David Coudert
David Simplot-Ryl
Ivan Stojmenovic (Eds.)

LNCS 5198

Ad-Hoc, Mobile and Wireless Networks

7th International Conference, ADHOC-NOW 2008
Sophia-Antipolis, France, September 2008
Proceedings

Lecture Notes in Computer Science 5198

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Alfred Kobsa
University of California, Irvine, CA, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

David Coudert David Simplot-Ryl
Ivan Stojmenovic (Eds.)

Ad-hoc, Mobile and Wireless Networks

7th International Conference, ADHOC-NOW 2008
Sophia-Antipolis, France, September 10-12, 2008
Proceedings

Springer

Volume Editors

David Coudert
Centre de Recherche, INRIA Sophia Antipolis
Mascotte, INRIA, I3S, CNRS UMR 6070, Univ. Nice Sophia
06902 Sophia-Antipolis Cedex, France
E-mail: David.Coudert@sophia.inria.fr

David Simplot-Ryl
Centre de Recherche INRIA Lille, IRCICA/LIFL
CNRS UMR 8022, Univ. Lille
BP 70478 59658 Villeneuve d' Ascq, France
E-mail: David.Simplot@lifl.fr

Ivan Stojmenovic
SITE, University of Ottawa
Ontario K1N 6N5, Canada
and
EECE,University of Birmingham, UK
E-mail: stojmenovic@storm.ca

Library of Congress Control Number: Applied for

CR Subject Classification (1998): C.2, D.2, H.4, H.3, I.2.11, K.4.4, K.6.5

LNCS Sublibrary: SL 5 – Computer Communication Networks
and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-85208-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-85208-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2008
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12459926 06/3180 5 4 3 2 1 0

Preface

The 7th International Conference on Adhoc, Mobile and Wireless Networks (AdHoc-NOW 2008) was held at INRIA Sophia Antipolis - Méditerranée, on the French Riviera, during September 10–12, 2008. The six previous conferences in the series were held in Morelia (2007), Ottawa (2006), Cancun (2005), Vancouver (2004), Montreal (2003) and Toronto (2002). The purpose of this conference is to provide a forum for researchers from academia/industry and practitioners to meet and exchange ideas regarding recent developments in the areas of ad-hoc wireless networks.

AdHoc-NOW 2008 received 110 submissions submitted by authors form the following 33 countries: Algeria, Australia, Austria, Belgium, Brazil, Canada, China, the Czech Republic, Denmark, Finland, France, Germany, Greece, India, Iran, Israel, Italy, Japan, Luxembourg, Macedonia, Norway, Pakistan, Poland, Slovakia, South Africa, South Korea, Sri Lanka, Sudan, Switzerland, Taiwan, Tunisia, the UK and the USA. Each paper was assigned to three members of the Technical Program Committee (TPC). Based on the reviews, we decided to accept 39 submissions as regular papers, 24 of them with 25 minutes' oral presentation time, and 15 as poster presentations. All of the accepted papers appear in this volume.

We thank the three invited speakers at this conference, Srdjan Krco (Ericsson, Ireland), Xuemin (Sherman) Shen (University of Waterloo, Canada), and Stephan Olariu (Old Dominion University, USA) for accepting our invitation to share their insights on new developments in their research areas.

We would like to express our sincere gratitude to all the members of the local organizing committee who invested their time and energy to organize this conference. In particular, we thank F. Huc, C. Jullien, P. Lachaume, X. Li, C. Molle, and H. Rivano.

Finally, we acknowledge the various sources of financial support for AdHoc-NOW 2008, namely the GDR ASR ResCom, European project IST AEOLUS IP-015964, INRIA Sophia Antipolis - Méditerranée, I3S, Orange Labs, Région Provence Alpes Côtes d'Azur, and the Université de Nice Sophia.

September 2008

David Coudert
David Simplot-Ryl
Ivan Stojmenovic

Organization

Steering Committee

Evangelos Kranakis (Carleton Univ.)
Michel Barbeau (Carleton Univ.)
S.S. Ravi (SUNY Albany)
Ioanis Nikolaidis (Univ. Alberta)
Violet R. Syrotiuk (Arizona State Univ.)
Thomas Kunz (Carleton Univ.)

General Chair

David Coudert (INRIA)

Program Co-chairs

David Simplot-Ryl (INRIA)
Ivan Stojmenovic (Univ. Birmingham)

Poster and Demonstration Chairs

Srdjan Krco (Ericsson)
Michel Syska (Univ. Nice Sophia)

Publicity Chair

Hervé Rivano (CNRS)

Submission Chair

Xu Li (Carleton Univ.)

Local Arrangements

Corinne Jullien (CNRS)
Florian Huc (CNRS)
Patricia Lachaume (INRIA)
Christelle Molle (DGA-CNRS)

Technical Program Committee

E. Altman (INRIA)
D. Barthel (Orange Labs)
J. Cao (Hong Kong Polytechnic Univ.)
N. Abu-Ghazaleh (SUNY Binghamton)
E. Chavez (Univ. Michoacana)
C. Constantinou (Univ. Birmingham)
C. Tung Chou (Univ. New South Wales)
M. Denko (Univ. Guelph)
M. Dohler (CTTC)
E. Fleury (ENS Lyon)
H. Frey (Univ. Southern Denmark)
H. Karl (Univ. Paderborn)
E. Kranakis (Carleton Univ.)
D. Krizanc (Wesleyan Univ.)
T. Kunz (Carleton Univ.)
X.-Y. Li (IIT)
W. Liang (Australian National Univ.)
H. Liu (Univ. Ottawa)
C. Mascolo (Univ. College of London)
L. Narayanan (Concordia Univ.)
I. Nikolaidis (Univ. Alberta)
J. Opatrny (Concordia Univ.)
M. Papatriantafilou (Chalmers Univ.)
P. Penna (Univ. Salerno)
P.M. Ruiz (Univ. Murcia)
Q.-A. Zeng (Univ. Cincinnati)

External Referees[1]

Andrea Clementi
Pierluigi Crescenzi
Mieso Denko
Ranran Ding
Ralph El-Khoury
Juan J. Galvez
Georgios Georgiadis
Vineet Josh
Ralf Klasing
Xu Li
Weifa Liang
Ke Liu
Hao Luan
Xufei Mao
Juan Antonio Martinez
Benoît Miscopein
Lata Narayanan
Giovanni Neglia
Napoleão Nepomuceno
Ioanis Nikolaidis
Bence Pasztor
Paolo Penna
Giuseppe Persiano
Hervé Rivano
Francisco J. Ros
Juan Antonio Sanchez
Divya Sardana
Jean Schwoerer
Qingyan Xie
Ping Xu
Xiao-Hua Xu
Cheng Zhu

Sponsors

GDR ASR ResCom
European project IST AEOLUS IP-015964
INRIA Sophia Antipolis - Méditerranée
I3S
Orange Labs
Région Provence Alpes Côtes d'Azur
Université de Nice Sophia

[1] This list has been automatically compiled from the conference's database. We apologize for any omissions or inaccuracies.

Table of Contents

Local Maximal Matching and Local 2-Approximation for Vertex Cover in UDGs (Extended Abstract) 1
Andreas Wiese and Evangelos Kranakis

Opportunistic Clock Synchronization in a Beacon Enabled Wireless Sensor Network 15
Nicola Altan and Erwin P. Rathgeb

Mitigating Reply Implosions in Query-Based Service Discovery Protocols for Mobile Wireless Ad Hoc Networks 29
Antônio Tadeu A. Gomes, Artur Ziviani, Luciana S. Lima, Markus Endler, and Guillaume Chelius

Adaptive MANET Routing: A Case Study 43
Liang Qin and Thomas Kunz

Self-interference in Multi-hop Wireless Chains: Geometric Analysis and Performance Study 58
Saquib Razak and Nael B. Abu-Ghazaleh

Energy-Efficient Multi-path Routing in Wireless Sensor Networks 72
Philipp Hurni and Torsten Braun

Approximating Minimum-Power k-Connectivity 86
Zeev Nutov

A Secure Cross-Layer Protocol for Multi-hop Wireless Body Area Networks 94
Dave Singelée, Benoît Latré, Bart Braem, Michael Peeters, Marijke De Soete, Peter De Cleyn, Bart Preneel, Ingrid Moerman, and Chris Blondia

Communication in Random Geometric Radio Networks with Positively Correlated Random Faults 108
Evangelos Kranakis, Michel Paquette, and Andrzej Pelc

The Mathematics of Routing in Massively Dense Ad-Hoc Networks 122
Eitan Altman, Pierre Bernhard, and Alonso Silva

Localized Spanner Construction for Ad Hoc Networks with Variable Transmission Range 135
David Peleg and Liam Roditty

Geographic Routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks 148
Luminita Moraru, Pierre Leone, Sotiris Nikoletseas, and Jose Rolim

DIN: An Ad-Hoc Algorithm to Estimate Distances in Wireless Sensor Networks 162
Freddy López Villafuerte and Jochen Schiller

Cheating on the CW and RTS/CTS Mechanisms in Single-Hop IEEE 802.11e Networks 176
Szymon Szott, Marek Natkaniec, and Andrzej R. Pach

Adapting BitTorrent to Wireless Ad Hoc Networks 189
Mohamed Karim Sbai, Chadi Barakat, Jaeyoung Choi, Anwar Al Hamra, and Thierry Turletti

Optimal Gathering Algorithms in Multi-Hop Radio Tree-Networks ith Interferences 204
Jean-Claude Bermond and Min-Li Yu

Distributed Qualitative Localization for Wireless Sensor Networks 218
Karel Heurtefeux and Fabrice Valois

A Lower Bound on the Capacity of Wireless Ad Hoc Networks with Cooperating Nodes 230
Anthony S. Acampora and Louisa Pui Sum Ip

Attacks on CKK Family of RFID Authentication Protocols 241
Zbigniew Gołębiewski, Krzysztof Majcher, and Filip Zagórski

On Backoff in Fading Wireless Channels 251
SeonYeong Han and Nael B. Abu-Ghazaleh

TSLA: A QoS-Aware On-Demand Routing Protocol for Mobile Ad Hoc Networks 265
C. Mbarushimana and A. Shahrabi

Query Dissemination with Predictable Reachability and Energy Usage in Sensor Networks 279
Zinaida Benenson, Markus Bestehorn, Erik Buchmann, Felix C. Freiling, and Marek Jawurek

A Prediction Based Cross-Layer MAC/PHY Interface for CDMA Ad Hoc Networks 293
Pegdwindé Justin Kouraogo, François Gagnon, and Zbigniew Dziong

Utility-Based Uplink Power Control in CDMA Wireless Networks with Real-Time Services 307
Timotheos Kastrinogiannis, Eirini-Eleni Tsiropoulou, and Symeon Papavassiliou

Adaptive Priority Based Distributed Dynamic Channel Assignment for Multi-radio Wireless Mesh Networks 321
Tope R. Kareem, Karel Matthee, H. Anthony Chan, and Ntsibane Ntlatlapa

Ranking and Sorting in Unreliable Single Hop Radio Network 333
Marcin Kik

Distributed Monitoring in Ad Hoc Networks: Conformance and Security Checking 345
Wissam Mallouli, Bachar Wehbi, and Ana Cavalli

Improved Distributed Dynamic Power Control for Wireless Mesh Networks 357
Thomas Olwal, Felix Aron, Barend J. van Wyk, Yskandar Hamam, Ntsibane Ntlatlapa, and Marcel Odhiambo

Identifying the Boundary of a Wireless Sensor Network with a Mobile Sink 369
Majid I. Khan, Wilfried N. Gansterer, and Günter Haring

Analysis of IEEE 802.11e Line Topology Scenarios in the Presence of Hidden Nodes 380
Katarzyna Kosek, Marek Natkaniec, and Andrzej R. Pach

Interference and Congestion Aware Reservations in Wireless Multi-hop Networks 391
Stéphane Rousseau, Laure Lebrun, Hervé Aïache, and Vania Conan

Low-Cost and Accurate Intra-flow Contention-Based Admission Control for IEEE 802.11 Ad Hoc Networks 401
Abdelouahid Derhab

An Energy-Efficient Query Aggregation Scheme for Wireless Sensor Networks 413
Jun-Zhao Sun

Novel Algorithms for the Network Lifetime Problem in Wireless Settings 425
Michael Elkin, Yuval Lando, Zeev Nutov, Michael Segal, and Hanan Shpungin

Message Quality for Ambient System Security 439
Ciarán Bryce

Request Satisfaction Problem in Synchronous Radio Networks 451
Benoît Darties, Sylvain Durand, and Jérôme Palaysi

A Novel Mobility Model from a Heterogeneous Military MANET Trace 463
Xiaofeng Lu, Yung-chih Chen, Ian Leung, Zhang Xiong, and Pietro Liò

Measuring Energy-Time Efficiency of Protocol Performance in Mobile Ad Hoc Networks 475
Ida Pu, Yuji Shen, and Jinguk Kim

A Framework for Joint Cross-Layer and Node Location Optimization in Mobile Sensor Networks 487
Vladimir Marbukh and Kamran Sayrafian-Pour

Author Index 497

Local Maximal Matching and Local 2-Approximation for Vertex Cover in UDGs
(Extended Abstract)

Andreas Wiese[1,⋆,⋆⋆] and Evangelos Kranakis[2,⋆⋆⋆]

[1] Technische Universität Berlin, Institut für Mathematik, Germany
[2] School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6

Abstract. We present $1-\epsilon$ approximation algorithms for the maximum matching problem in location aware unit disc graphs and in growth-bounded graphs. The algorithm for unit disk graph is local in the sense that whether or not an edge is in the matching depends only on other vertices which are at most a constant number of hops away from it. The algorithm for growth-bounded graphs needs at most $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \cdot \log^* n\right)$ communication rounds during its execution. Using these matching algorithms we can compute vertex covers of the respective graph classes whose size are at most twice the optimal.

1 Introduction

Unit Disk Graphs (UDGs) is a widely used concept for modeling ad hoc and wireless networks. In these graphs, the connectivity of two nodes is established if and only if their Euclidean distance of these two nodes is at most one. Therefore, UDGs model the setting of identical wireless devices on a plane without obstacles that could obscure the wireless signals. There are also other models for wireless networks, e.g., Quasi-Unit-Disk-Graphs (Q-UDGs) which were first introduced by Barriere et al. [2]. In Q-UDGs there is a certain radius ℓ such that two nodes which are closer to each other than ℓ are always connected whereas nodes with a larger distance than one unit are always disconnected. This and other models for wireless networks are captured by growth-bounded graphs. These are graphs in which for any vertex v the size of an independent set of the vertices which are at most r hops away from v is at most $f(r)$ (for a certain growth-function f).

In the setting of wireless and ad-hoc-networks there is usually no global communication backbone available. So for organizing the network traffic and solving

⋆ Research conducted while the authors were visiting the School of Computing Science at Simon Fraser University, Vancouver.
⋆⋆ Research supported by a scholarship from DAAD (German Academic Exchange Service).
⋆⋆⋆ Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada). Research supported in part by MITACS (Mathematics of Information Technology and Complex Systems).

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 1–14, 2008.
© Springer-Verlag Berlin Heidelberg 2008

problems like matching or vertex cover we need to find a method that does not rely on global information of the network. So we are interested in local algorithms. These are algorithms for which the result of a computation for a vertex or an edge depends only on the vertices and edges which are at most a certain distance away from them (the locality distance). With this constraint we ensure that we do not need knowledge of the entire network but only information about the network in a certain neighborhood of a vertex or an edge. It is also of interest in dynamically changing networks since if only small changes occur local algorithms need to recompute only small parts of the solution.

In our UDG graph model we assume that every node is aware of its geographic position in the plane. Allowing this positional knowledge we will see that the locality distance of our algorithms can be bounded by a constant. Note that this constant does not depend on the overall size of the network or the maximal vertex degree. Since positioning systems like GPS become more and more common, this setting seems to be relevant.

For organizing communication in wireless networks matching is a useful concept. In one communication round a node can usually receive data from only sender (due to interference) and each sender can send only one package at a time (usually to one receiver). Thus the sender/receiver pairs form a matching in the underlying network graph. Research has been done on finding matchings with certain properties [3] in order to deal with interference and noise issues. Also, in the computation of schedules for allocating bandwidth the matching problem can arise [12].

1.1 Related Work

The matching problem is in P for general graphs [5]. The first algorithm due to Edmonds requires a runtime of $O\left(n^3\right)$. For the restricted case of bipartite graphs there are improvements known, e.g., the Hopcroft-Karp algorithm [7].

The vertex cover problem is NP-hard in general graphs [6], but there are several polynomial time approximation algorithms which guarantee an approximation factor of 2, e.g., in [1]. However, it is NP-hard to approximate the problem with a factor better than $10\sqrt{5} - 21 \approx 1,3607$ [4]. Thus there can be no polynomial time approximation scheme (PTAS), unless $P = NP$. When we restrict the setting to unit disk graphs, vertex cover remains NP-hard. The same holds for growth-bounded graphs since this class includes unit graphs. However, for unit disk graphs PTASs are known. For the case where the embedding of the graph is known, Hunt III et al. [8] presented the first approximation scheme. The algorithm for independent set presented in [10] together with the technique in [14] yields a global PTAS for vertex cover that does not rely on the embedding of the graph. There is also a local PTAS known for the setting of location aware UDGs [14].

1.2 Our Results

We present the first local approximation algorithms for matching in location aware Unit Disk Graphs. It achieves an approximation ratio of $1-\epsilon$ for arbitrarily

small ϵ. In this setting we can show that the locality distance of our algorithm (i.e. the radius of the area that needs to be explored in order to compute the status of one edge) is bounded by a constant. In particular, this constant does not depend on the size of the entire network or the maximal degree of a vertex. For growth-bounded graphs we also give a $1-\epsilon$ approximation algorithm. For this setting we lift the assumption of positional information in the nodes and require only a unique ID in each vertex. The locality distance for this algorithm is in $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \log^* n\right)$. All matchings computed by these algorithms are maximal.

Each matching algorithm yields a local approximation algorithm for the vertex cover problem. The locality properties of these algorithms are identical to the respective matching algorithms. The size of the computed vertex covers are at most twice the size of an optimal vertex cover. As mentioned above, for location aware unit disk graphs there is a local PTAS known [14]. However, the locality distance of this PTAS when executed with approximation factor 2 is a lot larger than the locality of Algorithm 3.

1.3 Organization of the Paper

In Section 2 we present our local $1-\epsilon$ approximation algorithm for matching in location aware unit disk graphs. In Section 3 we show how the ideas of this algorithm can be used in order to derive a local $1-\epsilon$ approximation algorithm for the same problem in growth-bounded graphs. Our local approximation algorithms for vertex cover with approximation factor 2 are presented in Section 4. Finally in Section 5 we summarize our results and address open problems.

2 Maximum Matching in Location Aware UDGs

In this section we present a local $1-\epsilon$ approximation algorithm for the maximum matching problem in location aware unit disk graphs. First we give some basic definitions. Then we define a tiling of the plane that we are going to use in our algorithm. Finally we present the algorithm and prove its correctness.

2.1 Definitions

The graph $G = (V, E)$ considered in this section is a unit disk graph. For two vertices u and v let $d(u, v)$ be the hop-distance between u and v, that is the number of edges on a shortest path between these two vertices. Note that the hop-distance between two vertices does not necessarily equal the geometric distance between them. Denote by $N^r(v) = \{u \in V \mid d(u, v) \leq r\}$ the r-th neighborhood of a vertex v. In Section 3 we will consider growth-bounded graphs.

Definition 1. *An undirected graph $G = (V, E)$ is called a f-*growth-bounded graph *if there exists a polynomial bounding function $f(r)$ such that for every $v \in V$ and $r \geq 0$, the size of the largest independent set in the r-neighborhood $N^r(v)$ is at most $f(r)$.*

Similarly we define families of graphs to be growth-bounded.

Definition 2. *Let $\mathcal{G}$ be a family of graphs. We call* $\mathcal{G}$ polynomially growth-bounded*, if there exists a polynomial bounding function $f(r)$ such that for every graph $G \in \mathcal{G}$, every vertex v in G and every $r \geq 0$, the size of the largest independent set in the r-neighborhood $N^r(v)$ in G is at most $f(r)$.*

In the sequel when refering to growth-bounded we will mean polynomially growth bounded. In addition, we will implicitly assume that the family and its bounding function $f(r)$ are known and fixed for the given class $\mathcal{G}$ of graphs.

Let $M \subseteq E$ be a set of edges. We call M a *matching*, if no two edges in M share an end-vertex. We call M a *maximum matching*, if for all matchings M' it holds that $|M'| \leq |M|$. A *maximal matching* is a matching which cannot be extended by adding another edge.

Let M be a matching. We call a path p an *M-alternating path*, if it contains alternating matching- and non-matching edges. We call a vertex v an *isolated* vertex, is it is not adjacent to an edge from M. We call an M-alternating path p an *M-augmenting path,* if it starts from and ends on isolated vertices. Note: An M-augmenting path has an odd number of edges.

Lemma 1. A matching M is a maximum matching, if and only if there is no M-augmenting path.

2.2 Tiling of the Plane

Let $1 - \epsilon$ be the desired approximation ratio for the matching algorithm. We define k to be the smallest integer such that $\epsilon \geq \frac{2}{k+1}$. We tile the plane with an infinitely repeated pattern of rectangles as seen in Figure 1. Each rectangle is assigned class number 1, 2 or 3. The height of each rectangle is $2k + 2$, the width of each rectangle is $4k + 4$.

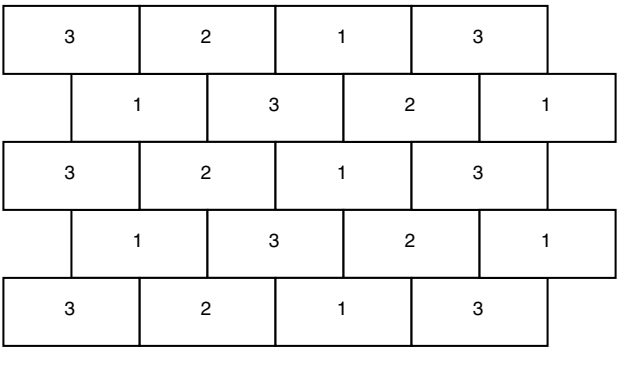


Fig. 1. The tiling of the plane

2.3 The Algorithm

Now we present the algorithm. It has three phases:

1. For each rectangle R we compute a matching that includes only edges that have both end-vertices in R.
2. For each class 1 rectangle R we check if there are augmenting paths in the subgraph induced by the vertices which are at most k hops away from R. If there are such paths, we augmented the matching until no such paths are left.
3. For each class 2 rectangle R we check if there are augmenting paths in the subgraph induced by the vertices in R and the vertices in class 3 rectangles which are at most k hops away from R. As in the step before, we augment the matching along all these paths.

