. Luo, L. Zhang, and M. Gerla. *The Impact of Multihop Wireless Channel on TCP Throughput and Loss.* Proc. of IEEE INFOCOM'03, 2003.
5. H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khavizov. *TCP over 2.5G and 3G wireless networks.* RFC 3481, February 2003.
6. L. Qi Y. Shu K. Xu, M. Gerla. *Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED.* Proc. ACM MOBICOM'03, 2003.

7. S. Kunniyur and R. Srikant. *End-to-End Congestion control Schemes: Utility Functions, Random Losses and ECN Marks*. IEEE/ACM Trans. on Networking, 11(5):689-702, October 2003.
8. G. Montenegro, S. Dawkins, M. Kojo, V. Marget, and N. Vaidya. *Long Thin Networks*. RFC 2757, January 2000.
9. F. Peng, S.D. Cheng, and J. Ma. *A Proposal to Apply ECN to Wireless and Mobile Networks*. Proc. INET'00, 2000.
10. K.K. Ramakrishnan, S. Floyd, and D. Black. *The Addition of Explicit Congestion Notification (ECN) to IP*. RFC 3168, September 2001.
11. V.A. Siris. *Resource Control for Elastic Traffic in CDMA Networks*. Proc. of ACM MOBICOM'02, 2002.
12. V.A. Siris and D. Triantafyllidou. *Seamless Congestion Control over Wired and Wireless IEEE 802.11 Networks*. Proc. of Networking 2004, 2004.

Providing Delay Guarantees and Power Saving in IEEE 802.11e Network

G. Boggia, P. Camarda, F.A. Favia, L.A. Grieco, and S. Mascolo

DEE - Politecnico di Bari, Via E. Orabona, 4 - 70125 Bari Italy
{g.boggia, camarda, f.favia, a.grieco, mascolo}@poliba.it

Abstract. Recently, the 802.11e Working Group (WG) has proposed the Hybrid Coordination Function (HCF), which has a HCF Controlled Channel Access (HCCA) and an Enhanced Distributed Coordination Access (EDCA), in order to provide QoS in WLANs.

In this paper an innovative HCCA-based algorithm, which will be referred to as Power Save Feedback Based Dynamic Scheduler (PS FBDS) providing bounded delays while ensuring energy saving, has been developed. The performance of PS FBDS has been extensively investigated using ns-2 simulations; results show that the proposed algorithm is able to provide a good trade-off between QoS and power saving at both low and high network loads.

1 Introduction

Infrastructure 802.11 WLANs are a well assessed solution for providing ubiquitous wireless networking [1]. The building block in the architecture of such networks is the Basic Service Set (BSS), which consists of an Access Point (AP) and a set of Wireless Stations (WSTAs). The traffic from/to the WSTAs is channeled through the AP. WSTAs are usually batteries supplied devices, such as laptops or PDAs (Personal Digital Assistants), with limited lifetime [2, 3]. As a consequence, the power-saving issue becomes critical and limiting for a broader diffusion of WLAN equipped hot-spots [4]. To this aim, the 802.11 standard proposes a power saving (PS) mechanism in the Distributed Coordination Function (DCF), which is based on turning off the WNIC whenever a wireless station has not any frames to transmit/receive [1]. However, several works [5, 6, 7] have highlighted that 802.11 PS presents several inefficiencies and can severely affect the frame delivering delay, thus, making the 802.11 WLANs useless for real-time applications which have specific Quality of Service (QoS) requirements. For this reason, the works [8, 9, 10, 11] propose some optimizations of the 802.11 MAC when the PS is not used.

Moreover, the 802.11e Working Group (WG) has recently proposed a set of innovative functionalities in order to provide QoS in WLANs [12]. In particular, the core of the 802.11e proposal is the Hybrid Coordination Function (HCF), which has a HCF Controlled Channel Access (HCCA) and an Enhanced Distributed Coordination Access (EDCA). Previous works have shown that HCCA

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 323–332, 2005.

can be fruitfully exploited in conjunction with efficient scheduling algorithms in order to provide a bounded-delay service to real-time applications [13, 14, 15], but not considering any requirements on energy consumption. In order to bridge this gap, an innovative HCCA-based algorithm, which will be referred to as Power Save Feedback Based Dynamic Scheduler (PS FBDS), that provides bounded delays while ensuring energy saving, will be proposed in this paper. The performance of PS FBDS has been investigated using *ns*-2 simulations [16], which have shown that it is able to provide a good trade-off between QoS and power saving at both low and high network loads.

The rest of the paper is organized as follows: Section 2 gives an overview of the 802.11 MAC protocol and of QoS enhancements; Section 3 describes the theoretical background PS FBDS; Section 4 reports *ns*-2 simulation results. Finally, the last section draws the conclusions.

2 The IEEE 802.11 MAC

The 802.11 MAC employs a mandatory contention-based channel access scheme called Distributed Coordination Function (DCF), which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and an optional centrally controlled channel access scheme called Point Coordination Function (PCF) [1]. With PCF, the time is divided into repeated periods, called *Super-Frames* (SFs), which consist of a Contention Period (CP) and a Contention Free Period (CFP). During the CP, the channel is accessed using the DCF whereas, during the CFP, is accessed using the PCF.

Recently, in order to support also delay-sensitive multimedia applications, such as real-time voice and video, the 802.11e working group has enhanced the 802.11 MAC with improved functionalities. Four Access Categories (ACs), with different priorities, have been introduced. To satisfy the QoS requirements of each AC, the concept of TXOP (Transmission Opportunity) is introduced, which is defined as the time interval during which a station has the right to transmit and is characterized by a starting time and a maximum duration. The contiguous time during which TXOPs are granted to the same QSTA is called Service Period (SP).

Moreover, an enhanced access function, which is responsible for service differentiation among different ACs and is referred to as Hybrid Coordination Function (HCF), has been proposed [12]. The HCF is made of a contention-based channel access, known as the Enhanced Distributed Coordination Access (EDCA), and of a HCF Controlled Channel Access (HCCA). The use of the HCF requires a centralized controller, which is called the Hybrid Coordinator (HC) and is generally located at the access point. Stations operating under 802.11e specifications are usually known as enhanced stations or QoS Stations (QSTAs).

The EDCA method operates as the basic DCF access method but using different contention parameters per access category. In this way, a service differentiation among ACs is statistically pursued.

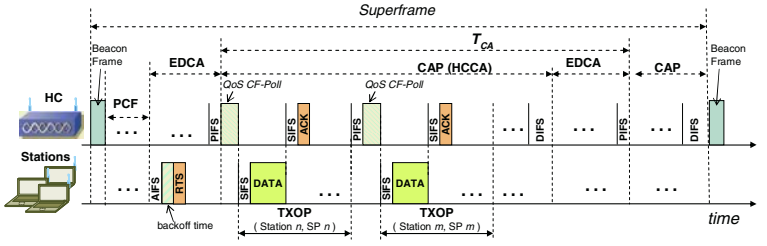


Fig. 1. Scheme of a superframe using the HCF controlled access method

The HCCA method combines some of the EDCA characteristics with some of the PCF basic features (see Fig. 1). The time is partitioned into superframes; each of them starts with a beacon frame after which, for legacy purpose, there could be a contention free period for PCF access. The remaining part of the superframe forms the CP, during which the QSTAs contend to access the radio channel using the EDCA mechanism.

During the CP, the HC can start a Contention Access Phase (CAP), in which only QSTAs, polled and granted with the *QoS CF-Poll frame*, are allowed to transmit during their TXOPs. Thus, the HC implements a prioritized medium access control. CAP length cannot exceed the value of the system variable *dot11CAPLimit*, which is advertised by the HC in the Beacon frame when each superframe starts [12].

According to IEEE 802.11e specifications, each QSTA can feed back queue length of each AC to the HC in each frame header. As shown in this paper, this information can be fruitfully exploited to design novel HCCA-based dynamic bandwidth allocation algorithms using feedback control theory. In fact, the 802.11e draft does not specify how to schedule TXOPs in order to provide the required QoS; it only suggests a simple scheduler which assigns fixed TXOPS using the static values declared in the Traffic Specifications (TSPECs) during the admission phase.

2.1 Overview of the Power Saving in 802.11 Infrastructure WLANs

The power saving issue has been addressed in the 802.11 standard [1], by defining two different power states for a station: the *Awake State* in which the station is fully powered (i.e., the WNIC is on and consumes the power needed to transmit/receive frames and to sense the channel); the *Doze State* in which the station is not able to transmit or receive (i.e, the WNIC consumes very low power). Moreover, two power management modes have been introduced: the *Active* Mode in which a station may receive frames at any time, i.e., it is always in awake state; the *Power Save* (PS) Mode in which a station is normally in the *Doze State* and enters in the *Awake State* to transmit frames and to receive beacon frames, broadcast, and unicast transmissions.

The AP cannot transmit data to stations operating in PS mode, but it has to buffer frames and to transmit them only when such stations are in awake state.

In all beacon frames, the *traffic indication message* (TIM) is sent to indicate stations in PS mode (i.e., PS stations) which have buffered data in the AP. The buffered broadcast and multicast frames are signaled through the *delivery* TIM (DTIM) element.

A PS Station shall wake up in order to receive the Beacon frame and to detect if the AP has buffered frames for it.

If the PS station accesses the channel with the DCF method and there are pending data in the AP, during the Contention Period (CP) it sends to the AP a PS-Poll frame in order to receive buffered data, then can transit in doze state. Broadcast frames are sent immediately after the beacon frame that includes DTIM.

2.2 IEEE 802.11e Power Saving Enhancements

The IEEE 802.11e Working Group have introduced a new power saving mechanism, known as *Automatic Power Save Delivery* (APSD) [12], which allows the delivery of downlink frames to a station according to a defined "schedule", i.e., the downlink frames are transmitted by the HC only in given service periods. In particular, when APSD is active, the HC buffers the data frames addressed to APSD stations (i.e., stations which use the APSD mode) in doze state and it transmits them according to two different type of service periods: *Scheduled* and *Unscheduled*. Scheduled service periods occur always at the same time instants in the superframe and they are assigned by the HC to a station when a new traffic stream starts. During its scheduled service period, a station is awake to receive buffered downlink frames and/or polls the HC. Unscheduled service periods are asynchronous in the superframe and they occur as soon as the HC knows that the APSD station wakes up by receiving any frame from the station.

3 The PS FBDS Bandwidth Allocation Algorithm

This section summarizes the FBDS bandwidth allocation algorithm proposed in [13] and introduces its power saving extension PS FBDS. The algorithm, which has been designed using classical feedback control theory, distributes the WLAN bandwidth among all the multimedia flows by taking into account the queue levels fed back by the QSTAs.

We will refer to a WLAN system made of an Access Point (AP) and a set of quality of service enabled mobile stations (QSTAs). Each QSTA has N queues, with $N \leq 4$, one for any AC in the 802.11e proposal. Let T_{CA} be the time interval between two successive CAPs. Every time interval T_{CA}, assumed constant, the AP must allocate the bandwidth that will drain each queue during the next CAP. We assume that at the beginning of each CAP, the AP is aware of all the queue levels $q_i,\ i = 1, \ldots, M$ at the beginning of the previous CAP, where M is the total number of traffic queues in the WLAN system.

The dynamics of the i^{th} queue can be described by the following discrete time linear model:

$$q_i(k+1) = q_i(k) + d_i(k) \cdot T_{CA} + u_i(k) \cdot T_{CA}, \qquad i = 1, \ldots, M, \tag{1}$$

where $q_i(k) \geq 0$ is the i^{th} queue level at the beginning of the k^{th} CAP; $u_i(k) \leq 0$ is the average depletion rate of the i^{th} queue (i.e., the bandwidth assigned to drain the i^{th} queue); $d_i(k) = d_i^s(k) - d_i^{CP}(k)$ is the difference between $d_i^s(k) \geq 0$, which is the average input rate at the i^{th} queue during the k^{th} T_{CA} interval, and $d_i^{CP}(k) \geq 0$, which is the amount of data transmitted by the i^{th} queue during the k^{th} T_{CA} interval using EDCA divided by T_{CA}.

The signal $d_i(k)$ is unpredictable since it depends on the behavior of the source that feeds the i^{th} queue and on the number of packets transmitted during the contention periods. Thus, from a control theoretic perspective, $d_i(k)$ can be modelled as a disturbance. Without loss of generality, a piece-wise constant model for the disturbance $d_i(k)$ can be assumed: $d_i(k) = \sum_{j=0}^{+\infty} d_{0j} \cdot 1(k - t_j)$, where $1(k)$ is the unitary step function, $d_{0j} \in \mathbb{R}$, and t_j is a time lag.

Due to this assumption, the linearity of the system (1), and the superposition principle that holds for linear systems, we will design the feedback control law by considering only a step disturbance: $d_i(k) = d_0 \cdot 1(k)$.

3.1 The Control Law

We design a control law to drive the queuing delay τ_i experienced by each frame of the i^{th} queue to a desired target value τ_i^T, representing the QoS requirement of the AC associated to the queue. In particular, we consider the control law:

$$u_i(k+1) = -k_i \cdot q_i(k) \tag{2}$$

which gives the way to compute the $\mathcal{Z}$-transform of $q_i(k)$ and $u_i(k)$ as follows:

$$Q_i(z) = \frac{z \cdot T_{CA}}{z^2 - z + k_i \cdot T_{CA}} D_i(z); \qquad U_i(z) = \frac{-k_i \cdot T_{CA}}{z^2 - z + k_i \cdot T_{CA}} D_i(z) \tag{3}$$

whit $D_i(z) = \mathcal{Z}[d_i(k)]$. From eq. (3)the system poles are $z_p = \frac{1 \pm \sqrt{1 - 4k_i \cdot T_{CA}}}{2}$, which give an asymptotically stable system if and only if $|z_p| < 1$, that is:

$$0 < k_i < 1/T_{CA}. \tag{4}$$

In the sequel, we will always assume that k_i satisfies this asymptotic stability condition stated by (4).

To investigate the ability of the control system to provide a queuing delays approaching the target value τ_i^T, we apply the final value theorem to Eq. (3). By considering that the $\mathcal{Z}$-transforms of the step function $d_i(k) = d_0 \cdot 1(k)$ is $D_i(z) = d_0 \cdot \frac{z}{z-1}$ the following results turn out:

$$u_i(+\infty) = \lim_{k \to +\infty} u_i(k) = \lim_{z \to 1} (z-1) U_i(z) = -d_0; \qquad q_i(+\infty) = d_0/k_i,$$

which implies that the the steady state queueing delay is:

$$\tau_i(+\infty) = |q_i(+\infty)/u_i(+\infty)| = 1/k_i. \tag{5}$$

Thus, the following inequality has to be satisfied in order to achieve a steady-state delay smaller than τ_i^T:

$$k_i \geq 1/\tau_i^T. \tag{6}$$

By considering inequalities (4) and (6) we obtain that the T_{CA} parameter has to fulfill the following constraint:

$$T_{CA} < \min_{i=1..M} \tau_i^T. \tag{7}$$

3.2 Implementation Issues

Starting from the allocated bandwidth u_i, if the i^{th} queue is drained at data rate C_i, the assigned $TXOP_i$ is obtained by the following relation [13]:

$$TXOP_i(k) = |u_i(k) \cdot T_{CA}|/C_i + O \tag{8}$$

where $TXOP_i(k)$ is the TXOP assigned to the i^{th} queue during the k^{th} service interval and O is the time overhead due to ACK packets, SIFS and PIFS time intervals (see Fig. 1). The extra quota of TXOP due to the overhead O depends on the number of frames corresponding to the amount of data $|u_i(k) \cdot T_{CA}|$ to be transmitted. O could be estimated by assuming that all frames have the same nominal size specified into the TSPEC.

The above bandwidth allocation is based on the implicit assumption that the sum of the TXOPs assigned to each queue is smaller than the maximum CAP duration, which is the *dot11CAPLimit*; this value can be violated when the network is saturated.

In this case, it is necessary to reallocate the TXOPs to avoid exceeding the CAP limit. Each computed $TXOP_i(k)$ is decreased by an amount $\Delta TXOP_i(k)$ proportional to C_i and $TXOP_i(k)$ [13], in order to obtain a fair bandwidth assignment.

3.3 Power Save FBDS

To manage the power saving, at the beginning of each superframe, a station using PS FBDS wakes up to receive beacon frames. Then, if the HCCA method is used, it does not pass in *doze state* until it has received the QoS-Poll frame and the TXOP assignment from the HC. After the station has drained its queue according to the assigned TXOP, it will transit in *doze state* only if there are not new poll or data frames from the HC.

When the EDCA is used, the station wakes up as soon as any of its queues becomes not empty. In this case, the backoff timer is set to zero; thus, a wireless station will gain the access to the channel with a higher probability than stations using the classical EDCA. In the sequel, we will refer to this slightly modified version of the EDCA as Power Save EDCA (PS EDCA).

4 Performance Evaluation

To test the effectiveness of PS FBDS, we have implemented the algorithm using the *ns*-2 simulator [16] and we have run computer simulations involving audio, video and FTP data transfers. We have considered a WLAN network shared by a mix of 3α audio flows encoded with the G.729 standard, α video flows encoded with the MPEG-4 standard, α encoded with the H.263 standard, and α FTP best effort flows. From each wireless node, a single data flow is generated. Main characteristics of the considered multimedia flows are summarized in Table 1.

Table 1. Main features of the considered multimedia flows

Type of flow	*Nominal (Maximum) MSDU Size*	*Mean (Maximum) Data Rate*	*Target Delay*
MPEG-4 HQ	1536(2304) byte	770 (3300) kbps	40 ms
H.263 VBR	1536(2304) byte	450 (3400) kbps	40 ms
G.729 VAD	60(60) byte	8.4 (24) kbps	30 ms

In the *ns*-2 implementation the T_{CA} is expressed in Time Units (TU), which in the 802.11 standard [1] are equal to 1024μs. We assume a T_{CA} of 29 TU. The proportional gain k_i is set equal to $1/\tau_i^T$. We have compared FBDS, PS FBDS, EDCA, and PS EDCA algorithms for different network loads, by varying the load parameter α.

For what concern the power consumption parameters, we consider a RF Transceiver which has: Tx power of 393 mW, Rx power of 357 mW, stand-by power of 125 mW, doze power of 33 μW. In the simulation each station has an initial energy equal to 10 J. Stations hosting FTP flows do not use any power saving extensions. FTP flows are used to fill in the bandwidth left unused by flows with higher priority. Power saving mechanism are enabled after the first 15 s of warm-up period; thus, delays are evaluated without considering this warm-up time interval.

Fig. 2.a show the average value of the one-way packet delay experienced by the MPEG flows for various values of the load factor α. It shows that both FBDS and PS-FBDS provide the smallest delays at high network loads (i.e., when $\alpha \geq 6$). The reason is that FBDS allocates the right amount of bandwidth to each flow by taking into account the transmitting queue levels of the wireless nodes, this allows a cautious usage of the WLAN channel bandwidth, so that QoS constraints are met also in the presence of a high number of competing flows.

Regarding the power saving issue, Figs. 2.b-2.d report the residual energy of a node hosting a MPEG traffic source for $\alpha = 3,\ 6,$ and 9. When $\alpha = 9$, i.e., at high network load, it is straightforward to note that a great energy saving can be achieved using PS FBDS. In fact, after 50 s of activity, PS FBDS leads to a total energy consumption less than 5 J, whereas, with the other schemes,

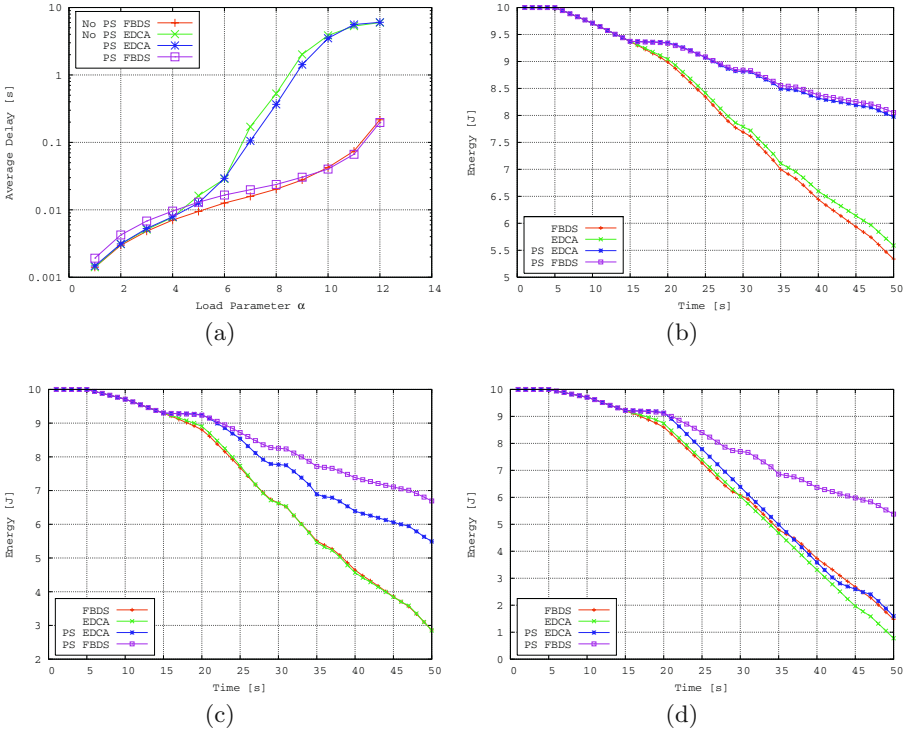


Fig. 2. Results for MPEG4 flows. (a) Average one-way delays vs. load parameter α. Average Residual Energy using (b) $\alpha = 3$; (c) $\alpha = 6$; (d) $\alpha = 9$

the energy consumption is larger than 8 J. When $\alpha = 3$, i.e., at low network load, PS HCCA and PS EDCA provide the same energy saving because, at low network load, almost all traffic is served during the CP using the EDCA. For $\alpha = 6$ intermediate results are obtained. Thus PS FBDS allows energy saving while providing the same delay bounds of the original FBDS algorithm.

Similar conclusions have been obtained for the H.263 traffic streams; results are not reported due to lack of space.

Results are very different when we consider G.729 flows (see Fig. 3). In fact, we have to consider that these flows are served with the maximum priority by the EDCA, and that the PS EDCA is more aggressive than standard EDCA.

Fig. 3.a shows that, when PS EDCA is used, the smallest delays are obtained. However, from these figures it can be noticed that delays provided by the other considered schemes are smaller than 100 ms also at high network loads, i.e., PS FBDS, FBDS, and standard EDCA provide acceptable performance.

Regarding energy consumption, Fig. 3.d reports the residual energy of a node hosting a G.729 traffic source when $\alpha = 9$. While PS FBDS, FBDS and EDCA provide almost the same energy consumption as in the previous simulations,

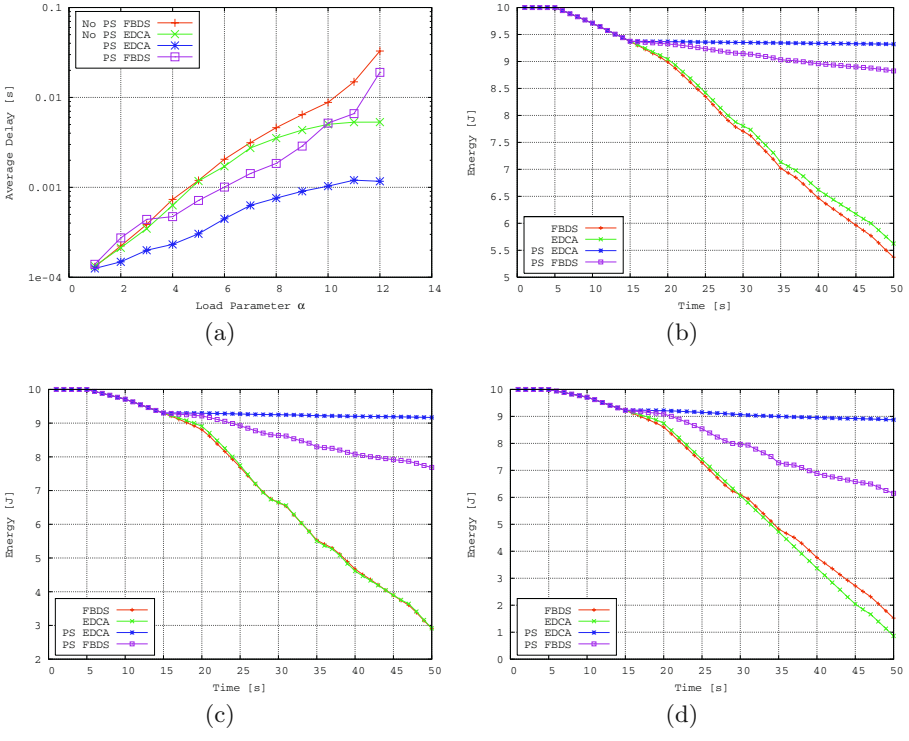


Fig. 3. Results for G.729 flows. (a) Average one-way delays vs. load parameter α. Average Residual Energy using (b) $\alpha = 3$; (c) $\alpha = 6$; (d) $\alpha = 9$

PS-EDCA enables the best energy saving. The reason is that the aggressiveness of PS EDCA allows stations hosting G.729 flows to listen the channel for very short time intervals before transmitting with an immediate impact on energy consumption. The gap between PS FBDS and PS EDCA diminishes for smaller values of α for the same reasons discussed above (see Figs. 3.b and 3.c).

5 Conclusion

In this paper, th PS FBDS scheduling algorithm has been proposed to achieve a good trade-off between power saving and QoS using 802.11e MAC. It has been designed using classical feedback control theory. Its performance has been investigated using *ns*-2 simulations in realistic scenarios where the wireless channel is shared by heterogeneous traffic flows. Simulation results show that PS FBDS is able to provide a bounded delay service to real-time flows and at the same time to significantly reduce energy consumption at both high and low network loads.

References

1. IEEE 802.11: Information Technology - Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std. 802.11, ISO/IEC 8802-11. First edn. (1999)
2. Baiamonte, V., Chiasserini, C.F.: An energy-efficient mac layer scheme for 802.11-based wlans. In: Proceedings of the 2004 IEEE International Conference on Performance, Computing, and Communications. (2004) 689 – 694
3. Woesner, H., Ebert, J.P., Schlager, M., Wolisz, A.: Power-saving mechanisms in emerging standards for Wireless LANs: the MAC level perspective. IEEE Personal Communications (1998) 40–48
4. Anastasi, G., Conti, M., Gregori, E., Passarella, A.: A performance study of power-saving polices for Wi-Fi hotspots. Computer Networks **45** (2004) 295–318
5. Choi, J.M., Ko, Y.B., Kim, J.H.: Enhanced Power Saving Scheme for IEEE 802.11 DCF Based Wireless Networks. In: Proceedings of the International Conference on Personal Wireless Communication (PWC'03). (2003) 835 – 840
6. Anand, M., Nightingale, E.B., Finn, J.: Self-tuning Wireless Network Power Management. In: Proceedings of the ACM Mobicom 2003. (2003)
7. Jung, E., Vaidya, N.H.: An energy efficient MAC protocol for wireless LANs. In: Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, ; INFOCOM 2002. Volume 3. (2002) 1756–1764
8. Bononi, L., Conti, M., Donatiello, L.: A Distributed Mechanism for Power Saving in IEEE 802.11 Wireless LANs. ACM/Baltzer Mobile Networks and Applications (MONET) **6** (2001)
9. Bruno, R., Conti, M., Gregori, E.: Optimization of Efficiency and Energy Consumption in p-Persistent CSMA-Based Wireless LANs. IEEE Transactions on Mobile Computing **1** (2002)
10. Yan, S., Zhuo, Y., Wu, S.: An Adaptive RTS Threshold Adjust Algorithm based on Minimum Energy Consumption in IEEE 802.11 DCF. In: Proceedings of the International Conference on Communication Technology (ICCT'2003). (2003) 1210–1214
11. Hsu, C.S., Sheu, J.P., Y.-C.Tseng: Minimize Waiting Time and Conserve Energy by Scheduling Transmissions in IEEE 802.11-based Ad Hoc Networks. In: Proceedings of the International Conference on Telecommunications (ICT 2003). (2003) 393–399
12. IEEE 802.11 WG: Draft Amendment to Standard for Information Technology - Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements. IEEE 802.11e/D10.0. (2004)
13. Boggia, G., Camarda, P., Grieco, L.A., Mascolo, S.: Dynamic bandwidth allocation with call admission control for providing delay guarantees in IEEE 802.11e networks. Computer Communications, special issue **28** (2005) 325–337
14. Grilo, A., Macedo, M., Nunes, M.: A scheduling algorithm for QoS support in IEEE 802.11e networks. IEEE Wireless Communications (2003) 36–43
15. Ansel, P., Ni, Q., Turletti, T.: FHCF: A Fair Scheduling Scheme for 802.11e WLAN. Institut National de Recherche en Informatique et en Automatique (INRIA). (2003)
16. Ns-2: Network simulator. available at `http://www.isi.edu/nsnam/ns` (2004)

Measuring Transport Protocol Potential for Energy Efficiency

S. Kontogiannis, L. Mamatas, I. Psaras, and V. Tsaoussidis

Democritus University of Thrace,
Dept. Of Electrical & Computer Engineering, Xanthi, Greece
{skontog, emamatas, ipsaras, vtsaousi}@ee.duth.gr

Abstract. We investigate the energy-saving potential of transport protocols. We focus on the system-related aspect of energy. Do we have to damage or enhance system fairness in order to provide energy efficiency? We depart from defining protocol potential; we compare different transmission strategies and protocol mechanisms; and we report our results on the impact of each mechanism on system energy. We highlight our conclusion that protocol fairness appears to be a key factor for system energy efficiency.