Now we present the algorithm in detail. For all rectangles R we do the following: Denote by V_R the vertices in R. For the subgraph induced by V_R we compute a maximum matching using a standard matching algorithm. Since all V_R for the different rectangles R are disjoint the order in which we do this does not matter. Now we come to phase 2: For each class 1 rectangle R we take the set of vertices which are in R or at most k hops away from a vertex in R. Denote this set by V'_R. In the subgraph induced by V'_R we augment the matching along all augmenting paths. Since the height of the rectangles is $2k+2$ and their width is $4k+4$, the order in which the class 1 rectangles are being processed does not matter. Finally we start phase 3: For each class 2 rectangle R we compute all vertices which are at most k hops away from R. Denote this set by V''_R. For the subgraph induced by V''_R we we augment the matching along all augmenting paths. Denote by M the resulting matching. We refer to the above as Algorithm 1.

In the following theorem we prove that Algorithm 1 is a local algorithm that computes a valid matching with a competitive ratio of $1-\epsilon$.

Theorem 1. *Algorithm 1 has the following properties:*

1. *The computed matching M is a maximal matching for G.*
2. *Let M_{OPT} be an optimal matching for G. It holds that $|M| \geq (1-\epsilon)\cdot|M_{OPT}|$.*
3. *Whether or not an edge $e=(u,v)$ is a matching edge depends only on the vertices which are at most $O\left(1/\epsilon^2\right)$ hops away from u or v, i.e. Algorithm 1 is local.*
4. *The processing time for an edge $e=(u,v)$ is bounded by a cubic polynomial in the number of vertices which are at most $O\left(1/\epsilon^2\right)$ hops away from u or v.*

2.4 Proof of Correctness

We will prove the four parts of this theorem in four steps.

Validity and Maximality. We prove that M is a valid matching for G and that it is maximal.

Algorithm 1. Algorithm for finding a matching in a unit disk graph $G = (V, E)$

```
// Phase 1;
foreach rectangle R do
    // denote by V_R the vertices in R;
    determine a maximum matching M_R for the subgraph induced by V_R;
end
Define M := ⋃_{R∈T} M_R;
// Phase 2;
foreach rectangle R with class(R) = 1 do
    Denote by V_R all vertices in R;
    Explore all vertices which are at most k hops away from vertices in V_R;
    // Denote these vertices by V'_R;
    Augment M along augmenting paths in the subgraph induced by V'_R;
end
// Phase 3;
foreach rectangle R with class(R) = 2 do
    Denote by V_R all vertices in R;
    Explore all vertices which are at most k hops away from vertices in V_R;
    // Denote these vertices by V''_R;
    Augment M along augmenting paths in the subgraph induced by V''_R;
end
```

Proof. (of part 1 of Theorem 1): The matchings constructed in phase 1 are clearly valid. Since in phase 2 and 3, M is only augmented along augmenting paths, the resulting matching is valid as well.

Now we want to prove that M is maximal. We call an edge that would extend M an *extending edge*. Since we augment the matching along augmenting paths a vertex which is adjacent to a matching edge once will be adjacent to a matching edge in the final matching as well. We see that after phase 1 all extending edges must have their adjacent vertices in different rectangles since we compute maximum matchings for each rectangle. From the construction of the tiling we see that these rectangles must have different class number (since the length of an edge is at most 1). After phase 2 there are no extending edges between class 1 and 2 rectangles left since the matching would be augmented along such "paths". With the same reasoning we see that after phase 3 there are no extending edges between class 2 and 3 rectangles left. So for the final matching there are no extending edges in the graph. This implies that the matching M is maximal.

Approximation Ratio. Let M_{OPT} be an optimal matching for G. We prove that $|M| \geq (1 - \epsilon) \cdot |M_{OPT}|$.

Proof. (of part 2 of Theorem 1): Denote by M_i the matching computed by the algorithm after phase i for $i \in \{1, 2, 3\}$. Let P_i be the set of augmenting paths for M_i with $i \in \{1, 2, 3\}$. From the construction of M_1 it follows that all paths in P_1 must have their start - and endvertices in two different rectangles. From the algorithm we see that all paths in P_2 either

– do not have their start- and endvertex in a rectangle of class 1 or
– are longer than k,

since all other augmenting paths in P_1 are eliminated in phase 2. Similarly, in P_3 all augmenting paths which are left are longer than k edges.

Consider $M' := M_3 \triangle M_{OPT} = (M_3 - M_{OPT}) \cup (M_{OPT} - M_3)$ and $G' := (V, M')$. All nodes in G' have a degree of at most two (since M_3 and M_{OPT} are both matchings). Its connected components are

– isolated vertices
– cycles of even length
– paths of three possible types

  - Paths starting and ending with an edge from M. This cannot happen since this would be an augmenting path for M_{OPT} and M_{OPT} is optimal.
  - Paths starting with an edge from M and ending with an edge from M_{OPT}. These paths have the same number of edges from M as from M_{OPT}.
  - Paths starting and ending with an edge from M_{OPT}. These are augmenting paths for M. Denote all these paths by P'_3.

Every augmentation would increase the number of edges in M by one, so $|M| + |P'_3| = |M_{OPT}|$. Since $P'_3 \subseteq P_3$ all paths in P'_3 have more than k edges. So every path in P'_3 contains at least $\frac{k+1}{2}$ edges of M_{OPT}. Since the paths are disjoint, it follows that $|P'_3| \leq |M_{OPT}| / \frac{k+1}{2}$. We then have

$$\begin{aligned} |M| &= |M_{OPT}| - |P'_3| \\ &\geq |M_{OPT}| - \frac{2\,|M_{OPT}|}{k+1} \\ &\geq (1-\epsilon)\,|M_{OPT}| \end{aligned}$$

Locality. We prove that whether or not an edge $e = (u, v)$ belongs to M depends only on the vertices which are at most $O\left(1/\epsilon^2\right)$ hops away from u or v. First we need to give a technical lemma.

Lemma 2. *Let R be a rectangle and $G[R]$ the graph G restricted to R. For each connected component C in $G[R]$ it holds that $diam(C) \leq 22k^2 + 58k + 39$. Let R' be a rectangle and $G[R']$ the graph G restricted to the vertices which are at most k hops away from R' (including the vertices in R' itself). Then for each connected component C' in $G[R']$ it holds that $diam(C') \leq 30k^2 + 70k + 31$.*

Proof. First we derive an upper bound for the maximum size of an independent set in $G[R]$. The area of R plus a surrounding belt of width $1/2$ around it is $(2k+3) \cdot (4k+5) = \left(8k^2 + 22k + 15\right)$. So there can be at most $\left\lfloor \frac{8k^2+22k+15}{\pi/4} \right\rfloor$ centers of non-overlapping discs of radius $1/2$ in R. We compute that $\left\lfloor \frac{8k^2+22k+15}{\pi/4} \right\rfloor \leq \left\lfloor \frac{32k^2}{\pi} \right\rfloor + \left\lfloor \frac{88k}{\pi} \right\rfloor + \left\lfloor \frac{60}{\pi} \right\rfloor \leq 11k^2 + 29k + 20$. It follows that the cardinality of a

maximum independent set in $G[R]$ is at most $11k^2 + 29k + 20$. Now consider a connected component C in $G[R]$ and two vertices $u, v \in C$ such that $d(u, v) = diam(C)$. Denote by p the shortest path between u and v in C. If we take every alternating vertex in p we get an independent set in R. As the size of such a set is bounded by $11k^2 + 29k + 20$, the length of p is bounded by $22k^2 + 58k + 39$ and therefore $diam(C) \leq 22k^2 + 58k + 39$.

Applying the same reasoning to R' we derive an upper bound of $15k^2+35k+16$ for an independent set in $G[R']$ (since $\left\lfloor \frac{(2k+3+k)\cdot(4k+4+k)}{\pi/4} \right\rfloor = \left\lfloor \frac{15k^2+27k+12}{\pi/4} \right\rfloor \leq \left\lfloor \frac{60k^2}{\pi} \right\rfloor + \left\lfloor \frac{108k}{\pi} \right\rfloor + \left\lfloor \frac{48}{\pi} \right\rfloor = 15k^2 + 35k + 16$) and therefore we get $diam(C') \leq 30k^2 + 70k + 31$ for any connected component C' in $G[R']$.

(of part 3 of Theorem 1): Denote by a_i the maximum number of hops which we need to explore around u and v in order to compute whether $e \in M$ after phase i (for $i \in \{1, 2, 3\}$).

In order to determine the status of an edge e after phase 1, we need to explore only the connected component of e in its rectangle if u and v are in the same rectangles or nothing if u and v are in different rectangles. From Lemma 2 it follows that $a_1 \leq 22k^2 + 58k + 39$. For computing the status of e after phase 2 we need to explore the connected component V'_R with $u \in V'_R$ and $v \in V'_R$ (if it exists) and what edges in V'_R were assigned to M after phase 1. It follows that $a_2 \leq a_1 + 30k^2 + 70k + 31$ (see Lemma 2). Analogously for computing the status of e after phase 3 we need to explore the connected component V''_R such that $u \in V''_R$ and $v \in V''_R$ (if such a component exists) and what edges in V'_R were assigned to M after phase 2. This implies that $a_3 \leq a_2 + 30k^2 + 70k + 31$. So altogether we get that $a_3 \leq 22k^2+58k+39+30k^2+70k+31+30k^2+70k+31 = 82k^2 + 198k + 101 \in O\left(k^2\right)$.

By definition k is the smallest integer such that $\epsilon \geq \frac{2}{k+1}$. This implies that $k \geq \frac{2}{\epsilon} - 1$ and thus $k \in O\left(1/\epsilon\right)$. It follows that $a_3 \in O\left(1/\epsilon^2\right)$.

Processing time. We want to show that the processing time of Algorithm 1 for a single edge e is in $O\left(\bar{n}(e)^3\right)$ where $\bar{n}(e)$ is the number of vertices within the locality distance of e (i.e. the number of vertices which we really need to explore in order to compute the status of e).

Proof. (of part 4 of Theorem 1): In phase 1 we need to compute a maximum matching for edges in a single rectangle. This can be done in $O\left(\bar{n}(e)^3\right)$ using any algorithm for computing a maximum matching (e.g. Edmonds algorithm [5]). In phase 2 we need to find augmenting paths in the subgraph induced by V'_R for several class 1 rectangles R. Since in the locality distance of e there can be only a constant number of class 1 rectangles this requires $O\left(\bar{n}(e)^3\right)$ time (note that the number of such class 1 rectangles does not depend on the desired approximation ratio). Applying the same reasoning in phase 3 we need a processing time of $O\left(\bar{n}(e)^3\right)$. This leads to an overall processing time of $O\left(\bar{n}(e)^3\right)$.

3 Maximum Matching without Location Awareness

In this section we present a local algorithm which computes a $1-\epsilon$ approximation for the maximum matching problem in growth-bounded graphs. In contrast to the algorithm presented in Section 2 we assume a graph model in which the embedding of the graph is unknown. We will specify this in the following section.

3.1 Graph Model

Let $G = (V, E)$ be a growth-bounded graph. We assume that every node has a unique identifier (ID). Apart from that there is no information available to distinguish the nodes from each other.

3.2 The Algorithm

Let $1 - \epsilon$ be the desired approximation ratio. The algorithm uses the same methodology as Algorithm 1 for ensuring the approximation ratio: We will compute a maximal matching M such that the length of each of the augmenting paths that could turn M into a maximum matching is at least a certain constant k. At the beginning of the algorithm we choose k according to ϵ.

The role of the rectangle classes in the algorithm above will be taken by a maximal independent set which is also a dominating set. In order to organize the computation distributively we use the same methods which were originally presented in [10].

Similarly as in Algorithm 1 we define k to be the smallest even integer such that $\epsilon \geq \frac{2}{k+1}$. We compute a maximal independent set I in G. This can be done locally using the distributed algorithm [5]. Then we define the clustergraph $\bar{G} = (\bar{V}, \bar{E})$ with radius $2k + 2$ by $\bar{V} := I$ and

$$(u, v) \in \bar{E} \Leftrightarrow d_G(u, v) \leq 2k + 2$$

Since G is a growth-bounded graph, the maximum degree $\triangle_{\bar{G}}$ of $\bar{G}$ is bounded by a constant. This allows us to use the algorithm in [11] for coloring the vertices of $\bar{G}$ with at most $O\left(\triangle_{\bar{G}}^2\right)$ colors. We initialize our matching M with $M := \emptyset$. Then we iterate over the different colors of $\bar{G}$. For each color c we do the following: For each vertex v_c which was colored with color c we compute the subgraph induced by $N^{k+1}(v_c)$. Denote by $G_c(v_c)$ such a subgraph around a vertex v_c. From the definition of $\bar{G}$ we see that the subgraphs are all disjoint. In each subgraph $G_c(v_c)$ we augment our matching M along augmenting paths until we cannot find any more augmenting paths. This can be done using a standard matching algorithm, e.g., the algorithm by Edmonds [5]. Since the subgraphs are disjoint this can be done distributively. After having iterated over all colors, we output M. We refer to this as Algorithm 2.

Theorem 2. *Algorithm 1 has the following properties:*

1. *The computed matching M is a maximal matching for G.*
2. *Let M_{OPT} be an optimal matching for G. It holds that $|M| \geq (1-\epsilon) \cdot |M_{OPT}|$.*

Algorithm 2. Algorithm for finding a matching in a unit disk graph $G = (V, E)$

```
// Let 1 − ε be the desired approximation ratio;
Define k to be the smallest integer such that (k+1)/(k+3) ≥ 1 − ε;
Compute a maximal independent set I for G;
Construct cluster graph Ḡ with radius 2k + 2;
Color Ḡ with γ = O(△²_Ḡ) colors;
M := ∅;
for i := 1 to γ do
    foreach vertex v_c with color c do do
        compute subgraph N^{k+1}(v_c);
        augment M along augmenting in N^{k+1}(v_c);
    end
end
```

3. *The algorithm requires at most* $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \cdot \log^* n\right)$ *communication rounds.*

3.3 Proof of Correctness

We will prove the four parts of this theorem in four steps.

Validity and Maximality. We want to prove that M is a matching and that it is maximal.

Proof. (of part 1 of Theorem 2): For the correctness of the subroutines for computing the maximal independent set and the vertex coloring we refer to their respective articles [9,11]. In each iteration the matching is augmented along augmenting paths. This clearly constructs a valid matching. Now we want to prove that M is maximal. Assume on the contrary that there is an edge $e = (u, v)$ with $e \notin M$ but such that $M \cup \{e\}$ is a valid matching. Since I is a maximal independent set it is also a dominating set. So there is a vertex $u' \in I$ which is adjacent to u. Let c be the color of u'. There is an iteration in which u' was considered. Since we always augment our matching along augmenting paths, both u and v were unmatched in this iteration (in $G_c(u)$). Since e is in $G_c(u)$ and we augment M along all augmenting paths in $G_c(u)$, the edge e is added to M. In all future iterations u and v will always be matched (adjacent to a matching edge). This is contradiction.

Approximation Ratio. We want to prove that for a maximum matching M_{OPT} for G it holds that $|M| \geq (1 - \epsilon) \cdot |M_{OPT}|$.

Proof. (of part 2 of Theorem 2): Like in the proof of Theorem 1 we show that there are no augmenting paths for M whose length is shorter or equal to k. Denote by $I_i \subseteq I$ all vertices in I which were colored with color i. Denote by

P_i all vertices which are either in I_i or adjacent to a vertex in I_i and denote by M_i the computed matching after the ith iteration. In the ith iteration of the algorithm we check for augmenting paths in the subgraphs $G_c(v)$ (for each $v \in I_i$). Thus after the ith iteration there are no more augmenting paths which start with an isolated vertex in P_i and whose length is at most k.

Now consider $M_i' := M_i \triangle M_{OPT}$. The edges in M_i' form either circles of even length or augmenting paths. When we compare M_i' with M_j' for $j > i$ we see that in M_j the paths from M_i are either unchanged, eliminated (because we augmented the matching along them), or two paths are connected (because we augmented along a path that connected these two paths). In both cases it still holds that all augmenting paths starting with an isolated vertex in P_i are longer than k edges.

Since I is a dominating set for G it holds that $\bigcup P_i = V$. Thus after all iterations there are no augmenting paths left which have at most k edges. So with the same argumentation as in part 2 of Theorem 1 we can show that $|M| \geq (1-\epsilon) \cdot |M_{OPT}|$.

Locality. We show that we need at most $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \cdot \log^* n\right)$ communication rounds.

Proof. (of part 3 of Theorem 2): Computing the maximal independent set I can be done in $O(\log \triangle \log^* n)$ communication rounds [9]. The coloring of the cluster graph takes $O(k \cdot \log^* n)$ rounds [11]. The computation of the matchings needs $O\left(k \cdot \triangle_{\bar{G}}^2\right)$ communication rounds since we have $O\left(\triangle_{\bar{G}}^2\right)$ different colors and we explore the vertices which are at most $k+1$ hops away from each vertex $v \in I$. The maximum degree of the cluster graph $\bar{G}$ is bounded by $O(f(2k+2))$ where $f(n)$ is the growth-bounding-function of G. By definition k is the smallest integer such that $\epsilon \geq \frac{2}{k+1}$. This implies that $k \geq \frac{2}{\epsilon} - 1$ and thus $k \in O(1/\epsilon)$.

Altogether this implies that Algorithm 2 needs at most $O\left(T_{MIS} + \frac{1}{\epsilon}\left(\log^* n + f\left(\frac{2}{\epsilon}+2\right)^2\right)\right)$ communication rounds where T_{MIS} are the communication rounds needed for computing a maximal independent set. Using the algorithm in [9] for this task we need $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \cdot \log^* n\right)$ communication rounds in total.

4 Vertex Cover

In this section we present local approximation algorithms for the minimum vertex cover problem. We use the local matching algorithms presented in Sections 2 and 3 respectively as subroutines. First we compute a maximum matching. Then we assign all vertices which are adjacent to matched edges to the vertex cover. Using a well-known reasoning we prove that this gives a factor 2 approximation for vertex cover.

4.1 The Algorithm

Let $G = (V, E)$ be a unit disk graph. First we use Algorithm 1 or Algorithm 2 in order to compute a maximal matching M. We modify the algorithm as follows:

Since we are not interested in a good approximation for the matching problem we choose $k := 1$. In order to improve the runtime of the algorithm, we consider only augmenting paths of length 1 in each phase (this is effectively a greedy-algorithm for the matching problem). Then we define our vertex cover VC as follows: $VC := \{u, v | (u, v) \in M\}$.

Using Algorithms 1 and 2 we cannot only compute maximal matchings, but also maximal matchings which are not much smaller that maximum matchings. However, for this algorithm, we could not prove a better performance ratio if we computed a matching with a certain performance guarantee. So in order to achieve a small locality distance we just compute a maximal matching.

Algorithm 3. Algorithm for finding a vertex cover in a unit disk graph $G = (V, E)$

Define $k := 1$;
Compute a maximum matching M using Algorithm 1 or Algorithm 2 and only augmenting along paths of length 1;
Define $VC := \{u, v | (u, v) \in M\}$;
Output VC;

Depending an which algorithm we use for computing the matching we get a different algorithm for vertex cover. Theorem 3 represents the algorithm that we get by using Algorithm 1, Theorem 3 the algorithm which is the result of using Algorithm 2 as a subroutine.

Theorem 3. *There is an algorithm for location aware unit disk graphs which computes a set VC with the following properties:*

1. *The computed set VC is a vertex cover for G.*
2. *Let VC_{OPT} be an optimal vertex cover for G. It holds that $|VC| \leq 2 \cdot |VC_{OPT}|$.*
3. *If a vertex v is in VC depends only on the vertices which are at most 381 hops away from v, i.e. Algorithm 3 is local.*
4. *The processing time for a vertex v is bounded by a linear polynomial in the number of edges whose adjacent vertices are both at most 381 hops away from v.*

There is an algorithm for growth-bounded graphs with unique vertex-IDs which computes a set VC with the following properties:

1. *The computed set VC is a vertex cover for G.*
2. *Let VC_{OPT} be an optimal vertex cover for G. It holds that $|VC| \leq 2 \cdot |VC_{OPT}|$.*
3. *The algorithm requires at most $O\left(\log \triangle \log^* n + \frac{1}{\epsilon}^{O(1)} \log^* n\right)$ communication rounds.*

4.2 Proof of Correctness

Here we only prove Theorem 3. The proof of Theorem 3 can be done similarly.

Proof. (of Theorem 3): From Theorem 1 we know that M is a maximal matching. Thus $V \setminus M = VC$ is a vertex cover. The cardinality of any matching in a graph forms a lower bound for the cardinality of a minimum vertex cover. This holds since every vertex of an optimal vertex cover can cover at most one edge of the matching. As we assign two vertices to VC for each edge in M we conclude that $|VC| \leq 2 \cdot |M| \leq 2 \cdot |VC_{OPT}|$. The other properties of the algorithm follow immediately from the respective properties of the matching subroutine.

5 Conclusion

We presented local $1 - \epsilon$ approximation algorithms for matching in the setting of location aware unit disk graphs and growth-bounded graphs without positional information. They are the first local approximation algorithms for matching in their respective settings. Since a local algorithm cannot perform optimally in all graph instances our approximation factors are the best possible. It remains open to find local algorithms which achieve the same approximation ratios but which need lower locality distances. For real applications low localities are always desirable since they reduce the size of the area that needs to be explored when computing the status of an edge. For Algorithm 2 the locality distance needed for computing a maximal independent set plays an important role. A local algorithm for this task with a lower locality would immediately lead to a lower locality distance of our algorithm. Also of interest would be lower bounds for the best possible approximation ratio of local algorithms for matching in these settings (depending on their locality distance).

In Section 4 we used the two matching algorithms for getting factor 2 approximation algorithms for vertex cover in the respective settings. Our algorithms achieve the best known locality distances for this approximation factor. For the setting of growth-bounded graphs without positional information, our algorithm is even the first non-trivial local algorithm for vertex cover. It remains open to fully analyze the price for good approximation ratios in terms of required locality distance. The first lower bounds on this are [13]. All improvements for the matching algorithms regarding locality distance would immediately lead to better locality distances for the vertex cover algorithms.

References

1. Bar-Yehuda, R., Even, S.: A local-ratio theorem for approximating the weighted vertex cover problem. Annals of Discrete Mathematics 25, 27–45 (1985)
2. Barrière, L., Fraigniaud, P., Narayanan, L.: Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In: DIAL-M, pp. 19–27. ACM Press, New York (2001)
3. Borbash, S.A., Ephremides, A.: The feasibility of matchings in a wireless network. IEEE/ACM Transactions on Networking 14(SI), 2749–2755 (2006)

4. Dinur, I., Safra, S.: The importance of being biased. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002), May 19–21, 2002, pp. 33–42. ACM Press, New York (2002)
5. Edmonds, J.: Paths, trees, and flowers. Canadian J. Math. 17, 449–467 (1965)
6. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the theory of NP-completeness. Freeman, NY (1979)
7. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2(4), 225–231 (1973)
8. Hunt III, H.B., Marathe, M.V., Radhakrishnan, V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. J. Algorithms 26(2), 238–274 (1998)
9. Kuhn, F., Moscibroda, T., Nieberg, T., Wattenhofer, R.: Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In: Fraigniaud, P. (ed.) DISC 2005. LNCS, vol. 3724, pp. 273–287. Springer, Heidelberg (2005)
10. Kuhn, F., Nieberg, T., Moscibroda, T., Wattenhofer, R.: Local approximation schemes for ad hoc and sensor networks. In: DIALM-POMC 2005: Proceedings of the 2005 joint workshop on Foundations of mobile computing, pp. 97–103. ACM Press, New York (2005)
11. Linial, N.: Locality in distributed graph algorithms. SIAM J. Comput. 21(1), 193–201 (1992)
12. Tassiulas, L., Sarkar, S.: Maxmin fair scheduling in wireless networks. In: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM 2002), Piscataway, NJ, USA, June 23–27, 2002, vol. 2, pp. 763–772. IEEE Computer Society, Los Alamitos (2002)
13. Wiese, A., Kranakis, E.: Impact of locality on location aware unit disk graphs. Technical Report TR-07-21, Carleton University, School of Computer Science, Ottawa (December 2007)
14. Wiese, A., Kranakis, E.: Local PTAS for Independent Set and Vertex Cover in Location Aware UDGs. In: Nikoletseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) DCOSS 2008. LNCS, vol. 5067. Springer, Heidelberg (2008)

Opportunistic Clock Synchronization in a Beacon Enabled Wireless Sensor Network

Nicola Altan and Erwin P. Rathgeb

University of Duisburg-Essen
{nicola.altan,erwin.rathgeb}@uni-due.de

Abstract. Wireless sensor networks (WSN) consisting of a large number of tiny inexpensive sensor nodes are a viable solution for many problems in the field of building automation. In order to meet the energy constraints, the nodes have to operate according to an extremely low duty cycle schedule. In such a scenario the reduction of the synchronization error, at least among directly communicating nodes, is a crucial functionality of the MAC layer. We propose a time synchronization mechanism based on the usage of a Kalman Filter (KF) on a smoothed sequence of measured beacon intervals. A side effect of this method is the introduction of a global clock synchronization. The implementation of the proposed solution is feasible on sensor devices with minimum processor, memory and energy capacity.

1 Introduction

Recent advances in electronics and radio communications have enabled the development of low-cost, low-power sensor nodes which are small in size and communicate through short range radio devices.

Building automation is, apart from the military applications, one of the most promising fields for the deployment of wireless sensor network (WSN) technologies. In this case, the WSNs will typically be deployed in an already existing building such that mains operation is much too costly because a large number of power outlets would have to be retrofitted for that purpose. Therefore, the nodes have to be autonomous, i.e. battery powered operation is necessary despite the fact that the network is operated in an indoor environment.

Simple economical considerations suggest that that the sensor nodes have to be as inexpensive as possible and they have to work unattended for several years such that the replacement of the nodes can be included into the scheduling for routine maintenance.

Considering the strict constraints with respect to the total energy budget and the expected life time imposed by the building automation scenario, the network nodes have to adopt an extremely low duty cycle scheduling. Therefore, the implementation of an at least local clock synchronization protocol is mandatory to enable the communication between the nodes over an extended period of time.

The contribution of this paper is to propose an efficient synchronization algorithm which relies only on the measurements of the beacon interval and hence

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 15–28, 2008.
© Springer-Verlag Berlin Heidelberg 2008

makes an opportunistic usage of the MAC management frames. It is applicable over a wide range of sensor network scenarios including in particular the case of very low duty cycles.

2 Requirements and Assumptions

With respect to scalability, network sizes from just a few nodes in residential buildings up to 1000 nodes in large complexes are realistic for building automation scenarios. We assume the number of nodes to be proportional to the number of rooms for the respective building.

Simple business plan considerations mandate the use of off-the-shelf sensor hardware and an unattended node life time of up to 10 years. In order to meet the total energy budget under these assumptions, the nodes have to operate with an extremely low duty cycle (no higher than 10^{-4}).

The medium access control protocol (MAC) we assume as basis for the WSN operation combines contention based reservation and contention free access periods into a single frame, which begins with the generation of a beacon signal. The reception of these beacons is the mandatory precondition for the communication with the respective sender.

The self-configuration process performed to bootstrap the network implicitly defines which nodes are able to receive the beacon signal a specific node generates, and hence the direction of the communication between neighbor nodes [1].

The assumption of a beacon based MAC protocol and of a tree-like routing structure with unidirectional links apply to a large class of WSNs. Therefore, the proposed synchronization method is generally applicable to many WSN scenarios - irrespective of the use of the specific MAC layer and bootstrap algorithm.

The interval between two consecutive beacons emitted by the same node is fixed and its nominal value $T_{beacon} = 6\,min.$ is known.

Each node is equipped with a free running clock, which is characterized by a precision ϵ_{clock}. A typical clock error of 20 ppm, e.g., results in a difference between nominal and real beacon interval of 7.2 ms in the considered network. Environmental changes may impose additional clock errors.

In order to accommodate the synchronization errors, each node activates the receiver prior to the expected beacon emission time to provide a guard period[1].

3 Related Work

There exist multiple proposals for solving the clock synchronization problem in the literature. Most of the approaches adopt the basic principle also used in the network time protocol (NTP) [2]: the nodes exchange data packets including

[1] Since hardware and energy constraints impose an upper limit on the duration of an activity period, the minimization of the guard period is of crucial importance in order to achieve a high utilization of the channel.

time stamps and then adjust their own clock in order to minimize the difference from the reference clock. All solutions based on this approach have two main drawbacks which make these unsuitable for the considered network. The synchronization protocol itself requires the transmission of data packets and hence causes additional energy costs which may be non-negligible if the system requires a tight synchronization. Moreover, all these solutions rely on the bidirectional exchange of synchronization packets and, therefore, they can not be used in a network where most of the communication links are unidirectional.

The synchronization problem in WSN has been specifically addressed in some recent publications ([3,4,5,6,7]). Each of these proposals seems to be optimized for a specific kind of WSN – and with respect to specific conditions.

Romer and Elson outline in [5] many of the challenges and issues with respect to synchronization in a WSN. In particular, the authors highlight the necessity of exploiting the a-priori knowledge about the involved systems. The same paper highlights the importance of creating an operational clock on top of the local free running clock instead of trying to modify the clock parameters.

The Reference Broadcast Synchronization (RBS) [7] performs well for multihop synchronization. It requires that nodes which receive the same synchronization packet are able to communicate. Even though this is likely in the network we consider, the additional traffic represents a source of energy expenses.

The approach proposed by Hu and Servetto [6] shows many interesting properties which meet the requirements of the WSN we consider. In particular it does not rely on the exchange of additional traffic, it uses the broadcast characteristics of the physical medium and it relies on local estimations for the emission time of synchronization pulses. In this case, the synchronization takes the form of an estimation problem. The authors show the optimality of the proposed approach if the number of nodes approaches infinity. An application to a network with a finite number of nodes is described in [8]. The algorithm used seems to be instable in networks of finite size and requires the introduction of a feedback from the reference node in order not to diverge. A critical aspect of the whole construction is the assumption of the wide sense stationarity of the process describing the measurements. In a real scenario, modifications of the environmental parameters have influence on the parameters of the stochastic process which describes the measurements. The process itself will, therefore, be no more stationary. A continuous adjustment of the parameters of the used estimator may solve the problem but it would make the algorithm particularly inefficient (and maybe infeasible) on the low-resource nodes we consider, e.g., because an update of the used estimator requires the computation of a matrix inversion.

4 Problem Description

We consider a sensor network consisting of a finite number of nodes M. Each node is equipped with a *free running* clock source according to the recommendations in [5]. Instead of adjusting the parameters of the local clock, we build an *operational clock* on top of the free running local clock, like in [6].

A simplified clock error model indicates that at time t, measured according to a hypothetic absolute time reference, the clock of the node i indicates the time $c_{i,t} = (1 + \delta_i) \cdot (t - t_0) + \Delta_{i,0} + \Psi(t)$, where $\Delta_{i,0}$ is the initial clock offset at the time t_0, $\delta_i \in [-\epsilon_{clock}, \epsilon_{clock}]$ is a constant frequency offset and $\Psi(t)$ is an additional error which takes into account the dependency of the clock on the environmental conditions.

We want to minimize the length of the guard period by forcing all the nodes to have the same beacon interval. From this perspective, the global clock synchronization is a side effect of our procedure. Once a node is chosen as reference, all other nodes have to adjust the own beacon interval to the one of the reference.

Each node generates periodically (every T_{beacon}) a beacon signal, which indicates the beginning of a MAC frame. The beacon interval is measured using the internal clock and hence it differs from the nominal value.

A node is able to measure the beacon generation interval for each one of its neighbor nodes, with a precision of one symbol duration. The error due to signal propagation is much lower than the measurement error.

We assume that the sink is providing the gateway to a public network (e.g. GSM or UMTS) for remote storage and processing of the measurement data by the WSN operator. It is connected to the regular power supply and is also more powerful. Since it is equipped with a better clock, we want that all nodes generate the beacon signal at the same rate the sink does.

5 Proposed WSN Synchronization Algorithm

We consider the synchronization problem to be an estimation problem. A node n tries to estimate the beacon interval of the reference node using the measurements of the beacon interval of all observed neighbor nodes ($I^n_{i,k}$ denotes the measurement of the k-th beacon interval of the node i, taken by the node n).

The reference node generates its beacon signal exactly each $T_k = T$ seconds. The measurements done by the node n are affected by additive errors $\xi^n_{i,k}$.

At the time k, the node n computes the estimation of the reference interval $\hat{T}^n_k$ by using the previous observations and uses this value as length of the next beacon interval.

If we consider the network between the reference node and the node n as a blackbox (Fig. 1), we can write the following relation (1) between the reference interval and the measurement taken by the node n (in order to improve the readability of the formulas, we omit the indexes n and i from the notation).

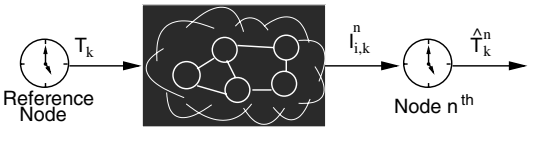


Fig. 1. Schematic representation of the blackbox approach

First we assume the node n having only one neighbor, then we introduce a simple extension in order to deal with a multiplicity of uplink nodes.

$$\begin{cases} T_{k+1} = T_k \\ I_k = T_k + \xi_k \end{cases} \tag{1}$$

The error ξ_k is the difference between the measured beacon interval and the beacon interval of the reference node. If the quantity ξ_k has the characteristics of white Gaussian noise (AWGN), it can be removed easily with a Kalman Filter (KF). In general, however, ξ_k is not AWGN.

A similar problem had been addressed in [9] for estimating the generation rate of ATM cells on the basis of the observation of their arrival time. However, it has to be noted that the origin of the measurement errors, and hence the characteristics of the error process, differs in the two cases. In the cited paper, ATM cells arrive with variable delays, which depend mainly on the queuing process in the intermediate nodes, while, in our case, the measurement error ξ_k is mainly due to the the adjustment process of the beacon generation time in all nodes between n and the reference and might experience sudden variations. In spite of these differences we argue that the prefiltering approach is useful in order to smooth the measurement error and make the problem tractable.

If ξ_k is AWGN the following approach is equivalent to the one proposed in [6].

As first step we smooth ξ_k by computing the moving average of the measurements taken during the last N intervals (2). The resulting signal, which has low-pass characteristics, is then modelled with an autoregressive process of the first order (AR(1)),which is one of the simplest low-pass models.

$$\bar{I}_k \;=\; \frac{1}{N}\sum_{i=0}^{N-1} I_{k-i} \;=\; T_k + \frac{1}{N}\sum_{i=0}^{N-1} \xi_{k-i} \;=\; T_k + \bar{\xi}_k \tag{2}$$

Now we can rewrite eq. 1 using eq. 2.

$$\begin{cases} T_{k+1} = T_k \\ \bar{I}_k = T_k + \bar{\xi}_k \end{cases} \tag{3}$$

We postulate that $\bar{\xi}_k$ can be modelled as AR(1) system as follows:

$$\bar{\xi}_k \;=\; a\,\bar{\xi}_{k-1} + r\,w_k \quad ; \quad w_k \in \mathcal{N}(0,1)\ i.i.d \tag{4}$$

We can compute the equations of the Kalman Filter (KF) for the given model with the colored noise according to eq. 4

initialization $(k = 0)$

$$\begin{cases} \hat{T}_0 = m_{T_0} - \frac{\sigma^2_{T_0}\left(m_{T_0} - \bar{I}_0\right)}{\sigma^2_{T_0} + r} \\ P_0 = \left(\left(\sigma^2_{T_0}\right)^{-1} + r^{-1}\right)^{-1} \end{cases} \tag{5}$$

$k \geq 1$

$$\begin{cases} G_k = \frac{H\ P_{k-1}}{H^2\ P_{k-1}+r} \\ P_k = (1 - H\ G_k)\ P_{k-1} \\ \hat{T}_k = \hat{T}_{k-1} + G_k\ \left(\bar{I}_k - a\ \bar{I}_{k-1} - H\ \hat{T}_{k-1}\right) \end{cases} \tag{6}$$

Where m_{T_0} and $\sigma^2_{T_0}$ are the expectation and the variance of T_0, respectively, $H \equiv 1 - a$, $\hat{T}_k$ is the estimation of T_k, $P_k$is the variance of the estimation error $\hat{T}_k - T_k$ and G_k is the gain of the KF.

To compute the KF we have to determine the parameters for the initial condition (m_{T_0} and $\sigma^2_{T_0}$) and the parameters of the autoregressive model (a, r).

The following description suggests how the former can be derived from the a priori knowledge about the clock, the latter can be estimated by using a small number of measurements collected before starting the synchronization procedure.

Initial conditions - Following the same reasoning as in [9] and assuming that for each node n the frequency f_n of the oscillator is uniformly distributed in the interval $[(1 - \epsilon_{clock})\ f_{nom}\ ,\ (1 + \epsilon_{clock})\ f_{nom}]$ and considering that $\epsilon_{clock} \ll 1$ we obtain $\sigma^2_{T_0} \approx 2/3\,(T_{beacon}\ \epsilon_{clock})^2$ and $m_{T_0} \approx T_{beacon}$.

AR(1) parameters - The stored measurements are used in order to compute an estimation of the autocovariance coefficients c_0 and c_1. By applying the covariance method we obtain $a = c_1/c_0$ and $r^2 = (1 - a^2)\,c_0$.

If a node observes the beacon signals generated by different neighbors, it uses the mean value of the measurements collected during the last beacon interval. This approach fits with our idea of considering the network as a blackbox. In fact, in absence of a-priori knowledge about the quality of the different measurements, this is the estimator which minimizes the square estimation error.

The continuous adjustment of the beacon generation interval modifies the model parameters an observer perceives. The system has to adapt itself to these changes and in particular it may be necessary to reinitialize the KF, as noted in [9], where the author suggested a continuous observation of the buffer state in an ATM switch in order to determine when the network conditions change and KF reinitialization was needed. In our model we try to follow the modification of the system parameters by continuously adjusting the parameters of the AR(1) model. At the same time we observe the mean absolute synchronization error (absolute value of the difference between the arrival time of a beacon and the corresponding expected value) and we reset the KF if this value starts growing.

An alternative approach based on using a second KF for the estimation of the model parameters seems to converge slower.

6 Simulation

The behavior of the proposed synchronization algorithm and its sensitivity to the parameter variations have been studied using simulation. A model of the

proposed algorithm has been implemented using the Omnet++ [10] simulation library, along with different models of the free running clock. Each node measures the time intervals using the own clock which is characterized by a precision in the order of the duration of a symbol (10^{-5} s for the analyzed network).

6.1 Clock Models

If the frequency offset is not constant, the time perceived by the i^{th} node can be expressed by the following equation

$$c_{i,t} = \int_{t_0}^{t} (1 + \delta_i(x))dx + \Delta_{i,0} + \Psi(t) \tag{7}$$

where t is the time measured by a hypothetical reference clock, $\Delta_{i,0}$ is the difference between the local and the global clock at time t_0 and $\delta_i(t)$ is the first derivative of the difference between the frequency of the local and the ideal oscillator. $\Psi(t)$ takes in account all other error components.

In order to identify the behavior of the proposed algorithm in response to different error conditions, we implemented the following four clock models:

Frequency offset (*FreqOff*) - The model considers only a constant frequency offset which is specific for each node: $\delta_i(t) = \delta_i \in [-\epsilon_{clock}, \epsilon_{clock}]$ where the clock tolerance ϵ_{clock} depends on the manufacturing process ($\epsilon = 40\,ppm$ for the simulation study). The additional error component $\Psi(t)$ consists only of the quantization error and is upperbounded by the symbol duration ($10^{-5}s$).

Aging - Since the expected lifetime of the considered network spans over many years, we modelled the effect on the clock stability due to the aging process of the electronic components assuming the frequency offset to be a linear function of the time ($\delta_i(t) = \delta_{i,0} + \rho_i \cdot t$). For a commercial quartz ρ may range between ± 5 and ± 10 ppm/year [11]. ($\rho_i \in [-10^{-4}\,ppm/s, 10^{-4}\,ppm/s]$ for this study).

Environmental changes (*EnvChange*) - Environmental parameters and in particular the temperature influence the frequency of a quartz driven oscillator. The nodes of an indoor network may experience significant temperatue variations over a relatively short time interval, especially during the cold season if the nodes are placed close to a radiator typical for e.g. a heat meter application. We try to reproduce this effect by means of a simple Markov model with two states (H,L). The transitions between the two states happen with rate λ_{HL} and λ_{LH}, respectively. The frequency offset is constant as long as a node does not change the state $\delta_i(t) = \delta_{i_{base}} + \delta_{i_{state}}$, where $\delta_{i_{state}}$ is specific for each state. It has to be noted that, from the point of view of the synchronization mechanism, this behavior is worse than the real case because it introduces a discontinuity in the frequency offset ($\lambda_{HL}^{-1} \in [500s, 1000s]$, $\lambda_{LH}^{-1} \in [1000s, 2000s]$, $\delta_{i_H} \in [-40ppm, 0]$, $\delta_{i_H} \in [0, 40ppm]$, $\delta_{i_{base}} \in [-40ppm, 40ppm]$ are the parameters used for the simulation runs).

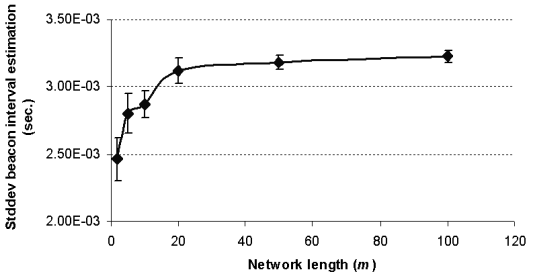
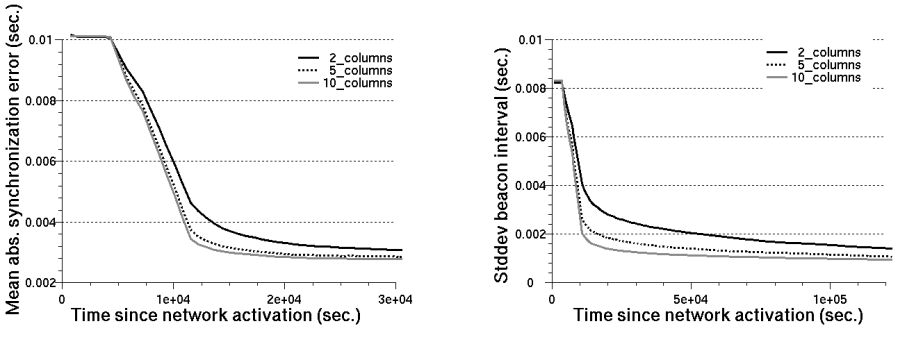


Fig. 6. Dependence of the SDBI metric on the network length m



(a) ASE metric (b) SDBI metric

Fig. 7. Behavior in presence of jitter for increasing number of nodes per level

Improvement Due to the Synchronization Algorithm. Table 1 summarizes the results for a network consisting of hundred aligned nodes.

The runs without synchronization algorithm have been done assuming that – upon reception of a beacon signal generated by a parent – a node simply updates the estimation of the arrival time of the next signal and leaves its own beacon interval unchanged. The length of the smoothing filter used by the synchronization algorithm has been set to N=8 points.

The synchronization algorithm causes a significant improvement of the observed metrics for all error models. As stated before the jitter error model represents a worst case for the synchronization algorithm. For the also critical EnvChange error model the synchronization algorithm is able to reduce the absolute synchronization error to 1% of the original one. However, the estimation of the beacon interval is only reduced to $\frac{1}{3}$. This is, however, not a shortcomming of the algorithm but is a consequence of the stochastic properties of the error process.