1 Introduction

Energy consumption is becoming a crucial factor for wireless, ad-hoc and sensor networks, which affects system connectivity and lifetime. Standard TCP, originally designed for wired network infrastructure, does not cope with wireless conditions such as fading channels, shadowing effects and handoffs, which influence energy consumption.

Wireless network interface cards usually have four basic states of operation and each of these states has different power requirements. The most power-demanding states are the active states where transmission and reception of data take place. The standby/listen state, is the state where a network interface card is simply waiting. The extended period of idle state may lead to a sleep state, which is the least power-demanding state, where the radio subsystem of the wireless interface is turned off. Note that the transition mechanism itself is also energy consuming. Regardless of the states, their number and the frequency of transition, energy consumption is itself device-specific.

Due to the complexity of energy management and the fact that the state transition is device specific, each transmission or reception attempt by a higher-layer protocol does not necessarily correspond to a similar power transition. That is, we cannot accept *apriori* that the measured energy expenditure reflects the ability of a protocol to administer energy resources. Therefore, we distinguish protocol energy potential from actual device expenditure. The former approaches the latter when the sophistication of devices increases in a manner that all network layers operate in parallel states. Otherwise, if higher-layer protocol operation is suspended but the power module does not adjust, the protocol potential cannot translate into energy efficiency.

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 333–342, 2005.

Several attempts have been made to measure the energy efficiency of transport protocols, (e.g. [10], [12]) as well as their potential for energy efficiency [14]. Energy efficiency is clearly device-specific while energy potential is not clearly defined. We attempt to define the latter, by introducing a corresponding index; we also attempt to measure actual expenditure, using specific device characteristics. We used Goodput in order to characterize protocol potential and an experimental extra energy expenditure index in order to characterize protocol energy performance.

Furthermore, we go beyond measuring energy potential within the confines of a single flow operation. We also investigate the system behavior of protocols attempting to address the question: "What are the design characteristics of transport protocols that impact system rather than single-flow energy efficiency"? In other words, what is the behavior of energy-efficient protocols within a multi-flow system? We noticed at this early stage of our investigation, some interesting results. While protocol Goodput is an important factor for energy efficiency (as we have also shown in [14]), protocol fairness seems to be another key factor for *system* energy efficiency.

The structure of this paper is the following: In section 2 we present the congestion control mechanisms that affect energy performance, according to distinct wireless conditions. In section 3 we present our proposed energy expenditure and energy potential metrics. In section 4 we present our scenario and evaluation plan and in section 5 we discuss the results.

2 Transmission Strategies and Network Conditions

The basic factor that determines the transmission strategies of the transport protocols is the window adjustments made by the congestion control algorithms. Different protocols employ distinct algorithms to control congestion. We focus on two basic categories of such algorithms. The first one considers the network as a black box and hence follow a blind procedure; the second one measures network conditions and adjust accordingly.

In the first category, in which most standard TCP versions belong, there are four widely available versions: Tahoe, Reno, New Reno and Sack. Tahoe is the most conservative version which includes Slow Start and Fast Retransmit [5], [8]. Reno is somewhat more aggressive due to its Fast Recovery mechanism. New Reno is even more aggressive when multiple drops occur within a single window of data, while Sack [9], the newest TCP version, is the most aggressive due to its selective acknowledgment strategy and its associated selective repeat mechanism.

The second category is represented by various standard (e.g. Vegas [2]) or experimental (e.g. Westwood [3], [11], Real [17], Jersey [7]) TCP protocols. We selected Vegas and Westwood for our experiments. TCP Vegas [2] congestion control is based on sample RTT measurements. The sender calculates throughput rate every RTT. This rate is compared to an expected rate, which is calculated based on what is measured as best RTT. TCP Westwood computes a sample

of Bandwidth by measuring and low pass filtering the rate of returning ACKs. TCP Westwood departs from the AIMD paradigm by proposing the additive increase adaptive decrease (AIAD) paradigm. No theoretical proof is given that AIAD converges to fairness.

In the context of transport protocol energy potential, we cannot isolate transmission strategy apart from distinctive error characteristics. We consider two major categories of errors, which are further classified into four different types. Each one of them calls for distinctive transmission tactics. We note that these types by no means traverse in detail the whole spectrum of distinct errors but are rather abstract. The first category, *congestion losses*, is separated into two types: *burst congestion losses* and *transient congestion losses*. During burst errors several consecutive transmitted packets are lost due to buffer overflow. By the term transient congestion errors, we characterize a situation where a small number of flows coexist in the same channel, causing in that way buffer overflowing sparsely, (e.g due to TCP synchronization). It is clear that both types of this category are associated with system's queuing delay. Under such conditions, we expect that the timeout mechanisms of the transport protocols have to be adjusted to accommodate the extra queuing delay. Furthermore, in case of burst congestion errors, the congestion window have to be drastically reduced, while transient errors may require smooth window adjustments.

The second category, *non-congestion losses*, includes the last two types of errors: *burst non-congestion errors* and *transient/random non-congestion errors*. Non-congestion losses, appear mostly in wireless/heterogeneous networks. Burst errors in the wireless portion of the network include handoffs, shadowing events, errors due to low SNR, etc. Under such conditions, data transmission would better be suspended until the communication channel recovers. This idea is implemented in TCP-Probing [13] where a probing mechanism gets aware of the situation and suspends data transmission for as long as the error persists.

Current TCP versions including these in our experiments, cannot distinguish those categories but mainly differentiate their mechanisms towards congestion losses. In other words, current TCP protocols are not suited for the distinct characteristics of wireless networks and thus an ideal protocol that can distinguish between those characteristics, could be much more energy efficient.

3 Measuring Energy Performance

In order to evaluate TCP performance over wireless networks and present useful directions in the context of energy consumption, we used traditional metrics, such as system *Goodput* and *Fairness Index* along with: *Extra Energy Expenditure* [10]. System Goodput is used to measure the overall system efficiency in bandwidth utilization and defined by (1). Fairness is measured by Fairness Index, derived from the formula given in (2).

$$Goodput = \frac{OriginalData}{ConnectionTime} \tag{1}$$

$$\mathcal{F.I.} = \frac{(\sum_{i=0}^{n} \|Throughput_i\|)^2}{n(\sum_{i=0}^{n} \|Throughput_i\|^2)} \tag{2}$$

The energy efficiency of a protocol is defined as the average number of successful transmissions per energy unit, which can also be computed as the average number of successes per transmission attempt as pointed out by Jones et al [6]. Energy expenditure or energy efficiency is a very important factor that has a major impact on wireless, battery-powered devices. However, apart from the overhead metric, there is no other metric in the literature that monitors the potential of a protocol for energy saving. Departing from that point and in order to capture the amount of *extra* energy expended, we introduce a new metric that was first presented in [16]. We call this new metric, *Extra Energy Expenditure (3E)*. The 3E metric, quantifies the extra effort expended without return in Goodput as well as the energy loss due to insufficient effort when aggressive transmission could have resulted in high Goodput. Three variables take place in this new metric. These are Throughput$_{max}$, Throughput and Goodput. The idea behind Throughput$_{max}$ is that it captures the best possible data transmission that can be achieved under the given network conditions. The other two variables are the Throughput and Goodput metrics that monitor protocol performance. The 3E metric is given by the following formula:

$$\mathcal{EEE} = \alpha \frac{Thr - Goodput}{Thr_{max}} + b \frac{Thr_{max} - Thr}{Thr_{max}} \tag{3}$$

In order to explore the extra energy expenditure of a system of flows, we introduce system's 3E. System's 3E is equal to the sum of all competing flows extra energy expenditure:

$$\mathcal{EEE}_s = \alpha \frac{\sum_{i=1}^{n}(Thr_i - G_i)}{Thr_{max}} + b \frac{Thr_{max} - \sum_{i=1}^{n} Thr_i}{Thr_{max}} \tag{4}$$

It is clear that in all cases, Throughput$_{max}$ $\geq$ Throughput $\geq$ Goodput. Extra Energy Expenditure (3E) takes into account the difference of achieved Throughput from maximum Throughput (Throughput$_{max}$) for the given channel conditions, as well as the difference of Goodput from Throughput, attempting to locate the Goodput as a point within a line that starts from 0 and ends at Throughput$_{max}$. All available energy is consumed into efficient transmissions only when $Thr - Goodput = Overhead$ and $Thr = Thr_{max}$. That is, for an ideal TCP protocol that has an overhead of 40 Bytes in a 1024 Bytes TCP segment, EEE should be:

$$\mathcal{EEE} = \alpha \frac{0.04}{Thr_{max}} \tag{5}$$

In order for the 3E index to estimate the device specific extra energy expenditure, the value of α must be linked with the device transmission power: $\alpha = P_{Tx}(W)$ and the value of b must be linked with the device idle power: $b = P_{Idle}(W)$. In our experiments we normalized our α, and b parameters according to the the Lucent OriNOCO wireless device. We used the values of $\alpha = 1$ and $b = 0.45$.

4 Experimental Methodology

We have implemented a scenario, with two wireless nodes: The sender (node 0) and the receiver (node 1). The simulator used was the ns-2 network simulator and the topology an area 100x100 meters with a stable 100 meter distance between transmitter and receiver. The wireless link capacity is 2 Mbit. We used ns-2 energy model to simulate a specific device energy expenditure. The power values that were used for transmit, receive and idle states, where those of the Lucent OriNOCO wireless card. In our experiments we used a two-state error model for the process of packet errors, combined with the Bernoulli geometric distribution, to simulate the probability of packet drops. This is also known as the Gilbert channel model [4].

Our evaluation plan is consisted of two stages. At the first stage we modified the error-rate for a single flow scenario. We used different transport protocols in order to confirm the impact of different congestion control strategies energy potential and energy expenditure, for the one-flow system. At the second stage of this plan we modified the number of the flows for distinct error rates. Points of interest for us were those ones with similar Goodput performance but different fairness performance, utilizing different energy potential; or, those with worse Goodput performance which however were counterbalanced by fairness performance, resulting in better energy performance.

5 Results and Discussion

5.1 One-Flow Scenario Results

Energy expenditure or energy efficiency is a very important factor that has a major impact on wireless battery-powered devices. It is known already that a communication channel with low error rates should be utilized aggressively; when the error rate increases, a more conservative behavior yields better results.

Figure 1(a) compares the three standard TCP versions. TCP Reno seems to be more energy consuming when the packet error rate is greater than 15%. This is probably happening because Reno does not back-off to its initial congestion window (like Tahoe does) and neither does it recover with Fast Recovery (like New Reno).

Similarly, in figure 1(b) Vegas does not waste much energy when the error rate is low, while for higher error rates, Vegas behaves aggressively and underachieves in terms of energy potential. More precisely, Vegas algorithm estimates accurately the available bandwidth at low error rates and thus presents better energy potential. However, when the error rate increases, Vegas estimator seems to estimate the available bandwidth without taking into consideration the persistent error conditions of the network. Under these conditions, Vegas false estimations are clearly outperformed by Tahoe's conservative strategy. Based on the above analysis, we confirm that a more aggressive behavior (Vegas) performs better under low error rate conditions, while the opposite might happen when

the error rate increases. Furthermore, Goodput proves one more time to be the most significant factor for TCP Energy Efficiency.

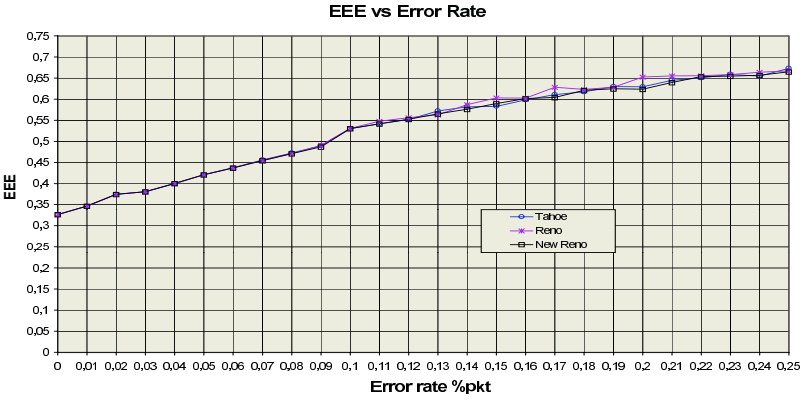
In the same scenario, Sack protocol neither appears energy efficient (figure 1(b)), nor does it achieve satisfactory Goodput performance. As Singh and Singh [12] stated, Sack "energy" performance suffers from extended timeouts and computational burden. In multi-drop situations where New Reno would timeout, Sack aggressively continues to retransmit packets. The aggressive retransmissions, along with the computational burden and the extended timeouts are translated into extra energy expenditure.

TCP Westwood occasionally fails to adjust to the level of the available bandwidth, mainly at burst errors. Also it utilizes an adaptive policy appropriate for congestive losses and not for wireless errors. That is why its performance cannot overcome the performance of conservative TCP Tahoe both at random and burst error rates. However, as shown in figure 1(c), Westwood estimates available bandwidth more accurately at low error rates. For Westwood, when Goodput increases also energy potential increases.