Comparison with the Algorithm of Servetto and Hu. As stated before the work of Servetto and Hu [8] has many similarities with our approach. It is therefore used for comparing our observations with their results. The graph in

Table 1. Comparison of the observations done using different clock error models, in a network with 100 aligned nodes (N=8)

	No Sync		Sync	
Clock error model	ASE-mean (sec.)	BI-stdev (sec.)	ASE-mean (sec.)	
FreqOff	9.5e-3	8.2e-3	5.7e-5	8.5e-4
Aging	2.3e-2	2.0e-2	4.4e-4	1.5e-3
EnvChange	4.9e-3	4.2e-3	4.7e-5	1.6e-3
Jitter	1.0e-2	8.2e-3	3.0e-3	1.5e-3

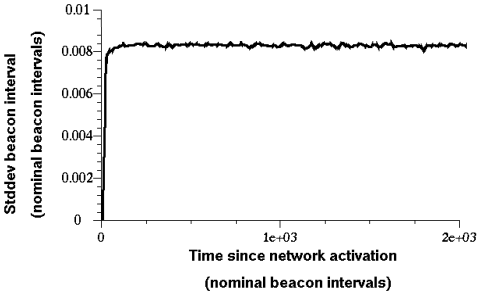


Fig. 8. SDBI metric in a grid network with 20x15 nodes, $_{jitter} = 0.1\mathsf{T}_{beacon}$

in Fig. 8 has been obtained averaging the observations done in an experiment consisting of 50 runs in a network with 300 nodes, which have been organized in a mesh structure with 15 columns and 20 rows. The standard deviation of the jitter amplitude has been set to 10% of the beacon interval like in [8]. The results have been normalized to the duration of a beacon interval in order to permit a direct comparison with the results in the cited paper.

The observed error parameter is – after convergence – less than 10 percent of the value reported in the reference paper. This indicates that our proposed algorithm performs better than the other one. However, differences in the clock error model do not allow to exactly quantify the improvement.

Contrary to the algorithm proposed in [8], the solution we propose does also not require any additional traffic in order to make the synchronization stable.

On the Implementation on Low-Resource Devices. An implementation of the proposed algorithm on a sensor node requires the availability of enough memory to store the data for the computation of the smoothing filter and of the coefficients of the AR(1) model. The number of data items which have to be stored is proportional to $N + P + 1$, were N is the number of points of the prefilter and P is the number of points considered for the computation of the autocorrelation values.

local computation is effective in reducing the impact of different clock errors and behaves better than the similar solution proposed in [8]. The algorithm may also be implemented in devices with severe RAM limitations by modifying the smoothing filter.

References

[1] Altan, N., Rathgeb, E.: Bootstrapping a very low power, beacon enabled, wireless sensor network. In: 12th IEEE Symposium on Computers and communications (ISCC) (2007)
[2] Mills, D.L.: Network time protocol (NTP). Network Working Group Request for Comments 958 (1985)
[3] van Greunen, J., Rabaey, J.: Lightweight time synchronization for sensor networks (2003)
[4] Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync protocol for sensor networks. In: SenSys 2003: Proceedings of the 1st international conference on Embedded networked sensor systems, pp. 138–149. ACM Press, New York (2003)
[5] Elson, J., Römer, K.: Wireless sensor networks: A new regime for time synchronization. Technical report, UCLA (2002)
[6] Hu, A.S., Servetto, S.D.: Asymptotically optimal time synchronization in dense sensor networks. In: WSNA 2003: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, pp. 1–10. ACM Press, New York (2003)
[7] Karp, R., Elson, J., Estrin, D., Shenker, S.: Optimal and global time synchronization in sensornets. Technical report (2003)
[8] Hu, A.S., Servetto, S.D.: Algorithmic aspects of the time synchronization problem in large-scale sensor networks. ACM/Kluwer Journal on Mobile Networks and Applications (2004)
[9] Kim, K.S., Lee, B.G.: Kalp: A kalman filter-based adaptive clock method with low-pass prefiltering for packet networks use. IEEE Transaction on communication 48, 1217–1225 (2000)
[10] Varga, A.: The omnet++ discrete event simulation system. In: European Simulation Multiconference (ESM 2001) (2001)
[11] Vig, J.: Introduction to quarz frequency standard. In: International Frequency Control Symposium and Exposition (2006)

service requests can be deduced implicitly depending on the service of interest.[2] Therefore, there is virtually no interference of the SbV mechanism in the format of existing SDP messages (see Subsection 3.3).

Our second assumption is that each node in the MANET hosts a local data structure (pendingList) used to control reply suppressions. In addition to reqID, numMaxReplies, and hopID, which are obtained from service requests, each entry of pendingList has a numReplies field (initially set to 0) that records the amount of replies overheard by the node, and an associated timer (cleanUp) that defines the lifetime of this entry in pendingList. Upon reception of a service request, a node records it as a pending request in pendingList before rebroadcasting it to neighboring nodes in the MANET. It is important to note that a rebroadcast service request has its hopID information (*i.e.* the source address field in its encapsulating packet) updated with the identification of the current rebroadcasting node, which allows neighboring nodes to keep track of the path traversed by the request in their local pendingList structures. This information will be used as the *return path* of corresponding replies towards the inquiring node (as explained in the following subsections), thus reducing the additional network load generated by ad hoc routing protocols.

3.2 The Proposed Algorithm

Figure 1 shows the pseudocode of the SbV mechanism as executed by each node as soon as it has received a reply. When a node receives a reply to a request it has previously originated (line 2), the node processes the message and does not forward it further in the MANET. If instead the reply is addressed to an inquiring node other than the receiver, the latter first checks whether there is an entry for the corresponding request in its pendingList (line 7). If so, the receiving node checks whether $N_R < N_M$, where N_R and N_M are (respectively) the values of the numReplies and numMaxReplies fields in the corresponding entry of its pendingList. If $N_R = N_M$, it means enough replies have already been sent towards the inquiring node, so the receiving node suppresses (*i.e.* discards) this reply. Otherwise, the receiving node increments the value of the numReplies field in the corresponding entry of its pendingList. It then compares its own identification with the value of retPath in the reply (line 10). If these values are equal, it means the receiving node is in the return path of the reply and hence can forward the message to the next node in the return path, as indicated by the hopID field in the corresponding entry of its pendingList (line 11).

Figure 2 illustrates the operation of the SbV algorithm. In the figure, only nodes w and z are within y's transmission range. Figure 2(a) shows the initial

[2] Note that all SDPs we have studied so far—with the exception of P2PDP [6] (see Section 4) and the work by Varshavsky *et al.* [2]—do not allow any control on the amount of replies per query nor automatic selection of the most suitable providers. Users must therefore manually select the service instances they are interested in from *all* received replies, possibly leading to bad selection (*e.g.* rashly selecting non-localized providers may increase inter-node interference in the MANET).

```
Require: msg, localID
 1: if firstCopy(msg) then
 2:   if myReply(msg) then
 3:     process(msg)
 4:     return
 5:   end if
 6:   entry ← pendingList [msg.reqID ]
 7:   if entry ≠ NULL then
 8:     if entry.N_R < entry.N_M then
 9:       entry.N_R ← entry.N_R + 1
10:       if msg.retPath = localID then
11:         forward(entry.hopID, msg )
12:         return
13:       end if
14:     end if
15:   end if
16:   discard(msg)
17: else
18:   ...{Deal with duplicate replies}
19: end if
```

Fig. 1. SbV pseudocode

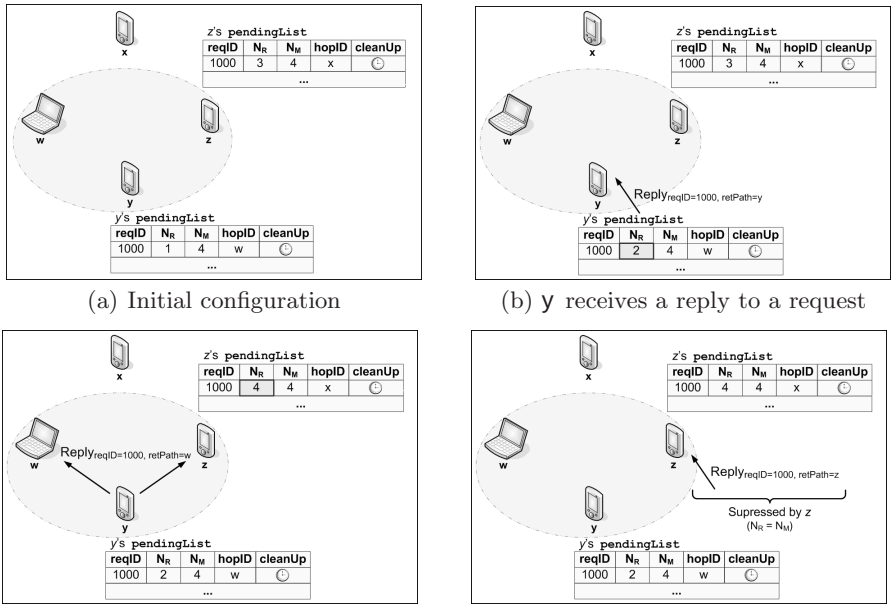
configuration of z and y's pendingList . In Fig. 2(b), y receives a reply to a request with reqID = 1000, and increments the value N_R of the numReplies field in the corresponding entry of its pendingList . As y is in the return path of the reply (Fig. 2(c)), y rebroadcasts the message towards w, which is y's next hop in the return path. z overhears such rebroadcast and also increments the value N_R of the numReplies field in the corresponding entry of its pendingList , but does not in turn rebroadcast that reply because it is not in the reply's return path. In Fig. 2(d), z receives another reply to the same request, but suppresses such a reply because $N_R = N_M$ in the corresponding entry of its pendingList .

To summarize, the SbV mechanism reduces the total number of replies conveyed in the MANET by eliminating unnecessary additional replies alongside the return path from replying nodes to the inquiring one. This alleviates the reply implosion problem, which is intrinsic of query-based SDPs.

3.3 Application-Level Forwarding Scheme

Using the SbV mechanism, the service replies are sent towards the inquiring node through application-level forwarding. There are two alternative mappings of this scheme onto the link level: using unicast or broadcast/multicast transmissions.

For link-level unicast mappings, the retPath value associated with replies is inferred from the destination address field in the encapsulating packets (*e.g.* the destination MAC address in IEEE 802.11 packets). This address field is filled with the value of the hopID field in the corresponding entry of pendingList (which indicates the link-level address of the next node in the return path), as



(a) Initial configuration

(b) y receives a reply to a request

(c) y rebroadcasts the reply towards w

(d) z receives another reply to the request

Fig. 2. Scenario illustrating the SbV mechanism

part of the forward operation (line 11 in Fig. 1). For a participating node to overhear replies from its neighbors, however, its network interface must work in promiscuous mode. Besides the security issues involved, this alternative has the drawback that, in promiscuous mode, the node must process the payload of *all* packets (not only those pertaining to the SDP) at the higher levels, which results in waste of resources (CPU, memory and energy) that are crucial to computational services.

For link-level broadcast/multicast mappings, nodes do not need to work in promiscuous mode; however, the destination link address field in packets encapsulating reply messages do not specify a single recipient, so an additional retPath field (with the link-level address of the next node in the return path) is needed in such messages. Further, a statement like $msg.retPath \leftarrow entry.hop\ ID$ must be added as part of the forward operation in Fig. 1; such a statement allows the receiving node to update the reply's retPath field with the value of the hopID field in the corresponding entry of pendingList, thus allowing the correct node to forward the reply to the inquiring node. As link-level broadcast/multicast mappings consume less computational resources, we have adopted them in our implementation of SbV for the P2PDP protocol.

It is worth noting that for MANETs in which the media access control is based on CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance), broadcast transmissions are less reliable and prone to collisions in comparison with unicast transmissions. This is mainly due to the lack of acknowledgments, RTS/CTS

(Request/ Consent to Send) dialogues, and a mechanism for collision detection. The problem of collisions in link-level broadcast transmissions may be rather alleviated if nodes are prevented from all replying at around the same time. Interestingly, the single-hop version of P2PDP already implements an algorithm in which replies from different collaborators are time-shifted, as discussed in the following section. Regarding the lack of acknowledgments, an implicit acknowledgment mechanism for broadcast transmissions could be used. To understand this, consider again the example of Fig. 2. When w receives the reply message from y (Fig. 2(c)), being in the return path, it will forward the message. Such a transmission will be overheard by y (as it is within w's range); y could then regard this transmission as a higher-level acknowledgement from w. Nonetheless, many subtle issues arise if a retransmission policy based on such implicit acknowledgments is devised to improve the reliability of the discovery protocol. We argue that such additional complexity is not worthwhile, as reply messages are always subject to suppression along the remaining path towards an inquiring node. In fact, the experimental results presented in Section 5.2 demonstrate that, in scenarios of pedestrian mobility, the discovery efficiency in the presence of the SbV algorithm is kept high even without such a retransmission policy.

4 Implementation

We have implemented our SbV mechanism as part of the P2PDP protocol [6]. Along this section, we give a quick overview of the protocol, emphasizing the points where changes were made to accommodate the SbV mechanism.

4.1 Peer-to-Peer Discovery Protocol

Nodes can play two main roles in P2PDP: *collaborators* or *initiators*. Initiators demand computational services from collaborators, which offer their resources—*e.g.* CPU cycles, memory and disk space—for the provisioning of such services. An initiator sends service requests (IReq messages) to the collaborators and, based on the received replies (CRep messages), define a list containing the collaborators that are more suitable to provide the service. Figure 3 depicts the format of IReq and CRep messages and illustrates an example of the protocol operation in multihop MANETs.

A collaborator adopts two criteria to decide whether it is able to provide the requested service. The first criterion acts as an admission control, checking weather the collaborator indeed offers the service (*e.g.* if it hosts a specific Web service or a Java virtual machine). The second criterion defines the suitability of the collaborator in providing the service. Crucially, the initiator maps the required service onto the amount of resources needed for its provision. The *context of interest*—indicated in the ctxtInfo field of IReq messages—allows the initiator to ask collaborators about the desired service, which resources are needed for the service provisioning, and the relative importance among such resources. The initiator also determines in the numMaxReplies field of IReq messages

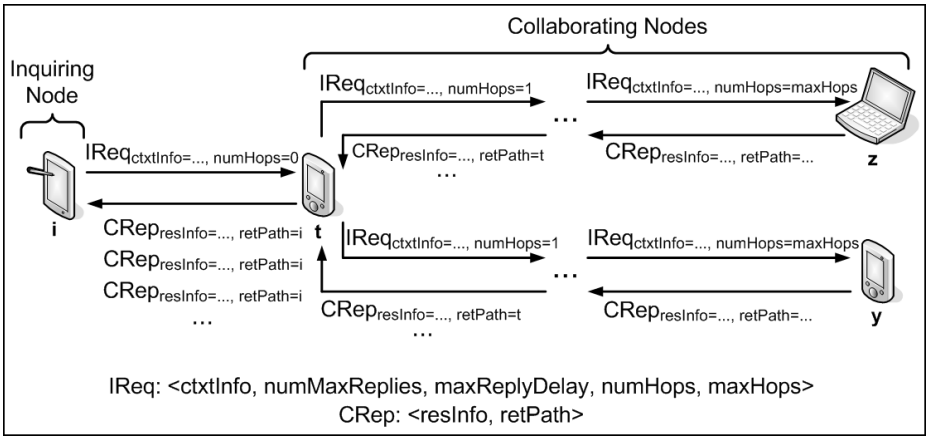


Fig. 3. Example of P2PDP messages

the number of service instances to be involved. Based on such information, a collaborator builds its CRep message, informing in the resInfo field the address of the service (*e.g.* a URL to a Web service or the network-level address of the node), and the resource availability related to the provisioning of such service.

We have introduced new fields in the IReq and CRep messages to allow the operation of P2PDP in multihop MANETs. The numHops and maxHops fields in IReq messages indicate respectively the current and maximum number of hops associated with such messages, and are used to constrain the diameter of service requests. The retPath field in CRep messages is used for forwarding such messages to inquiring nodes, and it is necessary due to the adoption of link-level broadcast transmissions in our application-level forwarding scheme, as discussed in Section 3.3.

4.2 Controlled Delay of CRep Messages

In the P2PDP protocol, each device willing to collaborate with the provision of a particular service delays the transmission of its CRep messages according to a timer. This timer is set to be inversely proportional to the availability of the required resources at the collaborating node. This way, nodes that are more resourceful reply earlier to service requests. If the total number of replies generated in the MANET is larger than the requested maximum number of replies N_M (which is set by the numMaxReplies field in IReq messages), the initiator selects the first N_M received messages as the most suitable replies. When a node receives a request, it gathers its current state in terms of the resources of interest for the given request to compute the reply delay. Importantly, all devices in the MANET must employ the same criterion for such computation. In the implementation of P2PDP, a collaborating nodes sets the reply delay to τ time units as given by

$$= \left(1 - \sum_{i=1}^{N} \left(\frac{{}_{i}\mathsf{P}_i}{\sum_{j=1}^{N} \mathsf{P}_j} \right) \right) \mathsf{D}_{\max} - 2\mathsf{HS}, \begin{matrix} 0 \leq & \leq 1 \\ 0 < & \leq 1 \end{matrix}, \tag{1}$$

where N represents the number of different resource types the collaborating node should take into account. P_i is the weight that describes the relative importance of each resource type i, $1 \leq i \leq N$. Both N and P_i are described as part of the ctxtInfo field in the request. α_i is the normalized level of current availability (in the interval $[0, 1]$) of resource type i at the collaborating node. $D_{\max}$ is the maximum reply delay, which is also obtained from the request (maxReplyDelay field). H and S are used for considering the transfer delays that IReq and CRep messages may experience. H is the distance in hops (obtained from the hopCount field in the IReq message) between the collaborating node and the inquiring node, and S is a tuning parameter representing the transfer delay at each transmission. Finally, ω indicates the willingness (also in the interval $[0, 1]$) of the collaborating node to participate in the resource provisioning. τ is undefined for $\omega = 0$; such a value means the user is not willing to participate, thus the collaborating node will not send replies. In this case, the node will only act as an intermediate in the message forwarding process.

We highlight that the delay reply mechanism provides a time shift in the transmission of replies, thus allowing for a reduction in the number of collisions of these messages when link-level broadcast transmissions are used.

5 Performance Evaluation

We carried out a set of experiments with the SbV mechanism. These experiments were conducted with two different simulators to evaluate two different aspects of our approach: scalability and discovery efficiency.

5.1 Scalability Analysis

We analyzed the scalability of the SbV mechanism using the ns-2 simulator [9]. All experiments in this simulator consider a fixed node density within the MANET (using topologies with a constant number of nodes within the same transmission range) so the impact of increasing the number of nodes in the MANET could be properly evaluated. The results presented in this section correspond to the average of a hundred sample runs per simulated scenario with a 95% confidence level. This analysis was mainly focused on the evaluation of two metrics: the number of reply messages in the MANET and the suppression diameter of these messages. Table 1 presents the parameters adopted in the simulated scenarios.

The average load of reply messages in the MANET was computed using, for each scenario, the mean number of packets involving these messages. Importantly, this metric also allows us to deduce whether there is a significant reduction in the energy consumption of devices in the MANET due to the suppression of replies, given that transmissions are known to be responsible for a high energy consumption. Using this metric, we compared two purely query-based SDPs: one in which service replies are sent by unicast to inquiring nodes (we called

Table 1. Parameters for ns-2 simulations

Parameter	Value
Number of nodes (N)	10 to 240
Percentage of collaborating nodes (p)	20% to 80%
Maximum number of replies (R)	1 to 10
Node density	5
Distance between nodes	10m

it UCast), and another in which replies are sent through application-level forwarding, with the SbV mechanism incorporated in the forwarding process. In both protocols, the inquiring nodes broadcast service requests by flooding, and no service announcements are employed. Figure 4 presents the number of reply messages as a function of the number of nodes for different percentages of replying devices. The vertical error bars indicate the confidence intervals. The results show that the adoption of the SbV mechanism allows for an increasing reduction—with respect to the UCast protocol—in the total number of transmissions, as the number of devices in the MANET increases. We also observe an even higher level of suppressions when there is a larger percentage of nodes (p) in the MANET with interest in collaborating on service provisioning. These results give a clear idea of the scalability that protocols adopting the SbV mechanism can achieve, such as in our implementation of P2PDP.

The suppression diameter of reply messages measures the distance (in number of hops) between the inquiring node and the nodes where suppressions occurred. This metric allows us to evaluate the degree of distribution of the load alleviation provided by SbV among the nodes in the MANET, and consequently the energy savings among the nodes due to the reduction in the amount of transmissions. Figure 5 presents the distribution of suppressions as a cumulative distribution function (CDF) for different numbers of nodes and percentages of replying nodes. To better illustrate the distribution of suppressions through the MANET, the results presented in Fig. 5 are contrasted with a uniform CDF (represented by the straight line in the figure). We observe a better distribution of suppressions as the number of nodes and the percentage of replying nodes (p) increase. Again, this suggests the scalability of our proposed approach.