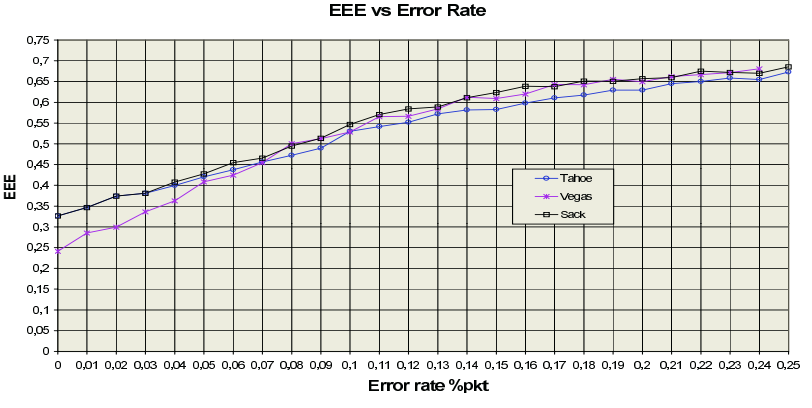
5.2 Multiple-Flow Scenario Results

We confirmed from previous one-flow scenario results that as Goodput performance increases, energy performance increases as well. The aforementioned conclusion is not quite accurate for a multi-flow system. In that case both Goodput and Fairness affect energy performance. In order to confirm the latter, we compare the behavior of two systems of flows. The first system utilizes TCP Vegas flows, while the second system utilizes TCP Tahoe flows. We focus on finding the points where both systems have the same amounts of Goodput but different values of Fairness index. According to figure 2(b), For a system of 5, 8 and 25 flows, Tahoe's Goodput is equal or more than Vegas. On the other hand, Vegas is more fair than Tahoe for the 5, 8 and 25 flows systems. The impact of such behavior on energy performance is depicted in figure 2(c). Vegas increases its energy performance towards Tahoe, even if Tahoe performs equal or even better than Vegas. That is Tahoe shows increased amount of Goodput compared to Vegas. This confirms further our assertion that fairness does contribute to the system's energy potential and energy performance. For a system of flows both Fairness and Goodput should be increased in order to improve protocol energy potential.

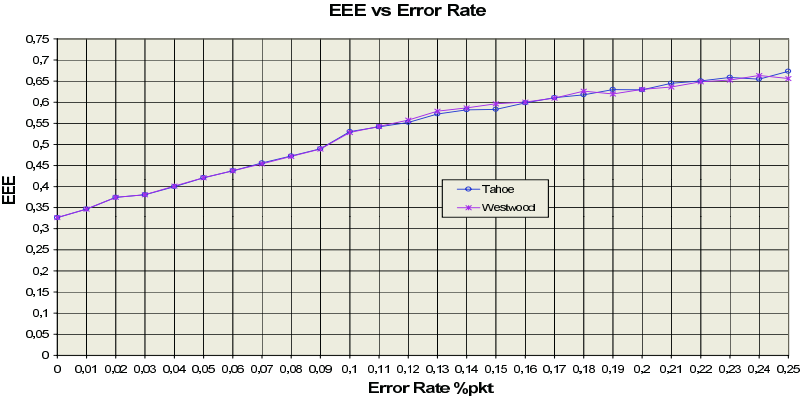
How far is fairness a dominant factor for energy efficiency? As we can see in figures 2(a) and 2(b), for a system of 3 flows, Vegas protocol is fair compared to Tahoe but performs poorly in terms of Goodput. The result for this system is that Tahoe has better energy potential. There is a point from where protocol energy performance is not affected by fairness, or, in other words, fairness impact on protocol energy performance is not the dominant factor. Moreover, as Goodput difference between two systems increases, fairness impact on protocol energy performance decreases. From a point and beyond, energy performance is mainly affected by Goodput performance. Systems Energy Expenditure accommodates the behavioral characteristics of systems energy potential. As depicted



(a) Tahoe, Reno and New Reno Extra Energy Expenditure vs Error rate.

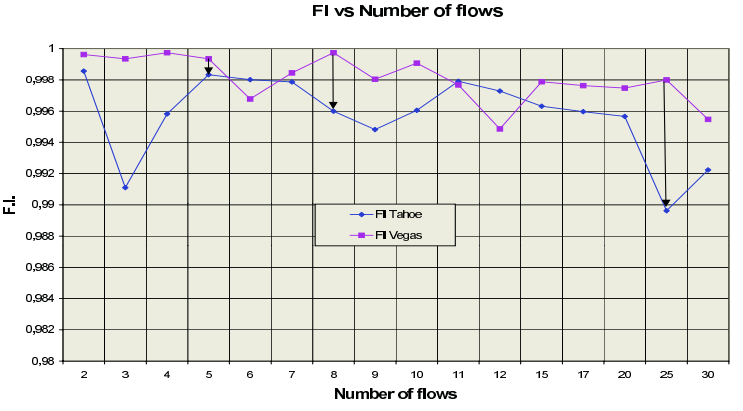


(b) Tahoe, Vegas and Sack Extra Energy Expenditure vs Error rate.

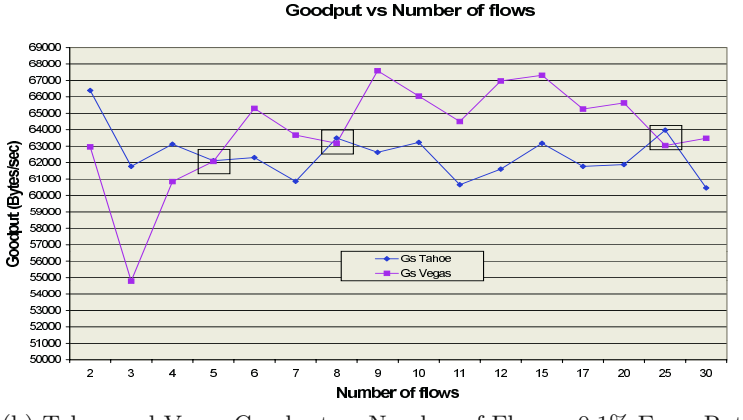


(c) Tahoe and Westwood Extra Energy Expenditure vs Error rate.

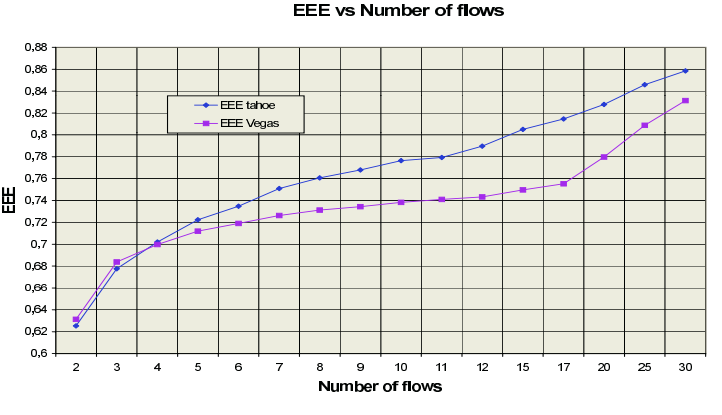
Fig. 1. TCP Protocols (a) Tahoe Reno and New Reno, (b) Tahoe Vegas and Sack (c) Tahoe and Westwood, Extra Energy Expenditure index vs Error rate



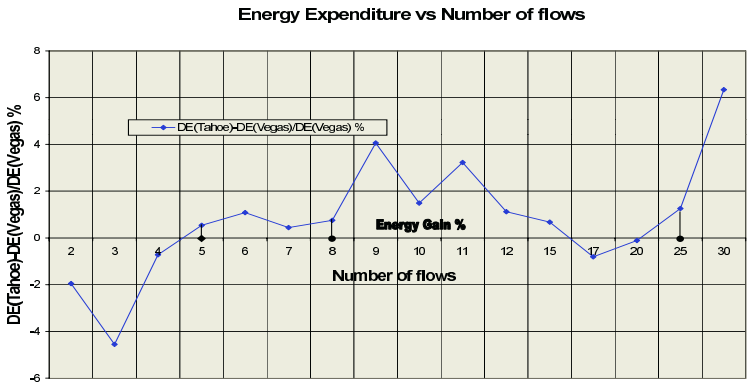
(a) Tahoe and Vegas Fairness Index vs Number of Flows - 0.1% Error rate.



(b) Tahoe and Vegas Goodput vs Number of Flows - 0.1% Error Rate.



(c) EEE vs Number of Flows - 0.1% Error Rate.



(d) Energy Gain vs Number of Flows - 0.1% Error Rate.

Fig. 2. TCP Protocols (a) F.I., (b) Goodput, (c) Extra Energy Expenditure and (d) Energy Gain

in figure 2(d), for the marked points 5, 8 and 25 of the Vegas flows system, fairness increases and Goodput decreases while system's protocol energy potential increases. The actual energy gain of Tahoe versus Vegas due to the difference in fairness does not exceed 1% of the transmitter node total energy expenditure, while in general energy gain of Tahoe reaches 6%. However both protocols are far from reaching energy-conserving strategies. That is a new design can clearly reach much greater levels of energy efficiency.

6 Conclusions

Energy saving is not a property of one operation, layer, or protocol: Many design factors of different levels can contribute to achieve energy gains. We attempted to isolate energy gains due to transport protocol design characteristics. Since the energy-saving functionality of transport protocols may not be reflected in actual energy savings, due to device limitations, we introduced the notion of energy potential and linked it with the Extra Energy Expenditure (3E) index. We also adjusted this index to a specific device in order to establish a relation of *potential* with *real* expenditure. Using the aforementioned criteria, we evaluated the energy behavior of transport protocols. We report two important conclusions. First, we confirmed our previous assertion that high Goodput does contribute towards energy saving. Second, we observed that fairness is inherently correlated with system energy: when two systems achieve similar Goodput performance, the one that is more fair appears to be more energy-efficient as well.

References

1. I. Batsiolas and I. Nikolaidis. "Selective Idling: Experiments in Transport Layer Energy Conservation". *In: Journal of Supercomputing, Special Issue on Design and Evaluation of Transport Services*, pages 101–114, 2001.
2. Lawrence S. Brakmo, Sean W. O'Malley, and Larry L. Peterson. TCP vegas: New techniques for congestion detection and avoidance. In *SIGCOMM*, pages 24–35, 1994.
3. C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi and R. Wang. "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links". *In: Proc. of ACM Mobicom*, pages 287–297, July 2001.
4. E.N.Gilbert. "Capacity of a Burst-Noise Channel". *In: The Bell System Technical Journal*, September 1960.
5. V. Jacobson. "Congestion avoidance and control". *Proc. of ACM SIGCOMM' 88*, August 1998.
6. Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh-Cheng Chen. "A Survey of Energy Efficient Network Protocols for Wireless Networks". *In: Journal of Wireless Networks*, 7(4):343–358, 2001.
7. Y. Tian K. Xu and N. Ansari. "TCP-Jersey for Wireless IP Communications". *IEEE Journal on Selected Areas in Communications*, vol. 22, No. 4:747–756, May 2004.
8. V. Paxson M. Allman and W. Stevens. "TCP Congestion Control", RFC 2581, April 1999.
9. S. Floyd M. Mathis, J. Mahdavi and A. Romanow. "TCP Selective Aknowledgment Options", IETF RFC 2018, 1996.
10. L. Mamatas and V. Tsaoussidis. "Protocol Behavior: More Effort, More Gains?, PIMRC 2004", September 2004, Barcelona, Spain.
11. Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. In *ACM Mobicom 2001*, 2001.
12. H. Singh and S. Singh. "Energy Consumption of TCP Reno, Newreno, and SACK in Multi-Hop Networks". In *ACM SIGMETRICS 2002*, 2002.
13. V. Tsaoussidis and H. Badr. "TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains". *In: Proc. of the 8th IEEE Conference on Network Protocols ICNP*, 2000.
14. V. Tsaoussidis, H. Badr, X. Ge, and K. Pentikousis. "Energy / Throughput Tradeoffs of TCP Error Control Strategies". *In: Proc. of the 5th IEEE Symposium on Computers and Communications (ISCC)*, 2000.
15. V. Tsaoussidis and A. Lahanas. "Exploiting the adaptive properties of a probing device for TCP in heterogeneous networks". *In: Computer Communications, Elsevier Science*, 2:40, Nov. 2002.
16. V. Tsaoussidis and I. Matta. "Open Issues on TCP for Mobile Computing". *The Journal of Wireless Communications and Mobile Computing*, February 2002.
17. V. Tsaoussidis and C. Zhang. "The dynamics of responsiveness and smoothness in heterogeneous networks". *In: IEEE Journal on Selected Areas in Communications (JSAC), Special issue on Mobile Computing and Networking*, Match 2005.
18. M. Zorzi and R. Rao. "Energy Efficiency of TCP in a Local wireless Environment". *Mobile Networks and Applications*, 6, Issue 3, ISSN:1383-469X:265–278, 2001.

STC-Based Cooperative Relaying System with Adaptive Power Allocation*

Jingmei Zhang, Ying Wang, and Ping Zhang

WTI Labs, Beijing University of Posts and Telecommunications, P.R. China
zhang_jingmei@sohu.com

Abstract. Cooperative relaying recently has emerged as a means of providing gains from spatial diversity to devices in a distributed manner. A cooperative relaying system deploying Alamouti's space-time coding (STC) design is investigated in this paper. According to amplify-and-forward (AF) and decode-and-forward (DF) modes, two TDMA-based cooperative transmission schemes are presented. Considering resource utilization efficiency, adaptive power allocation (PA) algorithms are proposed to adjust the power of each hop based on different channel conditions. Most importantly, the PA results also can be used to decide whether or not to relay, which recovers the loss of spectral efficiency due to the orthogonal transmission to a great extent. Numerical results indicate that the cooperative system with adaptive PA significantly outperforms the direct transmission system. Compared with the uniform power allocation (UPA), the proposed PA algorithm with power constraint of 1W can provide (52, 54)% capacity gains at most for Scheme (I, II), respectively.

1 Introduction

The next generation wireless systems are supposed to have an intense requirement for the very ambitious throughput and coverage, as well as the power and bandwidth efficiency. Transmission over wireless channels suffers from random fluctuations known as fading and from co-channel interference. Diversity is a powerful technique to mitigate fading and improve robustness to interference. Spatial diversity techniques are particularly attractive since they provide diversity gain without incurring an expenditure of transmission time or bandwidth. It has been indicated that Multiple-Input-Multiple-Output (MIMO) systems can combat the effects of fading in wireless system and provide better spatial diversity and higher system capacity [1][2].

In a different context, relaying is often regarded as a means of improving the performance of infrastructure-based networks by increasing their coverage [3]. However, the reduced end-to-end path loss comes at the cost of an inherent rate increase and the repetition-coded nature of relaying systems. Yet, relaying is a viable option for infrastructure-based networks, and it is a basic means for service provisioning in mobile ad-hoc networks. In addition, the integration of the cellular

* This paper is supported by NSFC (No. 60302024, 60496312).

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 343–353, 2005.

networks and the Wireless Local Area Networks (WLANs) has drawn considerable attention from the research and commercial communities, which can both enhance the capacity of the cellular systems and extend the coverage area of 802.11 terminals [4].

Cooperative relaying brings together the worlds of MIMO and relaying systems. By allowing multiple users to cooperate and share their antennas effectively, virtual antenna arrays [5][6] can be built to realize spatial diversity gain in a distributed manner, which overcomes the size constraint of mobile terminal and some drawbacks of conventional relaying due to repetition coding. In [5][7], it is also shown that for channels with multiple relays, cooperative diversity with appropriately designed codes, such as space-time coding (STC), realizes full spatial diversity gain.

In this paper, the cooperative system adopting Alamouti's STC design is extended for a multi-antenna system based on multi-relay cooperation. Two TDMA-based cooperative transmission schemes are specialized for amplify-and-forward (AF) and decode-and-forward (DF) modes, respectively. Taking account of resource utilization efficiency, the adaptive power allocation (PA) algorithm, which usually remains to discuss or is replaced by the uniform power allocation (UPA) algorithm, is provided to enhance the system performance. The end-to-end achievable rate of the proposed system is investigated and compared to that of the conventional multi-antenna system.

2 System Model

The cooperative system analyzed in this paper is shown in Fig.1, which uses two relay terminals, R_1 and R_2, to relay the information transmitted by a source terminal S, to the

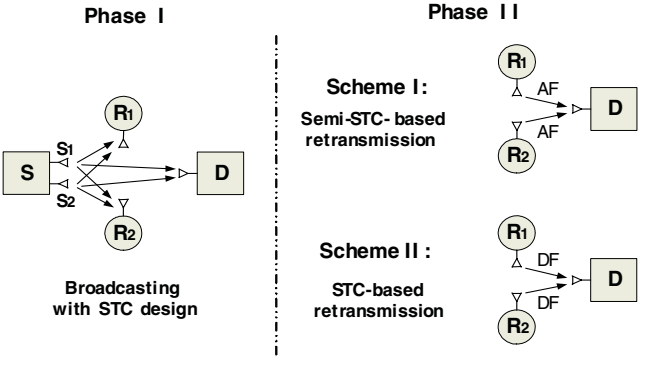


Fig. 1. Cooperative Relaying System and Two TDMA-based Transmission Schemes

destination terminal D. The terminal S is equipped with two antennas denoted by S_1 and S_2, and the other terminals are each with single antenna. Time division multiple access (TDMA) is adopted assuming a terminal cannot transmit and receive simultaneously. An Alamouti's STC design is employed in the source terminal S, and the relay terminal,

R_j (j=1,2) assists in communication with the destination terminal D by either AF or DF mode. In the AF mode, R_j simply amplifies and retransmits the signal received from S, which is corrupted by fading and additive noise. In the DF mode, the signal received from S is demodulated and decoded before retransmission. In this paper, it is assumed that the transmissions suffer from the effects of frequency-flat fading, no channel knowledge in the transmitters, perfect channel state information in the receivers and perfect synchronization. Based on Alamouti's STC scheme, at a given symbol period, two symbols, x_1 and x_2, are simultaneously broadcasted by terminal S from its two antennas. The symbol transmitted from antenna S_1 is denoted by x_1 and from antenna S_2 by $(-x_2^*)$. During the next symbol period, symbol x_2 and x_1^* are transmitted from antenna S_1 and S_2 respectively, where * is the complex conjugate operation, and the symbol energy of x_1 and x_2 are both unit 1. Assuming the effects of the transmission attenuation and multi-path fading between the transmitter and the receiver are constant during two adjacent symbols, then during two consecutive symbols (t=1,2), the signals received by the terminal D, $y_{SD}^{(t)}$, are

$$y_{SD}^{(1)} = \sqrt{P_S}\,(h_{S1D}x_1 - h_{S2D}x_2^*) + n_{SD}^{(1)}, \quad y_{SD}^{(2)} = \sqrt{P_S}\,(h_{S1D}x_2 + h_{S2D}x_1^*) + n_{SD}^{(2)} \tag{1}$$

where P_S is the transmit power at antenna S_i (i=1,2), h_{SiD} captures the path loss and multi-path fading between the source transmit antenna S_i and the destination terminal D, $n_{SD}^{(t)}$ is the additive white noise, which is zero-mean, independent identical distributed (i.i.d.) Gaussian random variable with variance σ_{SD}^2. The Alamouti's receiver [1] is used at the destination terminal D to process the received signals, then, the following estimated symbols, $\tilde{x}_{1,SD}$ and $\tilde{x}_{2,SD}$, can be built for the direct link.

$$\tilde{x}_{1,SD} = \sqrt{P_S}\,(h_{S1D}^* y_{SD}^{(1)} + h_{S2D} y_{SD}^{(2)^*}), \quad \tilde{x}_{2,SD} = \sqrt{P_S}\,(-h_{S2D} y_{SD}^{(1)^*} + h_{S1D}^* y_{SD}^{(2)}) \tag{2}$$

With transmission attenuation and fading realizations, h_{SiD}, the signal to noise ratio (SNR) of direct link can be expressed as [1]

$$\gamma_{SD} = (\alpha_{S1D} + \alpha_{S2D})P_S / \sigma_{SD}^2 = \beta_0 P_S / \sigma_{SD}^2 \tag{3}$$

where $\alpha_{SiD} = h_{SiD}h_{SiD}^*$ (i=1,2). Given W as the available bandwidth, the end-to-end achievable rate for direct transmission in terms of bps/Hz can be obtained as

$$C^{(D)}/W = \log_2(1+\gamma_{SD}) \tag{4}$$

Similarly, for the relay terminals, the signals received at R_j (j=1,2), $y_{SRj}^{(t)}$, are

$$y_{SRj}^{(1)} = \sqrt{P_S}\,(h_{S1Rj}x_1 - h_{S2Rj}x_2^*) + n_{SRj}^{(1)}, \quad y_{SRj}^{(2)} = \sqrt{P_S}\,(h_{S1Rj}x_2 + h_{S2Rj}x_1^*) + n_{SRj}^{(2)} \tag{5}$$

where h_{SiRj} (i=1,2; j=1,2) captures the path loss and multi-path fading between S_i and R_j, $n_{SRj}^{(t)}$ is a zero-mean, i.i.d. Gaussian random variable with variance σ_{SRj}^2. After the signal from S is received at R_j, it is processed and forwarded to D by R_j with either AF or DF mode. According to different relaying methods, two cooperative schemes are introduced and the associated signal models are also discussed in Sec. 3.

3 Cooperative Schemes and Performance Analyses

Fig. 1 describes two TDMA-based cooperative schemes, which employ different types of processing by the relay terminals. The transmission consists of two phases. In phase I, S broadcasts information to R_1, R_2 and D with Alamouti's STC. In phase II, for Scheme I, S keeps silent, while R_1 and R_2 communicate with D simultaneously using AF mode. However, since in AF mode, the received signal, which is transmitted by STC design, is only amplified and repeated, this scheme can be regarded as semi-STC-based retransmission. Scheme II operates in similar fashion to Scheme I, except that R_1 and R_2 decode, re-encode, and retransmit using a suitable STC. This scheme can be considered as STC-based retransmission. For both schemes, the destination terminal D combines the signals received in the previous two phases.

3.1 Scheme I (AF Mode)

In this scheme, the relay terminal R_j first normalizes the received signal, and then retransmits it to D with power P_{Rj}. This process can be regarded as amplifying signals with the amplification factor $G_{Rj}^2 = P_{Rj} / [(\alpha_{S1Rj} + \alpha_{S2Rj})P_S + \sigma_{SRj}^2]$, where $\alpha_{SiRj} = h_{SiRj} h_{SiRj}^*$. Using (5), the signals $y_{RD}^{(t)}$ (t=1,2) received at D in phase II are

$$\begin{aligned} y_{RD}^{(1)} &= h_{R1D} G_{R1} y_{SR1}^{(1)} + h_{R2D} G_{R2} y_{SR2}^{(1)} + n_{RD}^{(1)} = H_1 x_1 - H_2 x_2^* + N_1 \\ y_{RD}^{(2)} &= h_{R1D} G_{R1} y_{SR1}^{(2)} + h_{R2D} G_{R2} y_{SR2}^{(2)} + n_{RD}^{(2)} = H_1 x_2 + H_2 x_1^* + N_2 \end{aligned} \tag{6}$$

with

$$\begin{aligned} H_i &= \sqrt{P_S}\left(h_{SiR1} h_{R1D} G_{R1} + h_{SiR2} h_{R2D} G_{R2}\right) \\ N_t &= h_{R1D} G_{R1} n_{SR1}^{(t)} + h_{R2D} G_{R2} n_{SR2}^{(t)} + n_{RD}^{(t)} \end{aligned} \quad (i=1,2\ ;\ t=1,2) \tag{7}$$

where h_{RjD} (j=1,2) captures the path loss and multi-path fading between R_j and D, $n_{RD}^{(t)}$ is a zero-mean, i.i.d. Gaussian random variable with variance σ_{RD}^2. The estimated symbols, $\tilde{x}_{1,RD}$ and $\tilde{x}_{2,RD}$, for the relaying link can be obtained as

$$\tilde{x}_{1,RD} = H_1^* y_{RD}^{(1)} + H_2 y_{RD}^{(2)^*}, \quad \tilde{x}_{2,RD} = -H_2 y_{RD}^{(1)^*} + H_1^* y_{RD}^{(2)} \tag{8}$$

Assuming the noises at different receivers are uncorrelated, then $N_1 N_1^* = N_2 N_2^*$, and the SNR of the estimated signal in (8), $\gamma_{RD}^{(I)}$, can be written as

$$\gamma_{RD}^{(I)} = \frac{H_1 H_1^* + H_2 H_2^*}{N_1 N_1^*} = \frac{P_S(\beta_1 G_{R1}^2 + \beta_2 G_{R2}^2 + \beta_3 G_{R1} G_{R2})}{(\sigma_{RD}^2 + \alpha_{R1D} G_{R1}^2 \sigma_{SR1}^2 + \alpha_{R2D} G_{R2}^2 \sigma_{SR2}^2)} \tag{9}$$

where

$$\begin{aligned} &\beta_1 = (\alpha_{S1R1} + \alpha_{S2R1})\alpha_{R1D}\ ,\ \ \beta_2 = (\alpha_{S1R2} + \alpha_{S2R2})\alpha_{R2D} \\ &\beta_3 = (h_{S1R1}^* h_{S1R2} + h_{S2R1}^* h_{S2R2}) h_{R1D}^* h_{R2D} + (h_{S1R1}^* h_{S1R2} + h_{S2R1}^* h_{S2R2})^* h_{R1D} h_{R2D}^* \end{aligned} \tag{10}$$

Combines the signals received in two phases with the MRC receiver, the end-to-end achievable rate for Scheme I in terms of bps/Hz can be derived from (3) and (9) as

$$C^{(\mathrm{I})}/W = \frac{1}{2}\left[\log_2\left(1+\gamma_{\mathrm{SD}}+\gamma_{\mathrm{RD}}^{(\mathrm{I})}\right)\right] \tag{11}$$

where the factor 1/2 accounts for the fact that information is conveyed to the destination terminal over two phases.

3.2 Scheme II (DF Mode)

In this scheme, DF mode is adopted in phase II. Before forwarding, two relays detect the received signal with Alamouti's decoder [1]. According to (5), the estimated symbols, $\tilde{x}_{1,\mathrm{SR}}$ and $\tilde{x}_{2,\mathrm{SR}}$, for the 1st hop are as follows.

$$\begin{aligned}\tilde{x}_{1,\mathrm{SR}} &= h_{\mathrm{S1R1}}^{*} y_{\mathrm{SR1}}^{(1)} + h_{\mathrm{S2R1}} y_{\mathrm{SR1}}^{(2)\,*} + h_{\mathrm{S1R2}}^{*} y_{\mathrm{SR2}}^{(1)} + h_{\mathrm{S2R2}} y_{\mathrm{SR2}}^{(2)\,*} \\ \tilde{x}_{2,\mathrm{SR}} &= -h_{\mathrm{S2R1}} y_{\mathrm{SR1}}^{(1)\,*} + h_{\mathrm{S1R1}}^{*} y_{\mathrm{SR1}}^{(2)} - h_{\mathrm{S2R2}} y_{\mathrm{SR2}}^{(1)\,*} + h_{\mathrm{S1R2}}^{*} y_{\mathrm{SR2}}^{(2)}\end{aligned} \tag{12}$$

With (5) and (12), the 1st-hop SNR for Scheme II can be written as

$$\gamma_{\mathrm{SR}} = \frac{(\alpha_{\mathrm{S1R1}}+\alpha_{\mathrm{S2R1}}+\alpha_{\mathrm{S1R2}}+\alpha_{\mathrm{S2R2}})^2 P_{\mathrm{S}}}{(\alpha_{\mathrm{S1R1}}+\alpha_{\mathrm{S2R1}})\sigma_{\mathrm{SR1}}^2 + (\alpha_{\mathrm{S1R2}}+\alpha_{\mathrm{S2R2}})\sigma_{\mathrm{SR2}}^2} \tag{13}$$

Assuming that the received signal is decoded correctly at the two relays, the estimated symbols, $\hat{x}_1$ and $\hat{x}_2$, are transmitted to D also with Alamouti's STC scheme. The transmit power at R_j is denoted as $P_{\mathrm{R}j}$. The symbol energy is unit 1. The destination terminal D receives signals from the two relays are

$$\begin{aligned} y_{\mathrm{RD}}^{(1)} &= h_{\mathrm{R1D}}\sqrt{P_{\mathrm{R1}}}\,\hat{x}_1 - h_{\mathrm{R2D}}\sqrt{P_{\mathrm{R2}}}\,\hat{x}_2^{*} + n_{\mathrm{RD}}^{(1)} \\ y_{\mathrm{RD}}^{(2)} &= h_{\mathrm{R1D}}\sqrt{P_{\mathrm{R1}}}\,\hat{x}_2 + h_{\mathrm{R2D}}\sqrt{P_{\mathrm{R2}}}\,\hat{x}_1^{*} + n_{\mathrm{RD}}^{(2)}\end{aligned} \tag{14}$$

Also using Alamouti's combiner at the terminal D, the 2nd-hop SNR for Scheme II is

$$\gamma_{\mathrm{RD}}^{(\mathrm{II})} = (\alpha_{\mathrm{R1D}}P_{\mathrm{R1}} + \alpha_{\mathrm{R2D}}P_{\mathrm{R2}})/\sigma_{\mathrm{RD}}^2 \tag{15}$$

Similarly, the destination D combines the signals received in the two phases with the MRC receiver. Requiring both the relays and destination to decode perfectly, the end-to-end achievable rate for Scheme II can be readily shown to be [5]

$$C^{(\mathrm{II})}/W = \frac{1}{2}\log_2\left\{\min\left[\left(1+\gamma_{\mathrm{SR}}\right),\left(1+\gamma_{\mathrm{SD}}+\gamma_{\mathrm{RD}}^{(\mathrm{II})}\right)\right]\right\} \tag{16}$$

From (4), (11) and (16), it can be seen that the price to be paid for the relaying transmission over two phases is a reduction in spectral efficiency accounted for by the factor 1/2 in front of the log term.