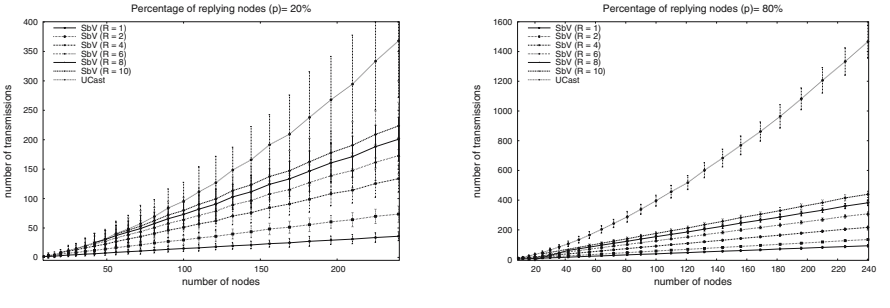


Fig. 4. Network load in the MANET due to reply messages

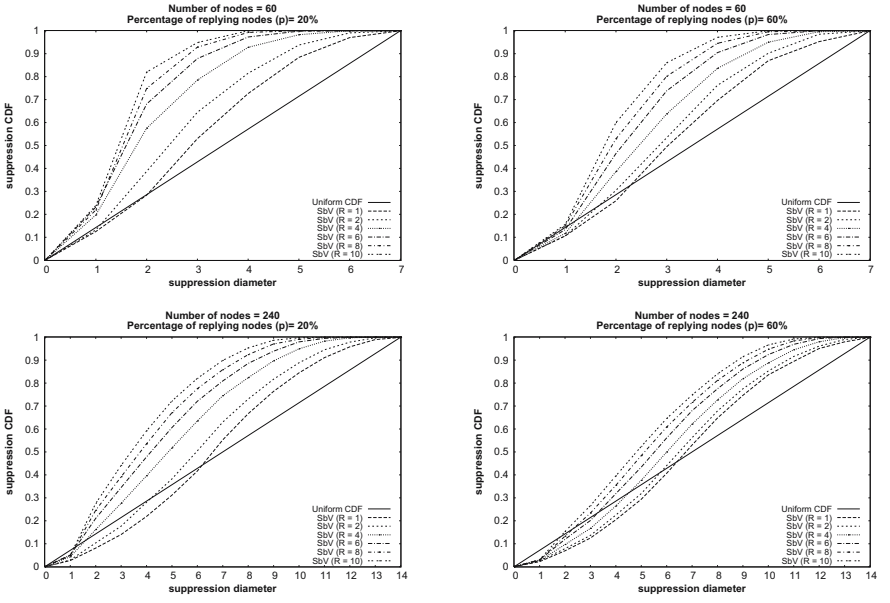


Fig. 5. Distribution of reply suppressions in the MANET

5.2 Discovery Efficiency

To observe the impact of mobility on the efficiency of the P2PDP discovery process using the SbV mechanism, we have implemented a modified version of this protocol for multihop MANETs, as well as a testing application to run on top of it. Both implementations were done in Java, using the CDC (Connected Device Configuration) J2ME profile as our reference platform. Our testing application consisted of a master-worker matrix-matrix multiplication program. For the purposes of our evaluation, we employed a very simple distributed multiplication algorithm: given matrices $\mathbf{A}_{m\times n}$ and $\mathbf{B}_{n\times p}$, a master node computes $\mathbf{C}_{m\times p} = \mathbf{AB}$ by selecting p worker nodes with the P2PDP protocol and sending to each worker node i ($1 \leq i \leq p$) a copy of matrix $\mathbf{A}$ along with matrix $\mathbf{b}^i_{n\times 1}$ (transposed vector whose elements are those of the i-th column of $\mathbf{B}$). Each worker node i computes matrix $\mathbf{c}^i_{n\times 1} = \mathbf{A}\mathbf{b}^i_{n\times 1}$ and returns it to the master node, which then builds each i-th column of $\mathbf{C}$ from $\mathbf{c}^i_{n\times 1}$. The selection of the worker nodes in the MANET that take part in the task is made by only considering those nodes with the most available CPU and memory resources—more specifically, $N = 2$, $P_{\text{CPU}} = 4$, and $P_{\text{mem}} = 1$ in Eq. 1.

We deployed our implementation in the NCTUns simulator and emulator [10]. To do so, we performed some changes to the underlying monitoring service that is part of the original P2PDP implementation. This service[3] is responsible for gathering information about the current state of a mobile node, including

[3] The monitoring service used by P2PDP corresponds to the implementation available at the MoCA architecture [11].

connectivity, CPU load, available energy and memory, and disk storage space. In the NCTUns platform, a single machine runs several (virtual) nodes interconnected by a simulated MANET, but with no kernel isolation between them. Thus, the use of the original monitoring service would lead to unrealistic scenarios in which all nodes in a simulated MANET would have the same state information. To tackle this, we have implemented a "fake" monitoring service that provides randomly generated state information for each different node in a simulated MANET.

The simulation scenarios consisted of 40 nodes placed in an obstacle-free, 500m X 500m area. The initial position of each node was set at random, with the constraint that at the beginning of the simulation the nodes formed a connected topology. The first scenario consisted of a stationary topology. In the remaining scenarios, the movement of nodes followed the random walk model. In such a model, each node moves in a random direction for some seconds—in a speed that is uniformly distributed in the range $]0, S_{max}]$—then chooses a new random direction, with no pause between the direction changes. This corresponds to a worst-case mobility scenario for each speed range. Table 2 summarizes the parameters adopted in the scenarios simulated with the NCTUns platform.

Table 2. Parameters for NCTUns simulations

Parameter	Value
Number of nodes (N)	40
Number of resource providers	10
Maximum number of replies (R)	4
Transmission range	100m
Maximum node speed (S_{max})	0 to 5m/s

The discovery efficiency for each simulation scenario was measured as a sample proportion calculated over 100 runs. Each run consisted of a single resource consumer issuing a single IReq message to a set of resource providers. The sample proportion indicates the percentage of runs in which the protocol delivered *at least* R replies to the resource consumer, as determined by the numMaxReplies field in the IReq message. The number of resource providers at each run was fixed to 10, which corresponds to 25% of the nodes in the simulated scenarios. Such a percentage was chosen based on the study by Hughes *et al.* [12], which states that in Gnutella—a famous P2P, collaboration-based file-sharing system—this percentage of participants is responsible for 98% of all service provisions.

Figure 6 presents the discovery efficiency of the P2PDP protocol extended with the SbV algorithm as a function of the maximum node speed (S_{max}). The vertical error bars correspond to the 95% confidence intervals for each sample proportion. The results show that the protocol behaves well under situations of human mobility (from 0.8 to 1.2m/s).

As it can be observed in Fig. 6, even for the stationary scenario ($S_{max} = 0$) the protocol does not reach 100% efficiency—the sample proportion is 92%, with

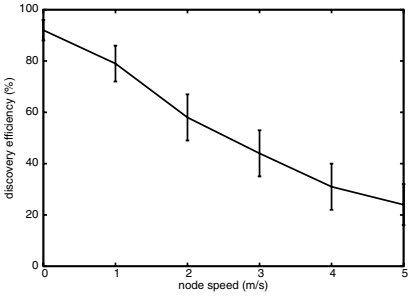


Fig. 6. Discovery efficiency in a mobile scenario

±4.13 confidence intervals. This is due to the drawbacks stated in Section 3.3 regarding the application-level forwarding scheme being mapped onto link-level broadcast transmissions in CSMA/CA enabled nodes.

6 Conclusions

In this paper, we have presented the design and implementation of a mechanism called Suppression by Vicinity (SbV) to reduce the implosion of reply messages in purely query-based SDPs for multihop MANETs. Our experimental results show that the proposed SbV mechanism is efficient in controlling the amount of service replies transmitted in the MANET. Moreover, the additional processing the SbV mechanism generates is well distributed among the nodes. In particular, this prevents greater energy drain rates on nodes nearby the inquiring node, thus promoting an indirect balance on energy consumption due to transmissions. Finally, the SbV mechanism behaves well in the mobile application scenarios we are interested in, which involves pedestrian (walking) mobility.

During the development of this work, some aspects have been identified for future investigation. The first one is the impact of the maxReplyDelay parameter on the efficiency of the SbV mechanism in the P2PDP protocol. Fine-tuning this parameter—*e.g.* as a function of the transmission delay of messages—is essential to reduce the discovery time without increasing the number of reply collisions, which is achieved through the asynchrony in the transmission of these messages. Still in this context, we believe it is important to investigate the influence of clock drifts among different equipment on the timers associated with the SbV mechanism and its implementation on the P2PDP protocol. A second point is that we have considered only low-mobility scenarios in our simulations. In more dynamic scenarios, the concept of return path the SbV algorithm uses for conveying reply messages is likely to reduce the discovery efficiency considerably. To deal with this, we are currently investigating alternative implementations of the SbV mechanism that automatically resort to using traditional ad hoc routing protocols whenever a failure is detected in the return path.

Adaptive MANET Routing: A Case Study

Liang Qin and Thomas Kunz

Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada
tkunz@sce.carleton.ca

Abstract. Node mobility plays an important role in the routing performance for MANETs. Many protocols provide parameters to adapt to different levels of mobility, but this is a global optimization (i.e., typically all nodes choose the same parameter values and they use these parameters throughout their participation in a MANET). We choose the monitored number of link breaks as key mobility metric and observe that the relative observable mobility varies widely for different nodes and over time for the same node. We utilize this (simple) mobility metric to allow a node using OLSR as routing protocol to dynamically adapt its behavior (changing the Hello Interval, selecting MPRs, etc.). Simulations with different mobility scenarios show that Adaptive OLSR can improve packet delivery ratio, reduce packet latency, and reduce routing overhead, especially in high mobility scenarios. As a general conclusion, we believe that designing adaptive routing protocols (protocols that change their behavior based on mobility and potentially traffic patterns) holds great promise in resource-constrained environments.

Keywords: MANET, routing, OLSR, mobility, simulation, NS2.

1 Introduction

A Mobile Ad Hoc Network (MANET) is defined by the MANET Working Group as "an autonomous system of mobile routers (and associated hosts) connected by wireless links - the union of which forms an arbitrary graph". Because of the antenna's limited transmission range, the nodes in the network may act as a router to forward packets to other nodes, and then a routing protocol is needed. The main characteristics of a MANET are:

- Packets may need to be forwarded by several nodes to reach the destination.
- Dynamic topology due to the nodes' mobility or nodes leaving/joining the network, which causes packet loss and route change.
- Resource constrains: wireless medium bandwidth, device's battery, processing speed and memory.

To obtain the correct network topology, frequent control message exchanges between nodes are required; on the other hand, these control messages will consume valuable wireless bandwidth resources. This tension poses a challenge for developing routing protocols. Existing MANET routing protocols basically can be classified as proactive

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 43–57, 2008.
© Springer-Verlag Berlin Heidelberg 2008

(table-driven), reactive (on-demand) and hybrid. Examples of proactive routing protocols are Destination-Sequenced Distance-Vector Routing (DSDV) [1] and Optimized Link State Routing Protocol (OLSR) [2]. Examples of reactive routing protocols are Ad hoc On-Demand Distance Vector (AODV) Routing [3] and Dynamic Source Routing (DSR) [4]. The Zone Routing Protocol (ZRP) [5] is a hybrid of proactive and reactive routing protocols. It applies proactive routing on a node's neighbors, and searches through the network using a reactive protocol. Detailed reviews and performance comparisons of these protocols can be found in [6] [7] [8].

The MANET routing protocol performance depends on the network conditions. For example, the simulation results in [9] show that under high network load proactive routing techniques outperform reactive routing techniques. Existing MANET routing protocols assume specific network conditions and preset certain parameters for all nodes. Because of the characteristics of a MANET, mobile nodes may experience a very dynamic environment over time, and different nodes may experience very different conditions at the same time. The term environment here not only refers to physical environment, which could impact on the transmission of wireless signal, but also includes the mobility of nodes, and traffic that is routed through nodes themselves or shares the wireless medium with the node. The dynamic nature implies that a node's environment changes with space and time. If nodes can apply routing parameters individually and be adaptive to the network environment based on observable metrics, the network performance might be improved.

The basic steps for adaptive routing consist of: monitor the current network characteristics based on some appropriate metrics; map these metrics to related routing parameters and adjust the parameters if necessary. In a MANET, the environment parameters that a node may monitor include the mobility of nodes in its neighborhood, the current number of flows or volume of traffic, the busy/idle time of the (shared) medium, the received signal strength, etc. These parameters impact the routing protocol performance in a number of ways. For example, high mobility typically causes frequent link breakage and invalid routes; high traffic on certain links will cause congestion; fluctuating signal strength makes route discovery and maintenance difficult. In this paper, first we propose a simple mobility metric that individual nodes can use to sense the mobility level changes around them. We then apply this mobility metric by redesigning OLSR so that nodes adjust their routing behavior individually. Our simulation results, using a range of mobility scenarios, show that this "Adaptive OLSR" protocol improves the routing performance in terms of packet delivery ratio, packet latency, and routing overhead.

The rest of the paper is organized as follows: Section 2 discusses the impact of mobility on routing protocol performance and how to adequately measure mobility with little overhead on a given node. Section 3 reviews related work and describes Adaptive OLSR, our case study for an adaptive routing protocol. Section 4 presents the simulation results, showing that enabling nodes to individually adapt their routing behavior in response to the locally observed mobility level does indeed increase protocol performance. Our conclusions and future work are listed in Section 5.

2 Mobility and Mobility Metrics

Packet loss is an important performance metric for MANET routing protocols. The main causes of packet loss are transmission errors, mobility and congestion. In this paper we focus on the mobility effect. A number of mobility metrics have been proposed in the literature and are used in the generation of mobility scenarios, such as node speed, pause time etc. They are useful for generating mobility scenarios for simulation purposes, but are not appropriate metrics for adaptive routing. For starters, link changes do not only depend on the mobility metrics of the node itself but also the (relative) speed of its neighbors. In addition, parameters such as "pause time" are mobility-model-dependent and therefore hard to generalize. A unifying mobility metric is proposed in [9]: "Mobility is defined as the average change in distance over time between all nodes (in m/s)."[9], which we use as well. By appropriately modifying the relevant mobility model parameters, we are thus able to generate mobility scenarios that show comparable levels of relative node mobility.

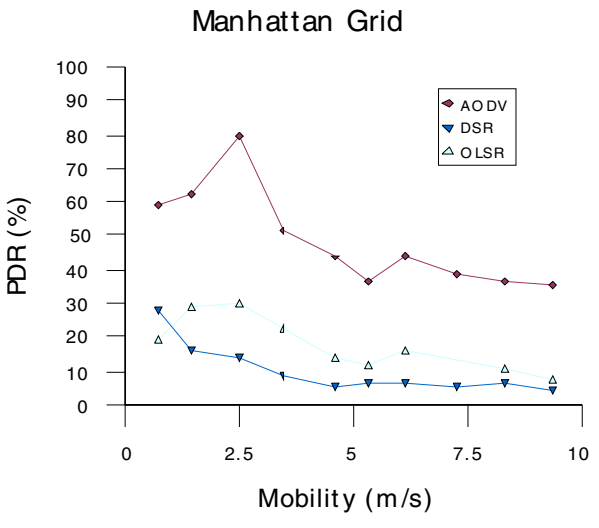


Fig. 1. PDR vs. Mobility for the MH Mobility Model

We first ran a series of simulations with different routing protocols and mobility models to explore the relationship between the overall performance and the mobility metrics. The purpose of these simulations is to (re-)confirm the impact of mobility on the performance of routing protocols. We experimented with two entity mobility models: Random Waypoint (RW) [8] and the Manhattan Grid (MH) [11], and one group model: Reference Point Group Mobility (RPGM) [12]. We conducted all our simulations in NS2 (the Network Simulator [13], which provides routing protocols such as AODV and DSR. In addition, we installed the UM-OLSR implementation [14] for NS2. The simulation area is 1000x1000 m, 25 CBR sources are sending 4 packets/s of size 64 bytes. Simulation time is 900 seconds. For each protocol (AODV, DSR, and OLSR), we repeated each run 5 times for each mobility scenario. We

generated and analyzed the mobility scenarios with BonnMotion [15], which can generate RWP, MH and RPGM model scenarios and compute statistical data on the generated mobility scenarios (including the average relative mobility). We set the total number of nodes in MH model to 170, and RWP to 80 (to achieve consistent node degrees). For the RPGM simulations, on average 5 nodes are in a group, the maximum distance from the center of the group is set to 25 m.

Figure 1 shows the packet delivery ratio (PDR) for AODV, DSR and OLSR for the MH mobility model with different mobility scenarios with increasing relative mobility. In all scenarios and for all mobility models AODV achieved the highest PDR, DSR has the worst performance, and OLSR falls somewhere in between. Also, we can see that the PDR has some correlation with mobility: when mobility increases, normally PDR decreases, but the rate of decrease (the sensitivity of the protocol to mobility) varies for different protocols in different mobility models. For example, we observed that the PDR of DSR, using the RW mobility model, decreases quickly when relative mobility exceeds 3m/s.

To allow a node to adapt to the level of relative mobility, it needs to monitor this parameter. Mobility metrics focus on a node and changes in its neighborhood. There are basically two ways to collect neighbor information: a mobile node can be equipped with some positioning device such as GPS and exchange its position information periodically; alternatively a node simply depends on exchanging "Hello" messages to sense the neighbors. In this paper we assume that mobile nodes do not have a positioning device, and only depend on message exchange to sense the neighbor changes. Based on this assumption, the *relative mobility metric* we used above to evaluate the overall protocol performance is not feasible because it requires that every node knows all nodes' positions and speeds all the time.

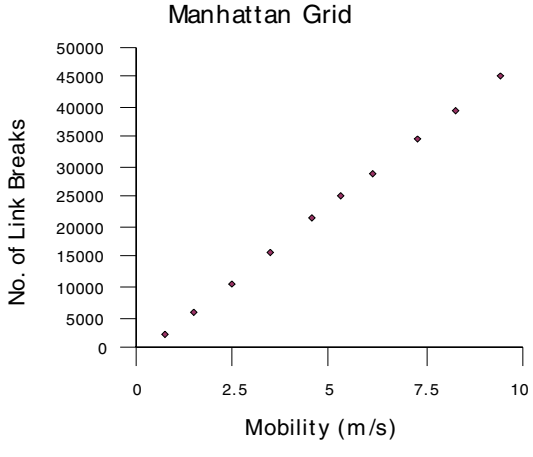


Fig. 2. Total Number of Link Breaks vs. Different Mobility Models

Link duration [10] is a mobility metric defined as the time period that two mobile are within transmission range. We explored the relationship between relative mobility and link duration and found that, in general, as mobility increases, the average link

duration decreases, indicationg that the links become less stable. However, *average link duration* alone does not accurately represent the current mobility status either. First, it is an average value; second, although we know that longer link duration means stable links, the exact value is mobility-model and network configuration dependent. Link duration may also need to be compared with historic data, which is the average value during a much longer period of time, in order to make judgments with respect to the mobility status.

As [9] observed, mobility has a good correlation with the number of link breaks. Our results in Figure 2 show a similar relationship (again showing only the results for the MH model, but a similar pattern shows for the other models). In addition, the *number of link breaks* is an easily obtained parameter from the routing table or by periodically exchanging Hello messages. From Figure 2, the total number of link breaks has nearly linear correlation with the relative mobility, which is strongly related to the protocol performance, so it is a good choice as mobility metric. However, some routing protocols such as DSR do not employ periodic Hello messages, in particular in networks where alternative mechanisms can provide indication of link failure (link-level callbacks, for example). However, in these cases alternative sensing mechanism based on promiscuous listening can be used instead. In DSR for example, to collect neighbor information, a DSR node can operate in promiscuous mode, monitoring packets that are not the destined for it to learn about new routes. So the node can periodically check its route cache to obtain the neighbor list (nodes one hop away), and monitor the change in the neighbor list over time.

As a final step, to confirm that nodes experience vastly different environmental conditions, Figure 3 shows the node degree for three randomly selected nodes over time for a mobility scenario generated with the MH mobility model at medium relative mobility. Different nodes experience very different neighborhood densities over time, and the number of neighbors of a node at any given point in time fluctuates widely as well.

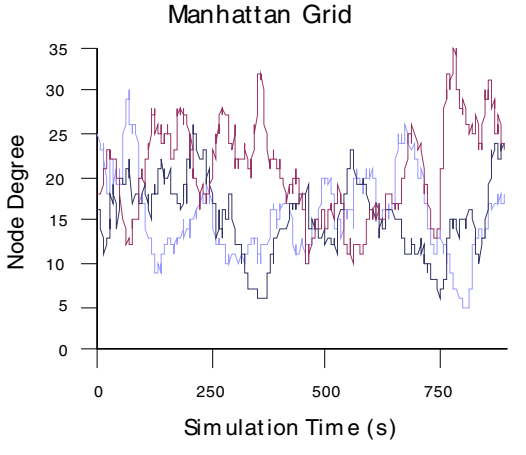


Fig. 3. Node Degree vs. Time for Three Randomly Selected Nodes

3 Adaptive OLSR

Most current MANET routing protocols preset certain parameters for a "typical" MANET scenario and apply the same parameters to all mobile nodes after protocol deployment. As hinted at by Figure 3, mobile nodes experience very different environmental conditions over time. Using a fixed set of identical parameter values will most likely not achieve the best possible routing protocol performance. The basic idea of adaptive routing is for each node in a MANET to adjust its routing behavior based on the sensed network environment around it.

3.1 Related Work

In [22][23], some simulations were run by varying HELLO_INTERVAL values for OLSR, and the same value is applied to every node in the network. The results show the tradeoff between packet delivery ratio and control message overhead. In the ARM (Adapting to Route Demand and Mobility) protocol [21], the rate of neighbor change is used as mobility metric. The routing messages contain a sender ID, update period and the sender's mobility metric. Each mobile node will average the mobility metrics of itself and its neighbors over time interval TW-SMOOTH and adjust the routing update period based on this average mobility value. The authors implemented ARM in DSDV, but only showed simulation results with two mobility patterns. In [17], the HELLO_INTERVAL of AODV changes according to the node mobility of its neighbors. The node mobility is determined by periodically checking the routing table, summing up the new and lost neighbors since the last check. This mobility metric will be used to decide the value of the HELLO_INTERVAL. The simulation results show that the packet delivery ratio and latency of adaptive AODV improve, but the improvement is rather limited and typically occurs only for scenarios with high node density or a high number of data sources. Fast-OLSR [18] [19] uses the number of neighbor changes as mobility metric. A node reduces its Hello-Interval when this metric reaches a predefined threshold. The papers only show simulation results for a network with 7 nodes, and without a performance comparison with the original OLSR protocol.

The purpose of Adaptive OLSR is to sense the link changes and adapt the routing behavior accordingly, increasing the protocol performance. We choose OLSR as an example for adaptive routing because it is a table-driven routing protocol and exchanges Hello messages between neighbors. It is therefore relatively straightforward to determine the number of link breaks and use it as mobility metric. Our Adaptive OLSR is inspired by Fast-OLSR [19], but with some major differences. First we use the number of link breaks as the mobility metric as discussed in Section 2. Second, in applying this mobility metric to OLSR, each node not only adjusts its HELLO_INTERVAL based on the mobility level it monitored, but also changes the MPR selection, a key component of the protocol. We conducted extensive simulation validation with different mobility scenarios. The simulation results show that our Adaptive OLSR can significantly increase packet delivery ratio, reduce packet latency, and reduce control message overhead.