4 Adaptive Power Allocation

Although the idea of deploying multi-antenna techniques through cooperation can enhance the system performance, the UPA algorithm, which allocates equal power to

each hop in the network, does not utilize the system resources effectively [8][9]. So the adaptive PA algorithms, which adjust the power of each hop based on different channel conditions, are proposed in this section for different relaying schemes.

In order to provide a fair comparison, it is crucial that the total consumed energy of the cooperative relaying system does not exceed that of the corresponding direct system. In the conventional direct transmission system, the source terminal S transmits signals with total power $2P_S=P_0$ over a period T. Its consumed energy is P_0T. For the relaying transmission, the source terminal S first broadcasts information with power of $2P_S$ over a period of T/2. Assuming two relays transmit with equal power, i.e. $P_{R1}=P_{R2}=P_R$, the total power over the next T/2 is $2P_R$. Then the consumed energy in the relaying system is $2P_S$(T/2)+ $2P_R$(T/2). Thus, the consumed energy and total transmit power should be normalized as follows.

$$P_0 = P_S + P_R = P^{(I)} = P^{(II)} \tag{17}$$

where $P^{(I)}$ and $P^{(II)}$ are the power constraint for Scheme I and II, respectively. For the proposed system, the achievable rate can be regarded as a function of P_S and P_R. Therefore the object of the PA algorithm is to characterize these two parameters to maximize the achievable rate under a certain power constraint.

4.1 Adaptive PA for Scheme I

Taking the achievable rate as the optimization criterion, with (11), the PA issue for Scheme I can be described as

$$\begin{aligned} &\max_{P_S,P_R}\{C^{(I)}\} = \frac{1}{2}\max_{P_S,P_R}\left\{\log_2(1+\gamma_{SD}+\gamma_{RD}^{(I)})\right\} \\ &s.t.\ \ P_S + P_R = P^{(I)} \ \ (0 < P_S \le P^{(I)}, 0 \le P_R < P^{(I)}) \end{aligned} \tag{18}$$

For simplicity, it is assumed that the noises at different receivers are with identical power σ_0^2. Using (3) and (9), after some elementary manipulations, (18) can be equivalent to maximize the function $F(P_S)$ with condition $0 < P_S \le P^{(I)}$, and

$$F(P_S) = \frac{P_S}{\sigma_0^2}\left[\beta_0 + \frac{\beta_1 f_2 + \beta_2 f_1 + \beta_3\sqrt{f_1 f_2}}{f_1 f_2 + (\alpha_{R_1D} f_2 + \alpha_{R_2D} f_1)(P^{(I)} - P_S)}(P^{(I)} - P_S)\right] \tag{19}$$

with

$$f_j = (\alpha_{S1Rj} + \alpha_{S2Rj})P_S + \sigma_0^2 \quad (j=1,2) \tag{20}$$

Applying Lagrange Multiplier, P_S can be obtained through a polynomial after some elementary manipulations. The coefficients are very complicated and not presented here. Comparing the achievable rates, which are derived from the boundary point $P_S=P^{(I)}$ (means direct transmission) and other values of P_S obtained from the above method (means relaying transmission), $P_S^{(I)}$ which maximizes the data rate and satisfies $0 < P_S^{(I)} \le P^{(I)}$ can be chosen as the optimal solution, and accordingly, $P_R^{(I)} = P^{(I)} - P_S^{(I)}$.

4.2 Adaptive PA for Scheme II

For cooperative Scheme II, also taking the achievable rate as the optimization criterion, with (16) the PA issue for Scheme II can be described as

$$\begin{aligned} &\max_{P_S,P_R}\left\{C^{(II)}\right\}=\frac{1}{2}\max_{P_S,P_R}\left\{\log_2\left\{\min\left[\left(1+\gamma_{SR}\right),\left(1+\gamma_{SD}+\gamma_{RD}^{(II)}\right)\right]\right\}\right\} \\ &s.t.\ \ P_S+P_R=P^{(II)}\ \ \left(0<P_S\le P^{(II)},0\le P_R<P^{(II)}\right) \end{aligned} \tag{21}$$

The possible solutions to such optimization issue have been discussed in [9]. Similarly, for (21) only when $\beta_{hop2}>\beta_0$, there may exist the optimal PA solution determined by $\gamma_{SR}=\gamma_{SD}+\gamma_{RD}^{(II)}$ for the relaying transmission, where $\beta_0=\alpha_{S1D}+\alpha_{S2D}$ and $\beta_{hop2}=\alpha_{R1D}+\alpha_{R2D}$. Otherwise, P_S=$P^{(II)}$ is the final PA result, i.e. the direct transmission is favorable. Thus the optimal PA solution to Scheme II is

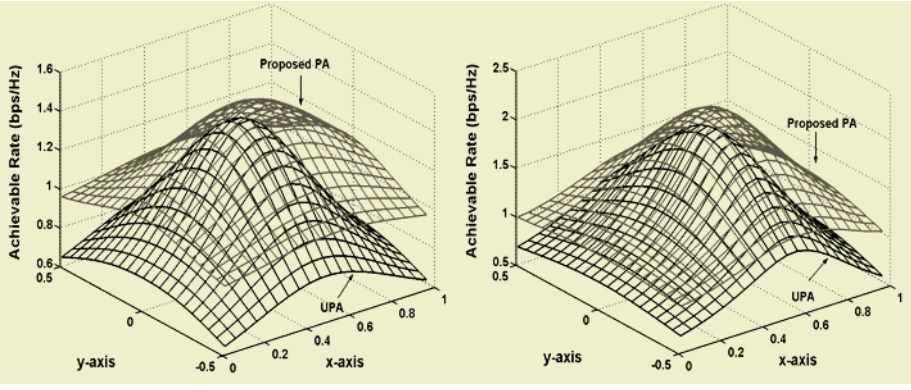
$$\begin{cases} P_S=\dfrac{\beta_{hop2}P^{(II)}}{\beta_{hop1}+\beta_{hop2}-\beta_0} \\ P_R=\dfrac{(\beta_{hop1}-\beta_0)P^{(II)}}{\beta_{hop1}+\beta_{hop2}-\beta_0} \end{cases};\quad \beta_{hop1}\ge\beta_0\ and\ \beta_{hop2}>\beta_0 \tag{22}$$

where $\beta_{hop1}=\alpha_{S1R1}+\alpha_{S2R1}+\alpha_{S1R2}+\alpha_{S2R2}$. Note that in (22), the constraint $\beta_{hop1}\ge\beta_0$ comes from the power constraint $0\le P_R<P^{(II)}$. From the above analyses, it can be concluded that if the conditions $\beta_{hop1}\ge\beta_0$ and $\beta_{hop2}>\beta_0$ are not satisfied, i.e. both channel conditions of the two hops are not better than that of the direct link, it may be beneficial to allocate all transmit power for direct communication rather than splitting power between the two hops.

Based on the discussions for the two schemes, it can be seen that the proposed PA algorithms not only can adjust the transmit power of each hop adaptively, but also can decide whether or not to relay grounded on the PA result, e.g. if P_R=0, it means that using the direct link only is superior to relay-assisted communication.

5 Numerical Results and Discussions

The performance of the cooperative system with two relaying transmission schemes and the efficacy of the proposed PA algorithms are assessed here with Monte Carlo simulation. The source terminal S and the destination terminal D are located at (0,0) and (1,0) respectively, and the two relays range from 0 to 1 along the x-axis and –0.5 to 0.5 along the y-axis. It is assumed that the two relays are spatially sufficiently close as to justify a common path loss; however, sufficiently far apart as to justify uncorrelated fading. Pass loss is given by d^{-4}, where d is the distance between the transmitter and receiver. The channel is assumed to obey the flat Rayleigh fading with an average power unity, and the noise power is normalized to unity.



(a) Scheme I (b) Scheme II

Fig. 2. Achievable Rate Comparison between UPA and Proposed PA Algorithm (P_0=1W)

Fig.2 shows the system achievable rate comparison between the UPA and the adaptive PA algorithm with P_0 =1W, where (a) is for Scheme I, and (b) for Scheme II. It shows that for both schemes, the system obtains the best performance when the relays are close to the midpoint between the source and destination terminals. As the distance with respect to that midpoint position increases, the performance degrades. Compared with the UPA, our proposed PA algorithm improves the achievable rate of the cooperative relaying system in the whole studied area.

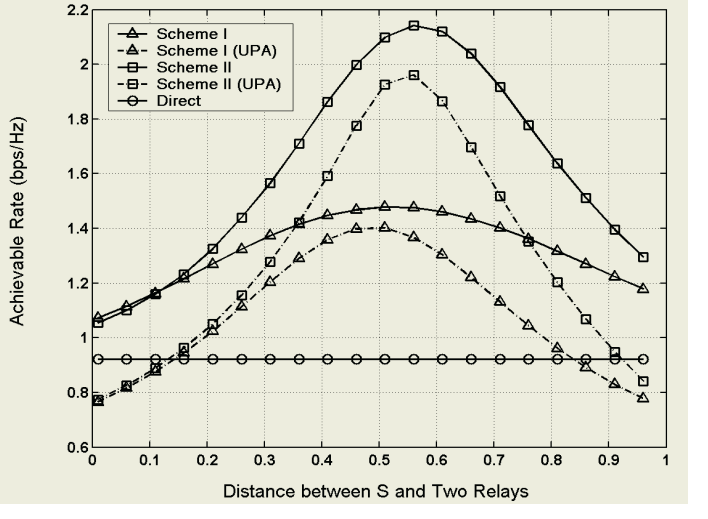


Fig. 3. Rate Comparison between Cooperative Relaying and Direct Transmission

In order to show the results more clearly, Fig.3 gives the rate comparison when the relays move along the line between S and D, where P_0 =1W. From Fig.3 it can be seen

that with the proposed PA algorithm, the cooperative relaying transmission significantly outperforms the conventional direct transmission, especially with Scheme II, since DF mode improves the performance in exchange of an increased complexity. However, for the UPA algorithm, when the relays are around the terminals S or D, the relaying transmission is even inferior to the direct transmission since UPA cannot switch the relaying to the direct transmission adaptively as the proposed PA algorithm. Compared with UPA, for Scheme I, the adaptive PA can provide the gains of (40, 52)% for the cases when the relays are around S and D, respectively. For Scheme II, the gains are (36, 54)% correspondingly. When the relays are around the midpoint of the terminals S and D, the gains decrease to (5, 9)% for Scheme (I, II), respectively on account of the similar PA solutions caused by the parallelism of the two-hop channel conditions.

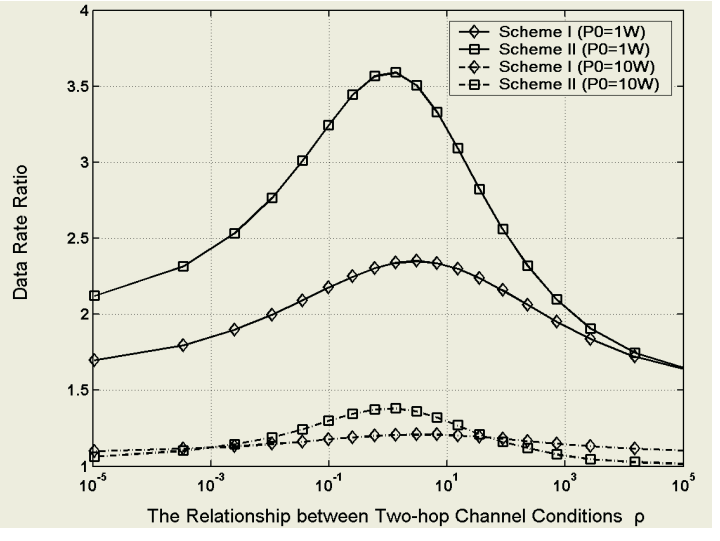


Fig. 4. Impact of the two-hop channel conditions on the performance of relaying system

Fig.4 depicts the data rate ratio of the relaying transmission to the direct transmission as a function of ρ, where $\rho = \beta_{\text{hop1}} / \beta_{\text{hop2}}$ represents the relationship between the two-hop channel conditions. The adaptive PA algorithm is adopted. For P_0=1W, it can be observed that the performance gain provided by the cooperative relaying is significant when $\rho \approx 1$, and if $\rho >> 1$ or $\rho << 1$, i.e. the two hops are badly unbalanced, the gain decreases, especially for Scheme II. Scheme I is less sensitive to ρ, since the AF mode responds gracefully to the channel condition on any individual link between two terminals, while the DF mode introduces decoding error if any hop produces error. On the other hand, it can be seen from (11) that there is not much immediate influence of ρ on the performance of Scheme I, which is different from

Scheme II (see (16)). At high power constraint (P_0=10W), the advantage of relaying transmission weakens since the cooperative diversity may not recover fully from the loss in spectral efficiency due to transmission over two phases. If it were not for the adaptive PA algorithm, direct communication would be even more attractive somewhere, which is testified in Fig.3 when UPA is used. In addition, it also can be seen that for high power, when $\rho >> 1$ or $\rho << 1$, Scheme I outperforms Scheme II since the latter may be limited with the deteriorated quality of channel at the worse hop which may becomes a bottleneck.

6 Conclusions

A cooperative relaying system is studied in this paper, where Alamouti's STC design is extended for a multi-antenna environment based on multi-relay cooperation. According to different relaying methods, AF or DF mode, two TDMA-based cooperative transmission schemes are presented and their performances are also analyzed. Considering the resource utilization efficiency, the adaptive PA algorithms are proposed to adjust the power of each hop with the achievable rate as the optimization criterion. Most importantly, the proposed PA algorithms can adaptively switch relaying to direct transmission based on PA results, which recovers the loss of spectral efficiency due to the orthogonal transmission to a great extent. The numerical results indicate that with low power constraint (P_0=1W), the cooperative system with adaptive PA algorithm significantly outperforms the direct transmission system, whose achievable rate can be improved to 2~3 times than before averagely by Scheme I and II respectively. With high power (P_0=10W), the superiority of the relaying transmission weakens, however, there are still performance gains if the adaptive PA algorithm is utilized. It can be concluded that with efficient PA algorithm and appropriate relaying selection criterion, the idea of applying cooperative schemes into effective point-to-point relaying channels can be easily extended to larger networks and more complex transmission environments.