3.2 OLSR Basics

OLSR is a proactive routing protocol; nodes exchange the topology information with other nodes in the network regularly. Every node will send a Hello message at least every HELLO_INTERVAL period. A node only broadcasts its Hello messages to its one-hop neighbors. The Hello message includes the list of a node's one-hop neighbors, and the corresponding link states, and is used for link sensing, neighbor detection and MPR (Multipoint Relay) selection signaling. Each node selects MPR nodes among its one hop symmetric neighbors; this set of MPRs will cover its strict two-hop neighbors. A node also declares its MPR set in its Hello messages, so that an MPR node can know the set of nodes that select it as their MPR, which is called MPR selector set of this MPR node. Finding the optimal MPR set is a NP complete problem, [2] proposes a simple heuristic for MPR selection. OLSR uses MPRs to optimize the flooding of control messages throughout the network, significantly reducing the number of retransmissions to reach every node. In addition, a node maintains a Link Set and Neighbor Set. The Link Set is populated with information about links to its neighbors and the Neighbor Set is updated according to the changes in Link Set. According to [2], the default HELLO_INTERVAL value is 2 seconds. Hello messages are used for link sensing. When node mobility is high, this default value may be too long, causing a node's MPR set or routing table entries to be inaccurate and resulting in packet loss. On the other hand, in low-mobility environments, there is no reason for nodes to frequently broadcast Hello messages, as the set of neighboring nodes is changing relatively slowly.

Each MPR node periodically broadcasts Topology Control (TC) messages, which includes links to its MPR selector set. TC messages are flooded throughout the network using the MPR optimization. Because all reachable nodes will select their MPR sets, all reachable destinations will be declared. Every node in the network uses TC messages to build a (partial) representation of the network topology, and to calculate the shortest paths to the destinations in the network. The protocol ensures that the partial knowledge is sufficient to determine the shortest path, and each path between two nodes is a sequence of MPR nodes.

3.3 Protocol Changes

The basic idea of Adaptive OLSR is that every node in the network adjusts its routing behavior based on the environment it experiences, which is the number of link breaks in this study. A mobile node will change its HELLO_INTERVAL according to the number of monitored link breaks. We define two HELLO_INTERVAL values: a default HELLO_INTERVAL of 2 seconds and a FAST_OLSR_HELLO_INTERVAL of 1 second. Each node checks its link table every second, and compares the number of its symmetric neighbors with the ones it stored when it checked last time. This allows it to determine the number of link breaks during this period of time. The node keeps records of link breaks over the past three seconds. When the number of link breaks reaches a threshold, a node will change its HELLO_INTERVAL to FAST_OLSR_HELLO_INTERVAL.

In addition, nodes adapt their MPR selection strategy. To model this, we introduce node states. Every node operates in one of three modes: *Default*, *Fast-Response*, and

Fast-OLSR, as shown in Figure 4. Nodes in different modes co-exist in the same network because they use the same message formats. Initially, every node is in *Default* mode, which refers to the original OLSR specification, exchanging control messages based on the default or configured protocol parameters. A node changes to *Fast-OLSR* mode once the number of link breaks reaches the UPPER_LINKBREAKS threshold, which we set to 2. In *Fast-OLSR* mode, a node changes its HELLO_INTERVAL to FAST_OLSR_HELLO_INTERVAL. On the other hand, when a node is in *Fast-OLSR* mode and the monitored number of link breaks is equal to or less than a lower threshold LOWER_LINKBREAKS, which is set to 1, for three consecutive periods, the node switches back to *Default* mode. The reason a node does not switch to default mode immediately once its mobility metric reaches the lower threshold is to reduce frequent mode switches.

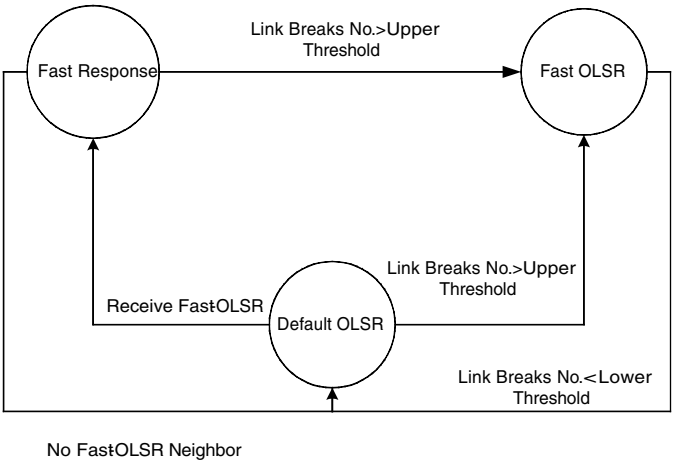


Fig. 4. Modes and Mode Transitions for Adaptive OLSR

The Hello message sent by a node in *Fast-OLSR* mode is called Fast-Hello message, which is in the same format as the default Hello message. However, the message only contains a node's MPR set and neighbors in *Fast-Response* mode. When a node in *Default* mode receives a Fast-Hello message, it switches to *Fast-Response* mode (which indicates that at least one of its neighbors is in FAST_OLSR mode, but not this node itself), changes its HELLO_INTERVAL to FAST_OLSR_HELLO_INTERVAL and sends empty Hello messages called OLSR_RESPONSE_FAST_HELLO_MSG. The purpose of the empty Hello is for the nodes in *Fast-OLSR* mode to sense neighbor changes quickly. A node in *Fast-Response* mode also sends regular Hello messages. To reduce the traffic overhead, we limit the node to only send one empty Hello message per second. A node *in Fast-Response* mode switches to *Fast-OLSR* mode when its number of link breaks is equal to or greater than the UPPER_LINKBREAKS threshold (same as a node in *Default* mode). On the other hand, when a *Fast-Response* node has not further neighbors in *Fast-OLSR* mode, it will switch back to *Default* mode.

Nodes in *Default* and *Fast-Response* mode select MPRs based on the heuristic in [2] and are candidates for being selected as MPRs themselves. Nodes in *Fast-OLSR* mode should not be an MPR node as they are experiencing rapid changes in their neighborhood. They therefore set their willingness to OLSR_WILL_LOW in their Fast-Hello message to avoid being selected as an MPR. In addition, a *Fast-OLSR* node only selects a limited number (currently up to 2) of MPRs, which should be neighbors in *Fast-Response* mode. We exclude neighbors in *Default* mode as potential MPR node because such nodes have not yet learned about the existence of this *Fast-OLSR* neighbors.

Every node in *Fast-OLSR* has an MPR set and an MPR candidate set. Every neighbor in *Fast-Response* mode but not in its MPR set will be in its MPR candidate set. When a node switches to *Fast-OLSR* mode, its first MPR set is built from its current MPR set with reduced size, the remaining MPR nodes are moved to the MPR candidate set if that MPR node is in *Fast-Response* mode. When a *Fast-OLSR* node receives a OLSR_RESPONSE_FAST_HELLO_MSG from one of its neighbors (which indicates that this neighbor is in *Fast-Response* mode), and this neighbor is not yet in its MPR or MPR candidate set, it adds it either to its MPR set (if it is not full) or to its MPR candidate set. When a neighbor in *Fast-Response* mode switches to *Fast-OLSR* mode and therefore becomes ineligible as an MPR, it is removed from either the MPR or MPR candidate sets. In the former case, a new node is moved from the MPR candidate set to the MPR set.

4 Simulation Results

Our adaptive version of the OLSR protocol is implemented using the UM-OLSR version 0.8.8 for NS2 version 2.29 (which we will refer to in the remainder of this paper as "Default OLSR"). In the following simulations, we used the Random Waypoint model, all the mobility scenarios are generated by the Random Trip Model Tool [20]. We summarize our mobility scenarios in the format speedMean-speedDelta-pauseMean-pauseDelta, based on the parameters for the Random Trip Model. For example, 10-5-1-1 is a mobility scenario with mean node speed of 10m/s, speed variation of 5m/s, mean pause time of 1 second and pause time variation of 1 second. The simulation area is 1000x1000m with 80 mobile nodes. The data rate is 4 packet/s with 25 data sources; each packet is 64 bytes in size. Simulation time is 900 seconds.

First we calculated the total number of link breaks for each mobility scenario during the simulation time. The calculation is based on a 250 m transmission range, comparing neighbors every second with the ones recorded a second earlier. Then we ran simulations in ns2 using the Default OLSR implementation with 5 cases for each mobility scenario and determined the average PDR. These values are shown in Table 1, showing as expected a strong correlation between PDR and the number of link breaks.

In a next step, rather than having nodes individually adjust their behavior based on the observed mobility level, we explored whether globally tuning protocol parameters can increase performance. The most relevant parameter related to mobility is the HELLO_INTERVAL, as this is the basis for link sensing in OSLR. We conducted a series of simulations with various global HELLO_INTERVAL values for the mobility

Table 1. Number of Link Breaks and Default OLSR PDR for Different Mobility Scenarios

Mobility Scenario	*3-2-1-1*	*10-5-1-1*	*12-6-5-2*	*15-4-10-5*
No. of Link Breaks	10422	35142	40202	46570
PDR (%)	94.98	81.07	74.07	64.02

scenarios listed in Table 1. The simulation results are shown in Table 2, where we varied the HELLO_INTERVAL from 1 second to 6 seconds in steps of 1 second. In all cases, all nodes use the specified Hello interval for the whole duration of the simulation. The results show that the PDR values change little when tuning the Hello-Interval globally, except for high mobility scenarios and for longer interval values. In these cases, the protocol performance (not surprisingly) deteriorated. In addition, with the exception of the most dynamic mobility scenario, a global HELLO_INTERVAL of 1 second performed slightly worse (on average) than the default value of 2 seconds.

Table 2. PDRs of Default OLSR with Different Global HELLO_INTERVAL Values

Hello-Interval (s)	*1*	*2*	*3*	*4*	*5*	*6*
3-2-1-1	92.43	93.37	93.83	93.66	93.40	93.19
10-5-1-1	81.54	81.95	80.81	80.45	78.08	77.11
12-6-5-2	77.49	78.30	76.62	77.09	73.69	72.47
15-4-10-5	65.98	64.62	64.46	64.01	61.61	60.03

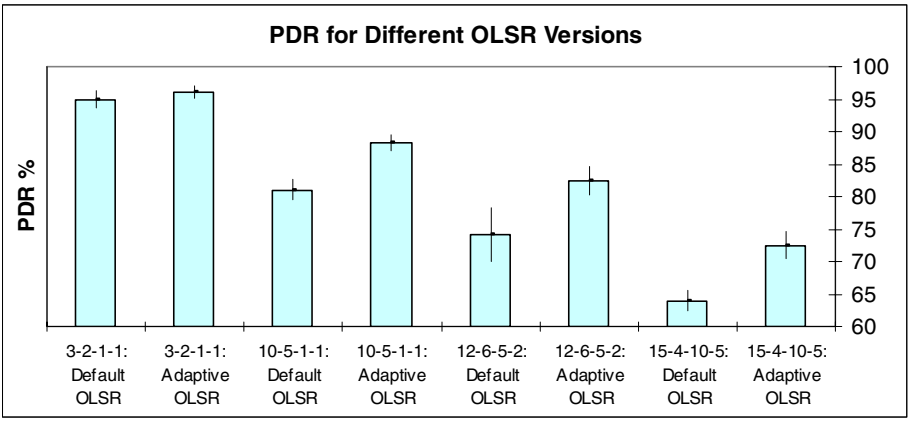


Fig. 5. PDR for Default OLSR and Adaptive OLSR

Figure 5 compares the simulation results for Default OSLR and Adaptive OLSR in terms of PDR, averaged over 5 cases for each of our mobility scenario, together with the 95% confidence interval for the average PDR. The results show that Adaptive OLSR consistently achieves higher PDR than Default OLSR, especially in higher mobility scenarios.

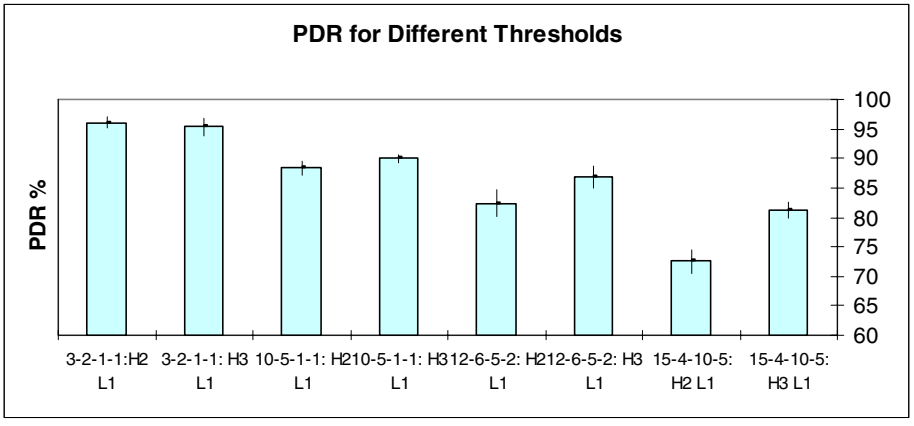


Fig. 6. PDR for Adaptive OLSR with Different Threshold Values

As discussed in Section 3, we defined two thresholds for Adaptive OLSR: UPPER_LINKBREAKS and LOWER_LINKBREAKS, which determine when a node switches in and out of *Fast-OLSR* mode. Initially, the values were set to 2 and 1 respectively. We also ran experiments where we increased UPPER_LINKBREAKS from 2 to 3. The results for both sets of threshold values are shown in Figure 6, together with the 95% confidence interval (Lx donates the value of LOWER_LINKBREAKS, Hx similarly donates the value of UPPER_LINKBREAKS). We can see that with the higher upper threshold, higher mobility scenarios can achieve even higher PDRs, though the lowest mobility scenarios suffer a slight degradation.

Table 3. Comparisons of Routing Performance Metrics

Mobility Scenario	OLSR Type	No Route	Link Broken	Control Message
3-2-1-1	Default	394	3269	183486
	H2, L1	357	2299	188753
	H3, L1	332	2824	192236
10-5-1-1	Default	609	13103	218829
	H2, L1	4294	4676	133841
	H3, L1	1859	5825	161861
12-6-5-2	Default	700	17232	228621
	H2, L1	8769	4952	124697
	H3, L1	3376	6685	150217
15-4-10-5	Default	801	22982	248283
	H2, L1	16743	4909	115316
	H3, L1	7438	7202	135047

Table 3 provides further details about the packet losses and routing overhead for Default OLSR and Adaptive OLSR. The first column describes the mobility scenario; the second column lists the protocol version (Default OLSR and Adaptive OLSR with different upper and lower threshold values). The numbers in the third and fourth column are the number of dropped data packets. A packet is either dropped because a node cannot find the destination address in its routing table ("No Route"), or because the link to the next hop broke ("Link Broken"). The last column shows the total number of protocol control message transmissions (sending and forwarding).

Table 4 summarizes the average packet latency for Default OLSR and Adaptive OLSR for the H2 L1 case. The values are averaged over all the data packets that reached their destination.

Table 4. Comparison of Packet Latency (in seconds): Default and Adaptive OLSR

Mobility Scenario	Default OLSR	Adaptive OLSR
3-2-1-1	0.0687	0.0427
10-5-1-1	0.2586	0.0681
12-6-5-2	0.4064	0.1043
15-4-10-5	0.580	0.1005

From these results, we draw the following conclusions:

– Using the number of link breaks as the mobility metric, a mobile node can adjust its routing parameter (HELLO_INTERVAL) to detect link changes quickly, thus reducing the number of dropped packets (Table 3). In conjunction with a change in the MPR selection, adaptive routing can significantly improve routing performance (in terms of both PDR and packet latency), especially for high mobility scenarios (Figures 5 and 6, Table 4).

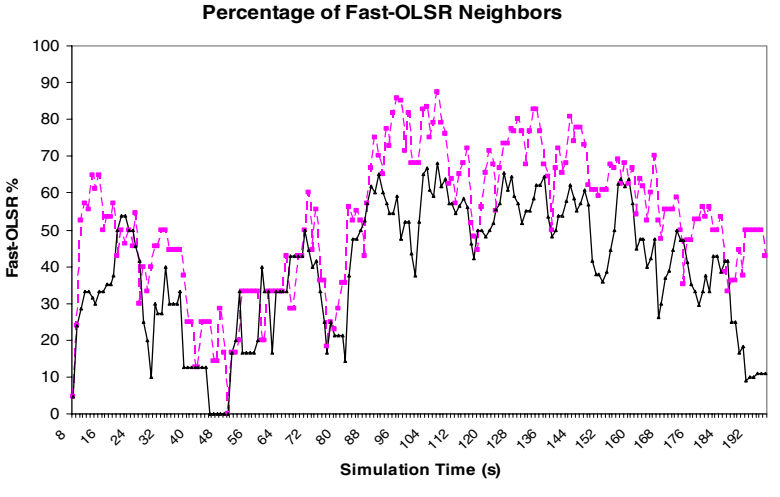


Fig. 7. Percentage of Neighbors in *Fast-OLSR* Mode

- The total number of control messages for Adaptive OLSR is almost always less than for Default OLSR. Nodes in *Fast-OLSR* mode select fewer MPRs, thus less TC messages are generated and flooded throughout the network (Table 3).
- Compared to Default OLSR, the number of dropped packets due to "No Route" increases. The relatively fewer TC messages in same cases prevent a node from determining routes. Increasing the upper threshold value will produce fewer *Fast-OLSR* nodes, generating more TC messages and reducing the number of packets lost due to "No Route". However, it also increases the number of packets dropped due to a lost link to the next hop (Table 3).

Figure 3 already visualized the highly dynamic neighborhood size of different nodes.To further indicate the highly variable dynamic environment a single node experiences, Figure 7 shows the percentage of neighbors in *Fast-OLSR* mode for a single node for different UPPER_LINKBREAKS values, during the initial 200 seconds of simulation and a mobility scenario with medium relative mobility. The solid line shows the H3 L1 case, the dashed line shows the H2 L1 case. Over time, a different percentage of neighbors experiences a high number of link breaks, operating in *Fast-OLS*R mode, while at the same time other neighbors monitor relatively more stable links and operate in the *Default* or *Fast-Response* modes. These figures graphically demonstrate the highly variable mobility environment from the point view of a single node over time. Figure 7 also shows the impact of changing the UPPER_LINKBREAKS value.

5 Conclusion and Future Work

In a MANET, a node's environment, such as its neighborhood, the traffic it carries, or the transmission conditions, are different for each node and also dynamic throughout time. However, most routing protocols assume some constant average network condition and predefine routing parameters for all the nodes in the network. In this paper, we focus on the impact of node mobility on routing performance, and choose the number of link breaks as mobility metric. Simulation results reconfirm that this metric correlates to routing protocol performance (for different mobility models and routing protocols). In addition, it can be easily measured. We apply this mobility metric to OLSR so that a node will reduce its HELLO_INTERVAL and change the MPR selection once the mobility metric exceeds an upper threshold. We conducted extensive simulations with the Random Waypoint mobility model; all results show that Adaptive OLSR can achieve better routing performance in terms of higher PDR, fewer control messages and reduced packet latency. We also show that tuning the mobility metric threshold can further improve the performance of Adaptive OLSR, especially in high mobility scenarios.

This case study confirms to us our general idea: allowing nodes to individually adapt their routing behavior to the dynamic environment they encounter can significantly improve the overall routing protocol performance. We plan to continue this work along a number of avenues. First, we are currently investing the performance of our Adaptive OLSR implementation over other mobility models. Second, a node's adaptive options are currently limited to changing its HELLO_INTERVAL values and the MPR selection. As part of future work, we will study the impact of adapting

additional routing parameters such as the TC-Interval, the MPR set size for *Fast-OLSR* nodes, etc. Finally, we also plan to apply the adaptive routing idea to other routing protocols such as protocols in the pro-active family of routing protocols.

References

1. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination-sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: Proc. ACM Special Interest Group on Data Communications (SIGCOMM), August 1994, pp. 234–244 (1994)
2. Clausen Ed., T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626, http://www.ietf.org/rfc/rfc3626.txt
3. Perkins, C.E., et al.: Ad Hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, http://www.ietf.org/rfc/rfc3561.txt
4. Johnson, D.B., et al.: DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In: Perkins, C.E. (ed.) Ad Hoc Networking, ch. 5, pp. 139–172. Addison-Wesley, Reading (2001)
5. Haas, Z.J.: A new routing protocol for the reconfigurable wireless network. In: Proc. 6th Int. Conf. on Universal Personal Comm., San Diego, USA, October 1997, pp. 562–566 (1997)
6. Royer, E.M., Toh, C.-K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications Magazine, 46-55 (1999)
7. Broch, J., et al.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In: IEEE/ACM Int. Conf. on Mobile Computing and Networking (MOBICOM), October 1998, pp. 85–97 (1998)
8. Das, S., et al.: Simulation-Based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks. Mobile Networks and Applications 5(3), 179–189 (2000)
9. Hoebeke, J., et al.: Towards Adaptive Ad Hoc Network Routing, http://www.ist-magnet.org/publications.html
10. Boleng, J., et al.: Metrics to Enable Adaptive Protocols for Mobile Ad Hoc Networks. In: Proc. Int. Conf. on Wireless Networks (ICWN 2002), pp. 293–298 (2002)
11. Bai, F., et al.: The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks. AdHoc Networks Journal 1(4), 383–403 (2003)
12. Hong, X., et al.: A Group Mobility Model for Ad Hoc Wireless Network. In: Proc. 2nd ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Seattle, USA, August 1999, pp. 53–60 (1999)
13. The Network Simulator NS2, http://www.isi.edu/nsnam/ns/
14. UM-OLSR, http://masimum.dif.um.es/
15. http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/
16. Qin, L., Kunz, T.: Mobility metrics to enable adaptive routing in MANET. In: Proc. 2nd IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob 2006), Montreal, Canada, June 2006, pp. 1–8 (2006)
17. Tan, H.X., Seah, W.K.G.: Dynamically Adapting Mobile Ad Hoc Routing Protocols to Improve Scalability. In: Proc. IASTED Int. Conf. on Communication Systems and Networks (CSN 2004), Marbella, Spain, September 1-3 (2004)
18. Benzaid, M., et al.: Integrating Fast Mobility in the OLSR Routing Protocol. In: Proc. 4th Int. Workshop on Mobile and Wireless Comm. Network (2002)

19. Badis, H., Al Agha, K.: Scalable Model for the Simulation of OLSR and Fast-OLSR Protocols. In: Proc. Med-Hoc-Net 2003 (June 2003)
20. http://lrcwww.epfl.ch/RandomTrip/
21. Anh, S., Shankar, A.U.: Adapting to Route Demand and Mobility in Ad Hoc Network Routing. Computer Networks 38(6), 745–764 (2002)
22. Stanze, O., et al.: Mobility adaptive self-parameterization of routing protocols for mobile ad hoc networks. In: Proc. IEEE WCNC, Las Vegas, USA, pp. 276–281 (2006)
23. Voorhaen, M., Blondia, C.: Analyzing the Impact of Neighbor Sensing on the Performance of the OLSR protocol. In: Proc. 4th Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, April 2006, pp. 1–6 (2006)

Self-interference in Multi-hop Wireless Chains: Geometric Analysis and Performance Study

Saquib Razak and Nael B. Abu-Ghazaleh

Dept. of Computer Science
State University of New York, Binghamton
and
School of Computer Science
Carnegie Mellon University - Qatar
{srazak,nael}@cs.binghamton.edu

Abstract. In the presence of interference, two single hop links can interact in a number of different ways, exhibiting significantly different behavior. In this paper, we consider the impact of these two-flow interactions on multi-hop chains. Specifically, we characterize the different types of interactions that arise in chains between hops that do not share a common node. We develop closed formed expressions to estimate the probability of occurrence of these interaction combinations. We use simulation to characterize the performance of the most common types of chains. We make a number of interesting observations: (1) the most destructive types of two-flow interactions do not arise commonly in chains; (2) the throughput of chains does not vary significantly with the types of arising interactions, because of the self-regulating effect of packets in the chain (later hops can only transmit when they receive packets from earlier ones); however, (3) the chains exhibiting destructive interactions suffer frequent collisions and require many more retransmissions. As such, in general scenarios, such chains reduce the available bandwidth within the network.

1 Introduction

Chains are fundamental in multi-hop wireless networks; however, our understanding of their behavior is limited. In multi-hop wireless networks, connections are made across chains of nodes. A chain is a sequence of nodes that a packets travels in order to go from a source node to a destination. The performance of chains is affected both by self-interference (different nodes in the chain transmitting different packets concurrently) [7, 10], as well as interference from other chains. However, due to the complexity of wireless interference our understanding of the behavior of chains remains limited. More accurate characterization of chain behavior, and understanding of what makes an effective or poor chain, is critical for designing routing, QoS and traffic engineering protocols for multi-hop wireless networks.

The CSMA MAC protocol relies on imperfect carrier sense to reduce collisions, which can, even in simple scenarios, lead to a number of different interaction

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 58–71, 2008.
© Springer-Verlag Berlin Heidelberg 2008

modes some of which exhibit inefficiency and short or long term unfairness. Carrier Sense Multiple Access (CSMA) based MAC protocols like IEEE 802.11 are widely used in multi-hop wireless networks. CSMA MAC protocols suffer from imperfect medium access, giving rise to a class of problems generally called hidden terminal problems [4]; we discuss CSMA MAC protocols in more detail in Section 2. Recent studies have shown that even with a simple scenario of two contending single hop flows, a number of different interaction modes arise [1, 6, 9]. We discuss these *two flow* studies and other related works in Section 3.

Self-interference in chains differs significantly from the two flow interference modes that have been previously studied. The structure of the chain changes the probability of occurrence of the different interaction modes. In addition, the dependent nature of traffic in chains leads to different behavior than that of independent traffic sources.

This paper contributes the following: (1) Classification of the types and frequency of occurrence of chains with respect to self-interaction among the hops; (2) Analysis of the performance of the types that most commonly occur in a 4-hop chain and generalization of this analysis to n-hops. Based on this analysis, we discover the following: (1) Some of the most destructive interaction modes rarely occur because of the structure of the chain (asymmetric interference from an upstream hop to a downstream one); (2) In isolation, there is little difference in throughput obtainable by the different chain types because of the self-dependent nature of the traffic. However, some chains suffer from persistent packet collisions, leading to a large number of retransmissions and therefore, significantly poorer throughput in a general network; (3) Existing routing protocols often pick poor quality chains with respect to self-interference, even in the presence of high quality ones. This places emphasis on mechanisms for detecting destructive self-interference and using that information in routing protocols. We discuss ideas for such mechanisms which form a part of our future work.

2 Background–MAC Protocol

In this section, we first briefly review the channel access mechanism and the IEEE 802.11 MAC protocol. We then discuss the modes of interactions that arise among two single hop interfering hops. The goal of this paper is to identify the impact of these interaction modes on a wireless chain.

2.1 IEEE 802.11

The signal power of a wireless transmission attenuates with distance and other environmental factors. A packet is successfully received if the signal strength at the receiver is above the receiver sensitivity threshold. Furthermore, the ratio of the signal to noise and interference power must be above the capture threshold. The *Boolean physical model* is a simplified model of this operation, where a transmission from a node can be sensed by all the nodes that are within a given *Interference Range*(R_i). In the absence of interfering signal, a packet can be

received by all the nodes that are within the *Communication Range* (R_c, where $R_c < R_i$). Under this model, packet collisions occur when a node is receiving a packet and an interfering node (within a distance of R_i from the node) transmits a signal.

The MAC layer protocol regulates access to the channel in an attempt to reduce collisions. IEEE 802.11 uses a Carrier Sense Multiple Access approach, augmented with Collision Avoidance (CSMA/CA). Difficulties arise in wireless settings because carrier sense is carried out at the sender, while correct reception requires the medium to be idle at the receiver. Therefore, IEEE 802.11 optionally uses small control packets to arbitrate the medium (Request to Send sent by the sender before a transmission and Clear to Send sent in response by the receiver if the channel is idle near it) to attempt to reduce collisions. However, since these packets can only block interferer in reception range (those outside cannot receive them), they are of limited use in preventing collisions. Finally, in response to a correctly received packet, the receiver sends an acknowledgement. If the acknowledgement packet is not received, the receiver attempts to retransmit.

Despite aggressive carrier sense, collisions can still occur between senders that are outside carrier sense range of each other, but that are in interference range of their respective receivers. To regulate the load in the presence of collisions, senders maintains a backoff window (BO) counter. When a collision occurs, the CW is doubled (up to a maximum limit), a backoff algorithm known as Binary Exponential Backoff (BEB).

2.2 Two Flow Interaction Modes

It has been recently shown that a number of different interaction modes arise among two interfering single hop flows [1, 9]. In a two flow scenario, two senders S_1 and S_2 communicate with two receivers D_1 and D_2 respectively. There exist four secondary (or cross-flow) channels that lead to the different modes of interactions; these are S_1S_2 S_1D_2, S_2D_1 and D_1D_2. The connections interfere differently depending on the state of these four secondary links. In this paper we assume Carrier Sense range and Interference range to be the same. Studying interactions with different Carrier Sense and interference ranges is part of our future work. Under a boolean interference model, the interactions can be grouped into five categories as described below [9].

Sender-Connected Symmetric Interference (SCSI): This category includes all scenarios where the two senders are in range and there is symmetric interference between opposite source and destination. An example of this scenarios is shown in Fig 1(a). In this scenario the channel is shared equally among the two flows.

Sender-Connected Asymmetric Interference (SCAI): In SCAI, senders are in communication range and only one destination is in interference range of the other sender. Fig 1(b) shows an example of this scenario. An ACK sent by $D2$ is received by $S1$ as a corrupted packet. $S1$ assumes that it is a DATA

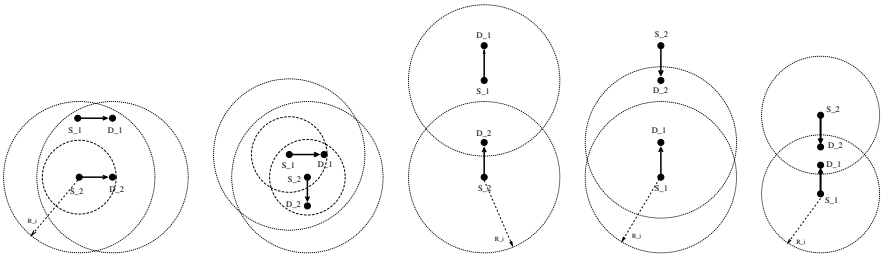


Fig. 1. Sample scenarios in each category

packet and defers for an Extended Inter Frame Separation (EIFS) period while $S2$ defers for the standard DIFS. Since EIFS is much longer than DIFS, $S2$ wins the channel most of the time. Hence SCAI exhibits severe unfairness problems.

Asymmetric Incomplete State (AIS): In the remaining scenarios the senders are not connected (Incomplete State) and carrier sense cannot prevent collisions. In Asymmetric Incomplete State, as shown in Fig 1(c), only one of the senders interferes with the other destination and only one of the flows experiences collisions, giving rise to unfairness.

Symmetric Incomplete State (SIS): In this category, the senders are not connected and both senders can interfere with the other destination. Fig 1(e) shows an example of this kind of interaction. This causes drops at both destinations and severely affects the throughput of both flows.

Interfering Destinations Incomplete State(IDIS): In this mode only destinations are in range as shown in Fig 1(d). The ACK sent by one destination interferes with packets being received by the other causing packets to be dropped. This scenario affects the throughput of both links.

In this paper we study the existence of different interference groups in a multi hop chain and its effects on chain throughput and goodput. We denote the absence of any interaction between two hops as NI (No Interaction).

3 Related Work

Analysis of throughput in chains has been studied extensively. Authors in [2, 3, 5, 7] compute the theoretical upper bounds on throughput of multi-hop ad hoc network. In [11], the authors evaluate the performance of TCP over a mulithop chain. They demonstrate that TCP traffic in a chain has instability problems that degrade the throughput of the chain.

In [8] the authors present a hop by hop analysis of a multi-hop chain and study the effects of hidden nodes on the throughput of a chain topology. They present a quantitative approach towards estimating the throughput of a chain. They provide two main observations about flows in a chain. Firstly the presence of hidden nodes cause packet drops that reduce the throughput of the chain

directly, and secondly packet drops cause reporting of broken links to the routing protocol and hence reducing the throughput indirectly.

Our observations in this study show that in a four-hop chain, packet drops have very little effect on the throughput of a chain both directly by extra transmissions or indirectly by way of rerouting because of false link breakage information. Extra lost transmissions come at a cost of decreased goodput of a chain hence introducing extra noise in the network. We also extend this analysis to an n-hop chain and conclude that chain interactions do not play vital role in determining the throughput but effect only the goodput of the chain. Cross chain interference is effected more by these interactions since different chains produce similar throughput but very different overall transmission levels.

Most of the studies are focused on finding the macro level behavior of chains in order to estimate the overall throughput of the network. Our study is focused on the micro level interactions in a multi-hop chain between different hops in order to better understand the interference present in a chain topology. In this paper we study the patterns of self interference in a chain. We feel that a better understanding of self interference is critical in understanding cross interference between chains.

4 Chain Self Interference

Links in a chain topology exhibit different modes of interference among hops, leading to significant impact on performance. We use, as the base for classifying the different chains, the two flow interference modes presented in an earlier work [9]. Given the restrictions of chain connectivity, the probability of the different cases changes. Moreover, given the nature of the traffic, it is likely that the impact of these modes will be different as well.

4.1 3-Hop Chains

Figure 2 shows a chain with 3 hops. In this chain, hops H1 and H3 are two link level flows within this chain that interact with each other according to the probabilities shown in Figure 3.

The plot in Figure 3 is obtained by creating different 3-hop chains using a Monte Carlo approach and then analyzing the existing interference interactions amongst the flows. It can be seen from the plot that at typical carrier sense/interference range of more than twice the communication range, only SCSI interactions are possible. But at lower ratios of carrier sense to communication the AIS group is the dominant interaction. AIS interaction has much lower throughputs than SCSI groups in two flow settings [9] but increasing the carrier sense range in a network exacerbates the exposed node problem.

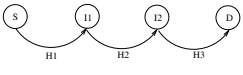


Fig. 2. A Chain with 3 hops

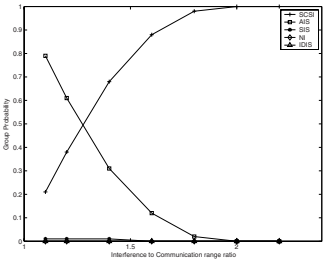


Fig. 3. Interaction Probabilities for a 3-Hop Chain

4.2 4-Hop Chains

A 4-hop chain as shown in Figure 4 presents more interesting problems. In this chain we have three different sets of two flow links that can interfere; note that links that share a node cannot be active at the same time and hence do not interfere with eachother. Node S can potentially transmit to Node I1 at the same time when Node I3 is transmitting an older packet to Node D. This makes hops H1 and H4 one set of simultaneous flows. Similarly H1 and H3, and H2 and H4 make up the other sets of two-flows. Hence in 4-hop chain we can have three different groups of interactions between the three sets of flows.

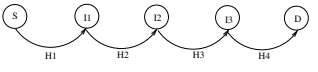


Fig. 4. A Chain with 4 hops

Mathematically there can be 5^3 kinds of interactions in a chain. To determine those interactions that are probable in a 4-hop chain topology we perform an exhaustive enumeration of all possible scenarios. More specifically, we fix the location of the source node S and move node $I1$ around it in a circular disc starting from radius 0 to a radius of Communication Range (250m in this case). Then we move node $I2$ around $I1$ in a circle making sure that $I2$ does not enter the communication range of node S. Similarly node $I3$ is moved around $I2$ and the destination D is moved to all possible locations around $I3$. For each position of these five nodes, we evaluate the scenario that occurs in this chain. The following interactions occur non-negligible percentage of times in a chain. The interactions are referred to in the format A/B/C where A is the interaction between H1 and H4, B is the interaction between H1 and H3 and C is the interaction between H2 and H4.

1. SCSI/SCSI/SCSI
2. AIS/SCSI/SCSI
3. NI/SCSI/AIS
4. AIS/AIS/SCSI
5. NI/AIS/AIS

Fig 5(a) plots the occurrence probabilities of the scenarios as carrier sense range is increased.

4.3 Geometric Models

We develop geometric models for computing the probability of occurrence of the five chain interactions listed above. Here we present the complete derivation of NI/AIS/AIS group while the rest of the groups are derived in a similar fashion.

NI/AIS/AIS. In this set of interactions nodes S and I1 are out of range of Nodes I3 and D. This Scenario has AIS interaction between H1 and H3, which requires that I1 to be out of range of I3 (Source of H1 is out of range of destination of H3). The last interaction AIS between H2 and H4 requires I1 to be out of range of Node D. This is already implied by the NI interaction between H1 and H4. For this chain we find the probability that for a given distance between nodes S and I1, I2 lies in an area that is outside the area of interference of S. Also given the distance between I1 and I2, we find the probability that I3 lies outside the area of interference of I1.

The derivation uses the following terminology: interference range and communication range are represented by r_i and r_c respectively. $C(X)$ refers to the area of communication range of Node X (circle of radius r_c around X) and $T(X)$ refers to the interference range of Node X (circle of radius r_i around X).

The probability that I1 is on a circle of radius x around S where x is always less than r_c is given by

$$p_1 = \int_0^{r_c} \frac{2x}{r_c^2}\, dx \tag{1}$$

Next we find the probability that I2 is out of range of S. Lets say that I2 is on a circle of radius y from I1. The arc length of circle with radius y around I1 that is intersected by circle with radius r_i around S gives us the portion of circle y that is within range of S. Subtracting this arclength from the perimeter of circle y will give us the portion that is out of range of S.

The minimum value of y has to be $r_i - x$ to guarantee that some portion of the circle is out of range of S. The maximum value of y is r_c.

$$p_2 = \int_{(r_i - x)}^{r_c} (2 * pi * y) - 2ycos^{-1}(\frac{y^2 + x^2 - r_i^2}{2xy})\, dy \tag{2}$$

Now we calculate the probability that I3 is out of range of I1. I3 has to be within the communication range of I2. We find $AreaR_iR_cy$ the area of intersection of circle r_i around I1 and r_c around I2 given the distance y between I1 and I2,. This is the portion of Communication range around I2 that is within range of I1. Subtracting this common area from $C(I2)$ gives the area that is out of range of I1. Let $AreaR_cR_cy$ be the area of intersection of circle of radius r_c around I1 and I2. Subtracting this area from $C(I2)$ will give us the area which is within communication range of I2 but outside the communication range of I1. Note

that since I3 is the next hop for I2, it can only be in $C(I2)$ - $AreaR_cR_cy$. Hence probability of I3 being out of range of I1 is given

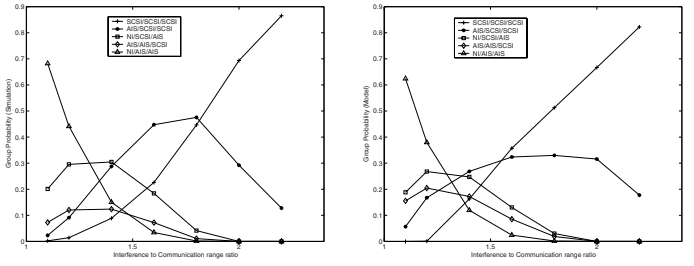
$$p_3 = \frac{C(I2) - AreaR_iR_cy}{C(I2) - AreaR_cR_cy} \tag{3}$$

The overall probability of NI/AIS/AIS is calculated by multiplying Eq[1,2,3]

$$p = \int_0^{r_c} \int_{(r_i - x)}^{r_c} p_3 \frac{2x}{r_c^2}((2 * pi * y) - 2ycos^{-1}(\frac{y^2 + x^2 - r_i^2}{2xy}))\ dy\ dx \tag{4}$$

4.4 Model Validation

We validate the geometric models for each interaction by comparing against exhaustive enumeration of all interactions in a chain.



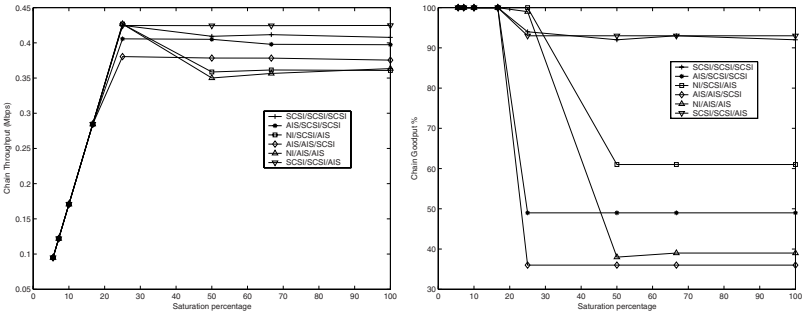
(a) Simulated (b) Predicted

Fig. 5. 4-Hop Interaction Percentages

Figures 5(a), and 5(b) show the probability of each interaction obtained using enumeration, and geometric model prediction. The plots indicate that the models closely match the results of simulation as the ratio of interference range and communication range is increased. As the ratio increases, the interactions move towards having all interacting hops Sender Connected. At lower ratios we have an increased percentage of interactions with hidden terminals.

5 Simulation Study of throughput

In this section we analyze the throughput of a 4-Hop chain under different interactions using NS2 Network Simulator. We use a fixed distance of 250m for transmission range and disable RTS/CTS mechanism. All transmissions are based on 802.11 DCF mode at data rates of 2Mbps and packet size of 1000 bytes. We change the saturation level of the channel by altering the rates at which the source pumps Constant Bit Rate (CBR) packets into the chain. We perform this analysis using the standard two-ray ground wireless propagation model which results in fixed communication and interference/carrier sense ranges.



(a) Throughput (b) Goodput

Fig. 6. Throughput and Goodput of a 4-hop Chain vs Channel Saturation

Fig 6(a) shows the throughput achieved at different saturation rates for the different chains. The throughput of a chain increases as we increase the saturation rate until it reaches an asymptotic limit. The limit represents the highest throughput possible in a chain topology as has been determined to be 1/4 of total bandwidth [7, 8, 11]. Figure 6(b) shows the percentage goodput of each chain. Goodput in our case is calculated as percentage of packets that are successfully transmitted. We analyze these plots of each chain separately.

5.1 SCSI/SCSI/SCSI

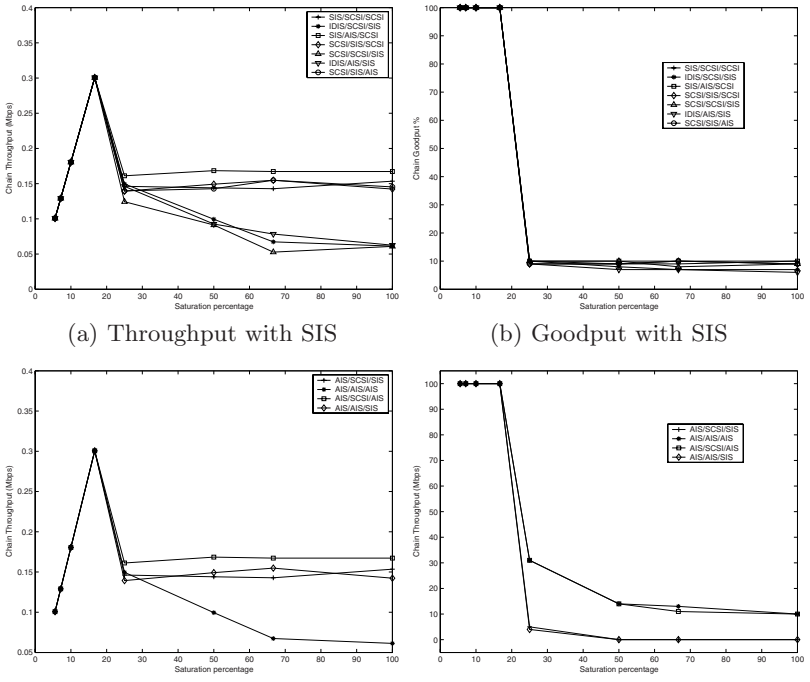
In this chain, all the hops are Sender Connected. As the source node competes for the channel with three other nodes (I1, I2 and I3), it is able to transmit at one fourth of the total bandwidth. The only drops in this interaction are due to two sources transmitting within very short duration of each other. Since the probability of these collisions is very little we see a goodput of more than 90%.

5.2 AIS/SCSI/SCSI

The chain effect dominates the throughput performance of this chain. In the AIS interaction between the first and the last hop, the first hop is the weak link and last hop is the strong link. Packets transmitted together on H1 and H4 cause the packet on Hop 1 to be dropped. Hence the lack of sender connectedness in the first group just increases the noise produced by wasted transmissions by Node S. This is depicted in the goodput curve of AIS/SCSI/SCSI in Figure 6(b). This chain transmits lots of packets on hop1 that are dropped.