References

[1] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications", *IEEE Journal on selected areas in communications*, Vol. 16, No. 8, Oct. 1998.

[2] E. Telatar, "Capacity of Multi-antenna Gaussian Channels," AT&T Bell Laboratories, Tech. Memo. June 1995.

[3] Pabst R., Walke B. H. and et al., "Relay-based Deployment Concepts for Wireless and Mobile Broadband Radio," *IEEE Communications Magazine*, vol. 42, pp. 80-89, Sept. 2004.

[4] Hung-yu Wei, and Richard D. Gitlin, "WWAN/WLAN Two-Hop-Relay Architecture for Capacity Enhancement," *IEEE Wireless Communications and Networking Conference (WCNC 04')*, March 2004.

[5] J. Nicholas Laneman, "Cooperative Diversity in Wireless Networks: Algorithms and Architectures," M.I.T. Doctoral Dissertation, Sep. 2002.

[6] Mischa Dohler, E. Lefranc, H. Aghvami, "Virtual Antenna Arrays for Future Wireless Mobile Communication Systems", *IEEE ICT 2002*, Beijing, China, 2002.
[7] R. U. Nabar and H. Bolcskei, "Space-time Signal Design for Fading Relay Channels," in *Proc. IEEE GLOBECOM*, San Francisco, CA, Dec. 2003.
[8] Zhang Jingmei, Zhang Qi, et. al, "Adaptive Optimal Transmit Power Allocation for Two-hop Non-regenerative Wireless Relaying System", *IEEE VTC'04 Spring*, 2004.
[9] Zhang Qi, Zhang Jingmei, et al. "Power Allocation for Regenerative Relay Channel with Rayleigh Fading", *IEEE VTC'04 Spring*, 2004.

Reducing Memory Fragmentation with Performance-Optimized Dynamic Memory Allocators in Network Applications

Stylianos Mamagkakis[1], Christos Baloukas[1], David Atienza[2], Francky Catthoor[3], Dimitrios Soudris[1], José Manuel Mendias[2], and Antonios Thanailakis[1]

[1] VLSI Design Center-Democritus University of Thrace, 67100 Xanthi, Greece
{smamagka, cmpalouk, dsoudris, thanail}@ee.duth.gr
[2] DACYA - Univ. Complutense de Madrid, 28040 Madrid, Spain
{datienza, mendias}@dacya.ucm.es
[3] IMEC vzw, Kapeldreef 75, 3001 Heverlee, Belgium,
Also professor at K.U.Leuven, Belgium
catthoor@imec.be

Abstract. The needs for run-time data storage in modern wired and wireless network applications are increasing. Additionally, the nature of these applications is very dynamic, resulting in heavy reliance to dynamic memory allocation. The most significant problem in dynamic memory allocation is fragmentation, which can cause the system to run out of memory and crash, if it is left unchecked. The available dynamic memory allocation solutions are provided by the real time Operating Systems used in embedded or general-purpose systems. These state-of-the-art dynamic memory allocators are designed to satisfy the run-time memory requests of a wide range of applications. Contrary to most applications, network applications need to allocate too many different memory sizes (e.g. hundreds different sizes for packets) and have an extremely dynamic allocation and de-allocation behavior (e.g. unpredictable web-browsing activity). Therefore, the performance and the de-fragmentation efficiency of these allocators is limited. In this paper, we analyze all the important issues of fragmentation and the ways to reduce it in network applications, while keeping the performance of the dynamic memory allocator unaffected or even improving it. We propose highly customized dynamic memory allocators, which can be configured for specific network needs. We assess the effectiveness of the proposed approach in two representative real-life case studies of wired and wireless network applications. Finally, we show very significant reduction in memory fragmentation and increase in performance compared to state-of-the-art dynamic memory allocators utilized by real-time Operating Systems.

1 Introduction

In the last years networks have become ubiquitous. Modern portable devices are expected to access the internet (e.g. 3G mobile phones) and communicate

T. Braun et al. (Eds.): WWIC 2005, LNCS 3510, pp. 354–364, 2005.

with each other wirelessly (e.g. PDAs with 802.11b/g) or with a wired connection (e.g. Ethernet). In order to provide the desired Quality of Experience to the user, these systems have to respond to the dynamic changes of the environment (i.e. network traffic) and the actions of the user as fast as possible. Additionally, they need to provide the necessary memory space for the network applications dynamically at run-time. Therefore, they have to rely on dynamic memory allocation mechanisms to satisfy their run-time data storage needs. Inefficient dynamic memory (DM from now on) allocation support leads to decreased system performance and increased cost in memory footprint due to fragmentation [1].

The standard DM allocation solutions for the applications inside the Terminals, Routers or Access Points are activated with the standardized malloc/free functions in C and the new/delete operators in C++. Support for them is provided by Real Time Operating Systems (e.g. uClinux [8]). These O.S. based DM allocators are designed for a variety of applications and thus can not address the specific memory allocation needs of network applications. This results in mediocre performance and increased fragmentation. Therefore, custom DM allocators are needed [7, 12] to achieve better results. Note that they are still realized in the middleware and usually not in the hardware. In our case we propose never to use hardware but instead use only a library (system layer) just on top of the (RT)OS in the middleware.

In this paper, we propose a systematic approach to reduce memory fragmentation (up to 97%) and increase performance (up to 97%), by customizing a DM allocator to be used especially for the network application domain. The major contribution of our work is that we explore exhaustively all the available combinations of de-fragmentation techniques and explain how our custom DM allocator can decrease fragmentation and improve performance at the same time in network applications. The remainder of the paper is organized as follows. In Sect. 2, we describe some related work. In Sect. 3, we analyze fragmentation. In Sect. 4, we show the de-fragmentation techniques and their trade-offs. In Sect. 5, we describe our exploration and explain the effect of each de-fragmentation technique in the network application domain. In Sect. 6 we present the simulation results of our case studies. Finally, in Sect. 7 we draw our conclusions.

2 Related Work

Currently, there are many O.S. based, general-purpose DM allocators available. Successful examples include the Lea allocator in Linux based systems [5], the Buddy allocator for Unix based systems [5] and variations of the Kingsley allocator in Windows XP [11] and FreeBSD based systems. Their embedded O.S. counterparts include the DM allocators of Symbian, Enea OSE [9], uClinux [8] and Windows CE [10]. Other standardized DM allocation solutions

are evaluated in [6] for a wide range of applications (without evaluating performance). In contrast to these 'off-the-shelf' DM allocation solutions, our approach provides highly customized DM allocators, fine tuned to the networking applications for both low memory fragmentation and high performance.

Also, in [12], the abstraction level of customizable memory allocators has been extended to C++. Additionally, the authors of [7] propose an infrastructure of C++ layers that can be used to improve performance of general-purpose allocators. Finally, work has been done to propose several garbage collection algorithms with relatively limited performance overhead [13]. Contrary to these frameworks, which are limited in flexibility, our approach is systematic and is linked with our tools [2], which automate the process of custom DM allocator construction. This enables us to explore and validate the efficiency of our customized DM allocators, combining both memory de-fragmentation and performance metrics.

3 Memory Fragmentation

When the application requests a memory block from the DM allocator, which is smaller than the memory blocks available to the allocator, then a bigger block is selected from the memory pool and allocated. This results in wasted memory space inside the allocated memory block. This is called internal fragmentation, which is common in requests of small memory blocks [5]. When the application requests a memory block from the DM allocator, which is bigger than the memory blocks available to the allocator, then these smaller memory blocks are not selected for the allocation (because they are not contiguous) and become unused 'holes' in memory. These 'holes' among the used blocks in the memory pool are called external fragmentation.

We measure the level of both internal and external fragmentation (we use the same cost function with [6]). Thus, we express fragmentation in terms of percentages over and above the amount of live data, (i.e. increase in memory usage), not the percentage of actual memory usage that is due to fragmentation. Therefore, we measure the maximum amount of memory requested by the application relative to the maximum amount of memory used by the DM allocator:

$$Fragmentation = \frac{Memory_{alloc.}}{Memory_{req.}} - 1$$

$$Memory_{alloc.} = Memory_{req.} + Memory_{Int.Fragm.} + Memory_{Ext.Fragm.}$$

4 Memory De-Fragmentation Techniques and Trade-Offs

We are going to analyze the de-fragmentation techniques and their trade-offs. All of the techniques are well known [5] but their trade-offs (when used in conjunction) have never been evaluated up to now:

1.-The most common technique to prevent internal memory fragmentation is the use of *freelists*. The *freelists* are lists (i.e. double or single linked lists) of memory blocks, which were no longer needed by the application and, thus, they were freed by the DM allocator. This technique can reduce internal fragmentation significantly and improve performance in most cases. The trade-off is that it increases external fragmentation, because the freed blocks are not returned in the main memory pool, where they can be coalesced with a neighboring free block to produce a bigger contiguous memory space.

2.-Another technique to prevent internal memory fragmentation is the use of specific fit policies. The two most popular fit policies are the *first fit policy* and the *best fit policy*. On the one hand, the *first fit policy* allocates the first memory block that it finds that is bigger than the requested block. On the other hand, the *best fit policy* searches a part (or even 100%) of the memory pool in order to find the memory block closest to the size of the requested block. Therefore, there will be the least memory overhead per block and, thus, the least internal fragmentation. The trade-off is that the performance of the DM allocator decreases, while it spends more time trying to find the best fit for the requested block.

3.-An additional technique to decrease internal fragmentation is the use of the *splitting mechanism*. When the DM allocator finds a block bigger than the requested block, then it can split it in two. The block can be split precisely to fit the request and, thus, produce zero internal fragmentation. The trade-off of this mechanism is that it reduces performance considerably. The mechanism itself needs a lot of time perform the splitting, plus it generates one more block inside the pool per split.

4.-Finally, a technique to decrease external fragmentation is the use of the *coalescing mechanism*. When the DM allocator frees a block, which has an adjacent memory address with another free memory block, then it can coalesce them to produce a single bigger block. In this way, external memory fragmentation can be reduced significantly. A positive by-product of the *coalescing mechanism* is that it results in one less block inside the pool per coalesce. This in turn reduces significantly the time needed to traverse all the blocks inside the pool to find a best or first fit. On the other hand, the trade-off of this mechanism is that it reduces some performance, because the mechanism itself needs some time to perform the coalescing.

It is obvious that these four different de-fragmentation techniques have contradicting effects on performance, internal and external fragmentation (e.g. an increase of usage of the *splitting mechanism* decreases internal fragmentation but also decreases performance). To make things even more complicated it appears that the efficiency of the techniques is interdependent (e.g. the performance of the *best fit policy* decreases when the usage of the *splitting mechanism* increases). So a Pareto trade-off exploration is necessary. In order to evaluate which techniques should be used to decrease fragmentation and how much they should be applied, we have explored exhaustively all the available combinations of de-fragmentation techniques in various levels of usage (ranging from full usage to no usage of the technique at all).

5 Customization of DM Allocators for Network Applications

For the purposes of the exhaustive exploration of the different de-fragmentation techniques we have used our powerful profiling tool (described in more detail in [2]). Our tool automates the process of building, implementing, simulating and profiling different customized DM allocators. Every one of these customized DM allocators implements a different combination of de-fragmentation techniques with a different combination of usage level for each technique. About 10 levels of usage have been used for each de-fragmentation technique. The total exploration effort took 45 days using 2 Pentium IV workstations. On average, there have been explored about 10.000 different customized DM allocator implementations for each one of two different networking applications: DRR scheduling and buffering in Easyport. Finally, 3 to 7 real network traffic trace inputs (of wired and wireless networks) have been used for each application to make sure that our exploration strategy is valid for a wide range of dynamic behavior scenarios.

In Fig. 1, a custom DM allocation exploration example for the Easyport buffering application can be seen (a network traffic trace of various real ftp sessions was used as input). Each dot in the figure is the simulation results for performance and memory footprint allocated by one out of the 10.000 explored custom DM allocators. The results were heavily pruned and (out of the 10.000 custom DM allocator implementations) only a handful with the best performance and lowest fragmentation were selected (as seen in the upper right corner of Fig. 1). The same procedure has been used for the other applications and for each one of the available inputs (i.e. network traffic traces).

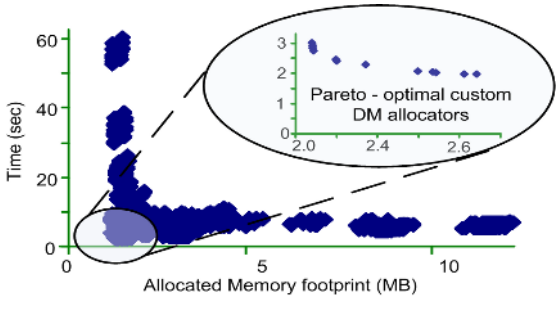


Fig. 1. Custom DM allocation exploration example for the Easyport buffering application and pareto-optimal DM allocators

Our simulations show that the limited list of resulting 'Pareto-optimal' custom DM allocators share some common characteristics, which favor particular de-fragmentation techniques at certain levels of usage:

1.-Contrary to most application domains (where about 6 different memory sizes amount for more than 90% of the total requested memory sizes [6]), in networking applications just 2 memory sizes amount for 30-70% of the total

requested memory sizes (an example of this bimodal distribution can be seen in the histograms of Fig. 3). These 2 object sizes are around the size of the *Acknowledgement* (or ACK) packet and the *Maximum Transmission Unit* (or MTU) packet of each network [3]. The rest of the requested memory sizes are evenly distributed between these 2 extreme sizes. Our exploration results show that custom DM allocators, with just 2 freelists of these 2 extreme memory sizes, managed to reduce considerably internal fragmentation and improve performance, without increasing much the external fragmentation. All five of the O.S. based DM allocators, which use from 6-64 different freelists, manage to do the same, but with a very high cost in external fragmentation.

2.-Contrary to most application domains (where memory usage comes in the form of very thin spikes and 10% of the memory sizes are freed back to the main memory heap or pool [5] [6]), in networking applications the memory usage form varies greatly [3] (in the upper 3 traces of Fig. 2 we can see thin and fat spikes, in the lower left trace of Fig. 2 we can see plateaus and in the lower right trace of Fig. 2 we can see a ramp). Additionally, about 30-70% of the memory sizes are returned to the main memory pool. This means that blocks are not always freed fast (this is the case of thin spike usage forms only) and that the main memory pool accommodates a huge number of memory blocks. It also means that the *best fit policy* used in all the O.S. based DM allocators (except Windows XP and CE) is extremely slow because it has to traverse too many blocks in order to find a good fit. Our exploration results show that custom DM allocators, which use *first fit policy* in combination with full usage of the *splitting mechanism* and the *coalescing mechanism*, increase performance dramatically and suffer only minimal internal fragmentation overhead.

3.-Contrary to most application domains (where about 38 different memory sizes constitute 99% of the total requested memory sizes [6]), in networking applications 30-70% of the total requested memory sizes are attributed to 700-1500 different memory sizes (an example of this fact can be seen in Fig. 3). This produces exceptionally high values of internal fragmentation, which is different from what is observed in other application domains. All the O.S. based DM allocators (except Linux) have a very low usage level of the *splitting mechanism* and therefore suffer massively from internal fragmentation. Actually, our exploration showed that this is the major contributor to fragmentation generally in network applications. Our exploration results show us that the only way to really decrease fragmentation is with the full use of the *splitting mechanism*.

4.-Finally, a common characteristic shared among the networking and the other application domains, is that objects allocated at the same time tend to die and get de-allocated at the same time. This temporal locality of the allocated objects is something common in both wired and wireless networks. The reason is that the traffic structure is imposed implicitly by the tasks initiated by Internet users at the application layer (e.g. a file or a Web page download). Therefore, allocated objects are not independent and isolated entities; rather they are part of a higher-layer logical flow of information [3]. This temporal locality can easily be converted to spatial locality of the memory freed, if we pursue high usage

levels of the *coalescing mechanism*, thus reducing external fragmentation. All the O.S. based DM allocators (except Linux) have an extremely low usage level of the *coalescing mechanism* and thus can not take advantage of the locality effect. Our exploration results have shown, that with full usage of the *coalescing mechanism*, external fragmentation in networking applications can be eradicated.

These favorable common characteristics are a combination of just two *freelists*, *first fit policy*, full usage of the *splitting mechanism* and full usage of the *coalescing mechanism*. Therefore, this is the custom DM allocator that we propose to use for network applications.

6 Case Studies and Simulation Results

We have applied the proposed custom DM allocator to two real case studies:

The first case study, is the Deficit Round Robin (or DRR) [14] application, which is a scheduling algorithm implemented in many routers and WLAN Access Points today [15]. In the DRR algorithm, the scheduler visits each internal non-empty queue, increments the variable deficit by the value *Quantum* (e.g. 9 Kbytes are used in most Cisco Routers) and determines the number of bytes in the packet at the head of the queue. If the variable deficit is less than the size of the packet at the head of the queue (it does not have enough credits), then the scheduler moves on to service the next queue. If the size of the packet at the head of the queue is less than or equal to the variable deficit, then the variable deficit is reduced by the number of bytes in the packet and the packet is transmitted on the output port. The scheduler continues this process, starting from the first queue each time a packet is transmitted. If a queue has no more packets it is destroyed. The arriving packets are queued to the appropriate node and if no such exists then it is created.

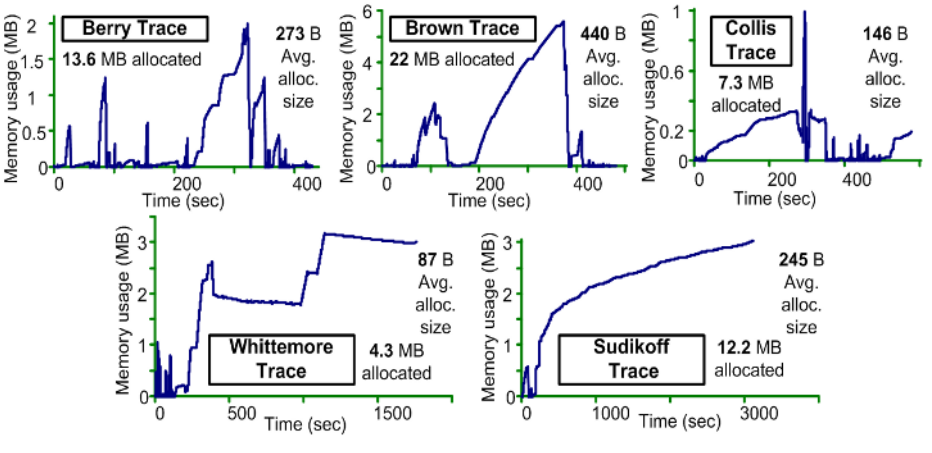


Fig. 2. Real memory usage of the DRR application for wireless traffic traces of different buildings [4] (50.000 packets)

It is important to stress that the simulation results of the DRR application were taken for 5 real wireless traffic traces. These traces represent the traffic of 5 different buildings in Dartmouth University Campus [4]. As noted in [6], a randomly generated trace is not valid for predicting how well a particular DM allocator will perform on a real program. The reason is that for different inputs there will be different dynamic allocation behaviors and allocation sizes(as shown in Fig. 2 and 3 respectively). The effect of the different dynamic behaviors can be seen in the variation of the simulation results (as shown in Table 2).

After an exhaustive exploration of all the custom DM allocators (as explained in the previous section), we use an instantiation of the proposed parameterized setup that is described at the end of section 5. Namely, we use 2 *freelists* for memory blocks of 16 Bytes and memory blocks of 1476 Bytes and we fully apply the *coalescing mechanism*, the *splitting mechanisms* and the *first fit policy*. Note that although in the packet traces the ACK packet has zero size and the MTU packet has a size of 1460 Bytes, 16 Bytes more are allocated per objects to store some application-specific data (e.g. like *Quantum*). From our exhaustive exploration we have concluded that the aforementioned custom DM allocator is the most balanced, giving both low fragmentation and good performance (other custom DM allocators give only good performance or only low fragmentation).

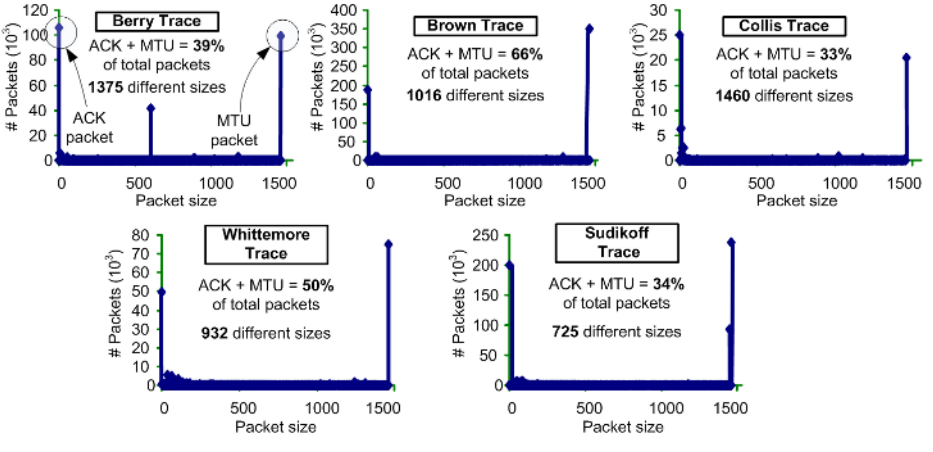


Fig. 3. Histograms of memory allocation requests of the DRR application for wireless traffic traces of different buildings [4]

Then we simulate and compare our customized DM allocator with O.S. based DM allocators for 5 different network traces (note that the very bad fragmentation and performance results of all the allocators for the Sudikoff trace are attributed to the ramp form of its memory usage, i.e. too much network traffic results in DM allocation bottleneck). We observe that for the average of all the traces our custom DM allocator is both faster and has less fragmentation than any O.S. based DM allocator. In fact, it can achieve memory fragmentation re-

Table 1. Simulation results for the DRR scheduling algorithm running for 50.000 packets per trace (lower fragmentation and execution-time is better)

DM	Fragmentation						Performance (execution-time sec.)					
Allocators	**Avrg.**	Ber.	Br.	Col.	Sud.	Whit.	**Avrg.**	Ber.	Br.	Col.	Sud.	Whit.
Windows CE	**83%**	70%	13%	59%	251%	20%	**1.78**	0.36	0.78	0.34	5.17	2.27
Windows XP	**142%**	169%	21%	183%	256%	80%	**1.69**	0.28	0.58	0.31	5.25	2.03
Linux	**66%**	35%	8%	59%	206%	23%	**2.19**	0.33	0.79	0.50	6.58	2.74
Enea OSE	**93%**	62%	8%	100%	212%	86%	**7.91**	8.40	10.88	8.55	8.04	3.67
uClinux	**152%**	93%	49%	153%	350%	117%	**2.40**	0.13	0.51	0.33	6.68	4.34
Avrg. Alloc.	**107%**	**86%**	**20%**	**111%**	**255%**	**65%**	**3.19**	**1.90**	**2.71**	**2.01**	**6.34**	**3.01**
Proposed Alloc.	**55%**	20%	1%	35%	183%	35%	**1.62**	0.17	0.46	0.24	5.13	2.09

Table 2. Simulation results for the Easyport buffering algorithm running for 4.200 packets per trace (lower fragmentation and execution-time is better)

DM	Fragmentation				Performance (execution-time sec.)			
Allocators	**Avrg.**	Trace 1	Trace 2	Trace 3	**Avrg.**	Trace 1	Trace 2	Trace 3
Windows CE	**46%**	30%	75%	34%	**0.51**	0.62	0.33	0.60
Windows XP	**49%**	33%	78%	37%	**0.49**	0.59	0.31	0.59
Linux	**40%**	29%	62%	28%	**0.59**	0.69	0.37	0.71
Enea OSE	**46%**	30%	70%	36%	**1.20**	1.42	0.81	1.39
uClinux	**60%**	40%	97%	42%	**0.87**	1.02	0.62	0.97
Avrg. Alloc.	**48%**	**32%**	**77%**	**35%**	**0.73**	**0.86**	**0.48**	**0.85**
Proposed Alloc.	**37%**	20%	61%	30%	**0.47**	0.52	0.32	0.59

ductions up to 97.82% (48.39% on average) and execution time reductions up to 97.20% (49.22% on average).

The second case study presented is the Easyport wireless network application produced by Infineon [16]. Easyport features packet and ATM cell processing functionality for data and voice/data Integrated Access Devices (IADs), enterprise gateways, access routers, and Voice over IP (VoIP) gateways. Easyport allocates dynamically the packets it receives from the Ethernet channels in a memory before it forwards them in a FIFO way. To run simulations of Easyport, we used 3 typical packet traffic traces provided by Infineon (mainly ftp sessions).

After an exhaustive exploration of the all the custom DM allocators (as explained in the previous section), we again use an instantiation of the proposed parameterized setup that is described at the end of section 5. Namely, 2 *freelists* for memory blocks of 66 Bytes and memory blocks of 1514 Bytes and we fully apply the *coalescing mechanism*, the *splitting mechanisms* and the *first fit policy*. Again this specific custom DM allocator is the most balanced. We observe that for the average of all the traces our custom DM allocator is both faster and has less fragmentation than any O.S. based DM allocator. In fact, it can achieve memory fragmentation reductions up to 50.16% (23.0% on average) and execution time reductions up to 63.39% (35.62% on average).

7 Conclusions

Dynamism is an important aspect of wired and wireless network applications. Therefore, the correct choice of a Dynamic Memory Allocation subsystem becomes of great importance. Within this context, memory fragmentation must be minimized without a performance reduction. In this paper we have presented a novel approach to explore exhaustively the combinations of the de-fragmentation techniques in custom DM allocator implementations. The results achieved with the use of our approach in real wired and wireless network applications show that our customized DM allocator solution can reduce memory fragmentation up to 97% and improve performance up to 97% compared to state-of-the-art, O.S. based DM allocators.

Acknowledgements

This work is partially supported by the European founded program AMDREL IST-2001-34379 and the Spanish Government Research Grant TIC2002/0750. We want to thank Matthias Wohrle (Advanced Systems and Circuits group, Infineon Technologies, Munich, Germany) and Arnout Vandecappelle (IMEC, DESICS group, Leuven, Belgium) for their help and support in the simulation of the Easyport application.

References

1. D. Atienza, S. Mamagkakis, F. Catthoor, et al. Dynamic Memory Management Design Methodology for Reduced Memory Footprint in Multimedia and Wireless Network Applications. In *Proc. of IEEE/ACM DATE 2004*.
2. D. Atienza, S. Mamagkakis, et al. Modular Construction and Power Modelling of Dyn. Mem. Managers for Embedded Systems. In *Proc. of LNCS PATMOS 2004*.
3. C. Williamson. A Tutorial on Internet Traffic Measurement In *Proc. of IEEE Internet Computing, Vol. 5, No. 6, 2001*
4. D. Kotz, et al. Analysis of a campus-wide wireless network. In *Dartmouth CS Technical Report TR2002-432*.
5. P. R. Wilson, et al. Dynamic storage allocation, a survey and critical review. In *Int. Workshop on Mem. Manag.*, UK, 1995.
6. M. Johnstone, et al. The Memory Fragmentation Problem: Solved? In *Proc. of Intnl. Symposium on Memory Management 1998*
7. E. D. Berger, et al. Composing high-performance memory allocators. In *Proc. of ACM SIGPLAN PLDI*, USA, 2001.
8. Dyn. Allocation in uClinux RTOS. `http://linuxdevices.com/articles/AT7777470166.html`
9. Dyn. Allocation in Enea OSE RTOS. `http://www.realtime-info.be/magazine/01q3/2001q3_p047.pdf`
10. Dyn. Allocation in MS Windows CE. `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcecoreos5/html/wce50conheaps.asp`
11. Dyn. Allocation in MS Windows XP. `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngenlib/html/heap3.asp`

12. G. Attardi, et al. A customizable memory management framework for c++. *Software Practice and Experience*, 1998.
13. D. Bacon, et al. A Real-time Garbage Collector with Low Overhead and Consistent Utilization. In *Proc. of SIGPLAN 2003*
14. M. Shreedhar, et al. Efficient Fair Queuing using Deficit Round Robin. In *Proc. of SIGCOMM 1995*
15. M. Gerharz, et al. A Practical View on Quality-of-Service Support in Wireless Ad Hoc Networks. In *Proc. of IEEE ASWN 2003*
16. Infineon Easyport. `http://www.itc-electronics.com/CD/infineon%2010063/cd1/html/p_ov_33433_-9542.html`

Author Index