5.3 NI/SCSI/AIS

In this chain, several packets are dropped at the AIS connection. The throughput is not affected severely because of the presence of SCSI in the middle that limits the transmission of packets from Source and $I2$ and hence limits the concurrently active packets in the chain. The goodput of this chain is affected by the lack of coordination between the hops.



(a) Throughput with SIS (b) Goodput with SIS

(c) Throughput with Reverse AIS (d) Goodput with Reverse AIS

Fig. 7. Throughput and Goodput of Chains with SIS and Reverse AIS interactions

5.4 AIS/AIS/SCSI

In this chain, the packets are sent in bursts because of the two AIS connections. The connection starts sending packets. When the packet reaches node I3, transmissions from I3 to node D cause packets to be dropped on Hop 1. For every packet sent successfully on Hop1, the next packet will be dropped, increasing the backoff at Node S. Hence the goodput for this chain is always less than 50%.

5.5 NI/AIS/AIS

This interaction is also dominated by the AIS group in terms of goodput. Since senders of all interactions are out of range, they will transmit together. This causes many wasted transmissions without any gain in throughput.

5.6 SIS Cases

In this section we consider those chains that have SIS interaction between any two hops. The probability of these interactions is really small using the default NS-2 parameters. Figure 7(a) shows the throughput of seven possible categories with SIS interactions. This type of interaction is really destructive as packets are dropped from both links and the throughput is drastically reduced. As can

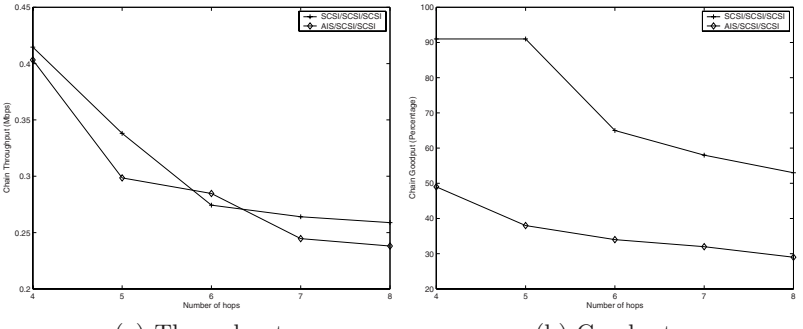
be seen in Figure 7(b), although the chain transmits many packet, few of these are successful.

5.7 Reverse AIS Cases

Another interaction that is possible in a chain (although the probability is low because of the geometry restrictions) is an AIS interaction between the first and the last hop where the first hop is the strong link and last hop is the weak link. The first hop transmits packets causing collisions at the last hop, which cannot empty packets as fast as it receives them (leading to queue drops). Figures 7(c) and 7(d) show the throughput and goodput of chains with Reverse AIS interaction.

6 Towards Generalization to n-Hop Chains

In this section we make some observations about generalizing the results from 4-hops to general chains via an inductive argument. In the future, we will attempt a more systemic generalization. First we add one more hop to our chain. The fifth hop can have two possibilities - it either interferes with the first hop or it does not. In the first case, the fifth hop interferes with the first, second, and third hops in either AIS or SCSI interactions because of symmetry. Our simulation results indicate that the throughput of all these scenarios is within 2.5% of each other.



(a) Throughput (b) Goodput

Fig. 8. Throughput and Goodput of 8-Hop Chains

In the second case where the fifth hop does not interact with the first hop, the situation is similar to evaluating a four hop chain where the second node of the chain is acting as the source of the 4-hop chain. As we have seen in section 5 the throughput of a chain does not depend on the type of interaction, hence the type of interaction between the 4 hops starting from the second node of the 5-hop chain would not effect the chain throughput. Hop count is the only dominating cause that effects the throughput of the chain. The goodput of the chain on the other hand is directly effected by the interactions amongst the hops. Chains with a higher number

of AIS interactions will have higher drops hence lower goodput, while SCSI dominated chains will produce a higher goodput. Figure 8 shows the throughput and goodput of 8-hop chains. In the SCSI/SCSI/SCSI chain, each group of 4-hops as obtained by shifting down the chain by one hop has SCSI/SCSI/SCSI interaction while in AIS/SCSI/SCSI has the same interaction for all sets of four-hops within the 8-hop chain.

7 Discussion

We have seen in this section that for all different interactions in a chain, the throughput of the chain with AIS and SCSI interactions depends only on the number of hops. Chains with SIS and Reverse AIS have very little throughput. The goodput of the chain is more influenced by the type of interaction. For chains with higher goodput, the channel utilization is more efficient which translates into less cross chain interference. Low goodput chains waste a lot of bandwidth for transmissions that in the end are dropped and hence wasted. Throughput of a chain should not be the only criteria for determining its performance. As we have seen from Figures 6(a) and 6(b) that chains that have similar throughput might have substantial difference in goodput. In routing decisions it is important to pick routes that minimize not only the interference within the chain but also across different chains in order to better utilize available channel bandwidth. Designing routing protocols that take consider chain interaction and pick high goodput routes is an area of our future research.

In chain interactions that occur more often, the type of interaction does not substantially affect the throughput of the chain although it does effect the amount of traffic generated. This observation leads us to believe that evaluating cross-chain interference and its effects on throughput are more important than self-interference in a chain.

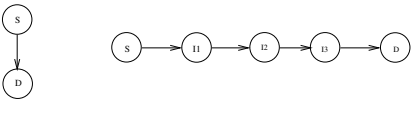
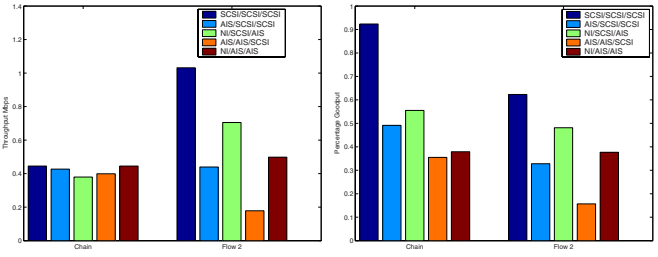


Fig. 9. A chain and an external flow

We consider the effect of noise from a chain on the throughput of other flows. Fig 9 shows a chain in close proximity to another flow. We determine the effect of this chain on the flow when the chain has different self interference patterns while the source of the chain has an AIS relationship with the second flow. In this AIS interaction, the second flow is the weaker link. Fig 10(a) and 10(b) show the effect of the chain on the throughput and goodput of an external flow. These are some preliminary results, a detailed study of cross-chain interference is a topic of our future research.



(a) Throughput (b) Goodput

Fig. 10. Effect of a chain on throughput and goodput of an external flow

8 Conclusion

This paper makes several contributions to the analysis of interference interactions multi-hop wireless chains. Specifically, we classify and study of all possible interactions within a chain and their effects on chain throughput and interference generated to other chains. We identify that some chains that produce high throughput in isolation, also experiences substantial drops hence wasting the available channel bandwidth with retransmissions and causing cross chain interference. The characterization of chains is important for routing protocols to be able to more intelligently select routes.

Our immediate goal is to extend this study to include a more realistic model where capture effects are taken into consideration. We would like to take the analysis performed in this paper to develop interference aware routing protocols that can look at a route and determine the types of interaction within the routes. Based on this study the protocol then decides on picking the best mix for throughput and goodput from all routes that are available.

References

[1] Garetto, M., Shi, J., Knightly, E.W.: Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In: MobiCom 2005 (2005)
[2] Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Trans. on Info. Theory (2000)
[3] Jain, K., Padhye, J., Padmanabhan, V.N., Qiu, L.: Impact of interference on multi-hop wireless network performance. In: MobiCom (2003)
[4] Kleinrock, L., Tobagi, F.: Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics. IEEE Transactions on Communications (1975)
[5] Kodialam, M., Nandagopal, T.: The Effect of Interference on the Capacity of Multi-hop Wireless Networks. Bell Labs Technical Report (2003)
[6] Lee, J., Lee, S.-J., Kim, W., Jo, D., Kwon, T., Choi, Y.: Rss-based carrier sensing and interference estimation in 802.11 wireless networks. In: SECON 2007, pp. 491–500 (2007)

[7] Li, J., Blake, C., De Couto, D.S.J., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: MobiCom 2001: Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 61–69. ACM Press, New York (2001)
[8] Ng, P.C., Liew, S.C.: Throughput analysis of ieee802.11 multi-hop ad hoc networks. IEEE/ACM Trans. Netw. 15(2), 309–322 (2007)
[9] Razak, S., Kolar, V., Abu-Ghazaleh, N.: Modeling and Analysis of Two-Flow Interactions in Wireless Networks. In: The Fifth Annual Conference on Wireless On demand Network Systems and Services (2008)
[10] Xu, S., Saadawi, T.: Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks. Comput. Networks 38(4), 531–548 (2002)
[11] Xu, S., Saadawi, T., Lee, M.: On tcp over wireless multi-hop networks. In: Military Communications Conference. Communications for Network-Centric Operations: Creating the Information Force, vol. 1, pp. 282–288. IEEE, Los Alamitos (2001)

Energy-Efficient Multi-path Routing in Wireless Sensor Networks

Philipp Hurni and Torsten Braun

Universität Bern, Neubrückstrasse 10, CH-3012 Bern
{hurni,braun}@iam.unibe.ch

Abstract. The paper investigates the usefulness of multi-path routing to achieve lifetime improvements by load balancing and exploiting cross-layer information in wireless sensor networks. Performance gains in the order of 10-15 % could be achieved by altering path update rules of existing on-demand routing schemes. Problems encountered with concurrent traffic along interfering paths have been identified as a direct consequence of special MAC protocol properties.

Keywords: Sensor Networks, Energy Efficiency, Routing Protocols.

1 Introduction

1.1 Benefits of Multi-path Routing

Standard routing protocols in ad hoc wireless networks, such as AODV [3] and DSR [4] are mainly intended to discover one single route from a source to a destination. During the route discovery process, these protocols aim to find the best route with the lowest cost. Multi-path routing protocols aim to find multiple routes. Multiple routes can be useful to compensate for the dynamic and unpredictable nature of ad hoc networks, also in energy and bandwidth constrained sensor networks. Multi-path routing has been investigated in the Internet, in metropolitan and local area networks, in wireless mobile ad hoc networks, as well as in wireless sensor networks. In [6] goals, problems and recent suggestions for multi-path routing protocols in wireless ad hoc networks have been discussed. Discovering and maintaining multiple paths causes certain overhead, but yields several advantages, namely load balancing, fault tolerance, bandwidth aggregation, and reduced delay [2].

***Load Balancing*:** Multi-path routing can avoid congestion and improve performance. When certain nodes and links become over-utilized and cause congestion, multi-path routing can spread traffic over alternate paths to balance the load over those paths. In wireless sensor networks, the main focus of multi-path routing is typically on the load balancing issue. As nodes are constraint to a limited amount of energy, and traffic is expected to be low, the main concern is to keep the network operable for a maximum amount of time. In sensor networks, one has to deal with traffic generated by many leaf nodes attempting to deliver data to one or a few sinks. Usual on-demand routing schemes tend to utilize always the same set of nodes to forward packets, whereas

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 72–85, 2008.
© Springer-Verlag Berlin Heidelberg 2008

many other nodes remain unused. It has been observed that in such cases nodes that have to forward traffic from large sub-trees suffer much earlier from energy depletion, whereas other nodes have only slightly been used. When nodes collaborate in sensing and data forwarding and packets are not always routed on the same routes, but the load is balanced over multiple routes, network lifetime can be increased significantly.

***Fault Tolerance*:** Multi-path routing protocols can increase the degree of fault tolerance by having redundant information routed to the destination over alternate paths. This increases the energy overhead, but helps to reduce the probability that communication is disrupted and data is lost in case of link failures. Sophisticated algorithms have been developed to increase the degree of reliability. The trade-off between the additional overhead and the reliability gain has been investigated in [5].

***Bandwidth Aggregation*:** By splitting data to the same destination into multiple streams, each stream is routed through a different path. The effective bandwidth can be aggregated. This strategy is especially beneficial when a node has multiple low bandwidth links but requires higher bandwidth than each individual link can provide.

***Reduced Delay*:** In wireless networks running single path on-demand routing protocols, route failures trigger the path discovery process to find new routes causing route discovery delay. Delay can be reduced in multi-path routing, as backup routes can be identified immediately. Furthermore, discovering several paths and observing Quality-of-Service (QoS) characteristics of both paths permits to switch the load to another route whenever the service parameters of another route promise better quality.

In wireless sensor networks, the focus of multi-path routing is often on load-balancing or fault tolerance, rather than on the aggregation of bandwidth. Often, the goal of multi-path routing protocols is to maximize the time the network is operable and fulfills its observation task.

1.2 Route Coupling

Using multiple paths in ad hoc networks to achieve higher bandwidth, balance load or achieve fault tolerance is not as easy as in wired networks. As nodes in the network communicate through the wireless medium, radio interference must be taken into account. Transmissions along one path may interfere with transmissions along another path, even if the paths are link-disjoint or even node-disjoint. The interference may limit the achievable throughput and lead to two paths with impact on each other for forwarding packets. This phenomenon is often referred to as ***route coupling***. Route coupling occurs when two routes are located physically close enough to interfere with each other during transmission. As a result, the nodes along those two routes are constantly competing for medium access. The advantages of two routes being available are therefore limited.

Route coupling in wireless networks caused by radio interference between paths can have serious impact on the performance of multi-path routing protocols, even if the paths are disjoint [7]. In some cases, route coupling can even lead to worse results than routing over one single path. The shared transmission medium forces all nodes in the interference range of a sender to remain silent until completion of a transmission. The problem even gets worse when applying an RTS/CTS scheme. In [9] the

influence of route coupling in wireless networks applying multi-path routing has been studied. The following types of routes can be distinguished:

a) routes with no common collision domain
b) routes with a common link
c) routes sharing a common node

Paths of type a) produce the best throughput results, because the common collision domain of the multiple paths is reduced to source and destination nodes, and transmission along the path are independent to the largest possible extent. Although more efficient network utilization due to better load balancing can justify the use of a multi-path routing strategy compared to single path routing, the benefits of multi-path routing in terms of throughput quickly vanish in case of interference [9].

In [10] it is argued that many multi-path routing protocols mainly find routes that are too close to each other to actually behave much different than single path routing schemes. To save energy, multi-path routes must ensure that traffic is routed along routes that do not interfere with each other at all, which is in most cases hard to achieve.

None of the established and well-investigated proposals have considered and incorporated the route-coupling phenomenon for effective load balancing. Recent research has been pursued on the issue of on-demand construction on non-interfering multiple paths in sensor networks [8]. The proposed mechanism routes packets along paths that have a gap of two transmission ranges in between. The mechanism strongly relies on the position-awareness of the sensor nodes and the knowledge of the position of the receiver.

1.3 Overview

This paper investigates the usefulness of multi-path routing in wireless sensor networks. After discussing related work in Section 2, we propose in Section 3 a multi-path routing protocol for wireless sensor networks based on the AODV multi-path extensions called AOMDV. The protocol has been evaluated by simulations as discussed in Section 4. Section 5 concludes the paper.

2 Related Work

2.1 Multi-path Routing Protocols

Several multi-path protocols for wireless ad-hoc networks such as the Ad-hoc On Demand Distance Vector Multi-path routing protocol (AODVM) [14] and Split Multi-path Routing (SMR) [12] have been proposed. The protocol described in this paper has mainly been influenced by the Ad hoc On-demand Multi-path Distance Vector protocol (AOMDV) [13], which is an extension of AODV for discovering node-disjoint or optionally link-disjoint paths. It finds node-disjoint paths by exploiting a particular property of flooding. By appending the first-hop to the RREQ (Route Request) header, and bookkeeping about the first-hops of the recently received RREQs, nodes receiving duplicate RREQs by different neighboring nodes can easily determine whether the routes are node-disjoint. The first-hop is the first node a RREQ

traverses after the initiating source. To find node-disjoint routes, nodes do not immediately reject RREQs. Each RREQ arriving via a different neighbor of the source has a different first-hop in the RREQ header, and therefore defines a node-disjoint path. Nodes do never rebroadcast duplicate RREQs, so any two RREQs arriving at an intermediate node via a different neighbor of the source could not have traversed the same node. As in AODV, RREQ duplicates are discarded in intermediate nodes. RREQs with equal destination sequence number, but incoming from another intermediate node are simply ignored in AODV, unless they advertise a better hop count value. In AOMDV, intermediate and destination nodes reply to such RREQs with RREP (Route Reply) messages, if their first-hop is different from the one in the prior received RREQ. Using this policy, AOMDV guarantees node-disjoint paths whenever it takes up a second routing entry to the same destination. AOMDV further allows discovering link-disjoint paths by exploiting RREQ duplicates arriving at the destination via different intermediate nodes. AOMDV [13] leaves the choice to use the option to the user.

Figure 1 and Figure 2 illustrate the AOMDV mechanisms to find node-disjoint paths. The illustration shows node 1 initiating a route request to node 8. The RREQ is flooded via node 2 and node 3. There, the first-hop field is set accordingly. The RREQs finally reach destination node 8, where both incoming requests create new path entries for source node 1, because the incoming RREQs exhibit a different first-hop. Furthermore, to establish the full bidirectional routes, both RREQs are replied. Node 6 similarly receives two RREQs via nodes 4 and 5. Both RREQs, however, exhibit the same first-hop. Node 6 therefore knows that the paths to the source node advertised by these RREQs are not node-disjoint, and does not add a second path entry. To support multi-path routing, the AOMDV route tables contain a list of intermediate nodes and hop counts for each destination node. The path entries (cf. Table 1 Table 3) to a destination have all the same destination sequence number, as they have been obtained in one single RREQ-RREP query cycle. When receiving a path advertisement with a higher sequence number, all routes with the old sequence number are removed.

[11] considers how to construct secondary paths, which are in the optimal case node disjoint. The study is focused on the question how to keep the overhead as small as possible if only one node or one link in the network fails. The authors argue that when a small number of paths are kept alive, failures on the primary path can usually be recovered without invoking network-wide flooding for path discovery. This feature is important in sensor networks since flooding is very costly and can vastly reduce network lifetime. Node-disjoint paths are a very strong condition when aiming to find multiple paths between two nodes and may result in rather inefficient and suboptimal paths in terms of hop count. Long detours around many nodes can be necessary to fulfill the condition of node-disjoint paths. Alternate node-disjoint paths can become very long, and therefore require significantly more energy than the primary path. To overcome this problem, and yet retain the robustness advantages of multiple paths, the authors suggest the construction of so-called braided paths. Braided paths relax the requirement for node-disjoint paths. Such paths are only required to leave out some of the primary path's nodes. They are free to use other nodes on the primary path. In [11] it is proposed to construct two different kinds of redundant paths - node-disjoint paths and braided paths. It depends on the failure patterns which of the two schemes shall be used. It is claimed to achieve better path resilience with the braided path approach.

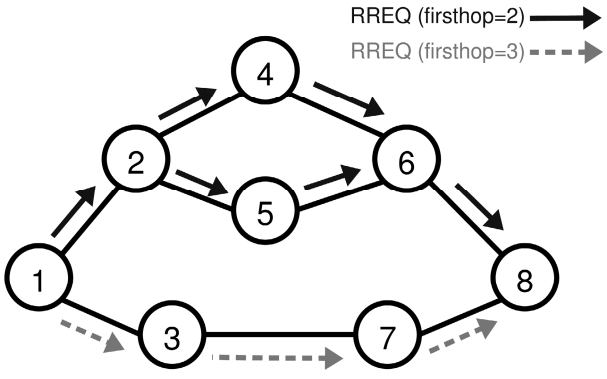


Fig. 1. AOMDV Route Request

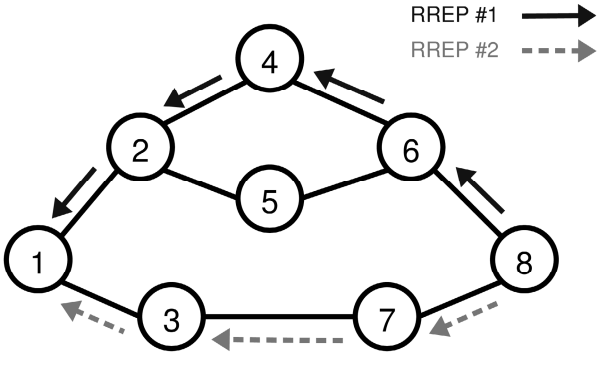


Fig. 2. AOMDV Route Reply

Table 1. Routing table node #1

dest	next	hops	seq
8	3	3	37
8	2	4	37

Table 2. Routing table node #6

dest	next	hops	seq
1	4	3	11
8	8	1	37

When discovering and maintaining multiple paths from a source to a destination, it may make sense to occasionally use suboptimal paths in terms of hop count that use more energy for an end-to-end transmission than the optimal one. Traffic load can be

Table 3. Routing table node #8

dest	next	hops	seq
1	7	3	11
1	6	4	11

spread over multiple paths, which leads to more nodes participating in the forwarding process. Using the lowest energy path for all packets is not necessarily best for the long-term health of a sensor network, as important forwarders might run out of energy first. In [15] a quite simple approach to probabilistically incorporate suboptimal routes is suggested. Each node maintains an energy cost estimate for each of its path entries. This cost estimate determines the probability that a packet is routed over a certain path. If a node aims to transmit a packet to a certain destination for which it has multiple paths, it chooses the forwarding node according to a probability assigned to that path. Each intermediate node does the same and forwards packets according to the probability assigned to the different paths in the table. This is continued until the data packet reaches the destination node. Using this simple mechanism to send traffic over different routes helps in using the nodes' resources more equally. An overall gain of ~40% of network lifetime increase with this probabilistic routing scheme has been achieved. Taking suboptimal paths occasionally into account pays off as nodes use their scarce resources more equally, which helps to remove load from central forwarder nodes that would otherwise run out of energy first.

2.2 Sensor MAC Protocols

Routing performance in wireless sensor networks heavily depends on the underlying MAC protocol. Cross-layer designs are required to optimize performance in terms of throughput, energy efficiency, delay etc. WiseMAC [1] appears to be one of the most efficient MAC protocols for wireless sensor networks. It is based on preambles submitted prior to data. If the receiver's wake-up pattern is still unknown, the preambles are slightly longer than the time between two wakeups of a sensor node, such that a sensor node waking up will discover an upcoming transmission from another node and remain active until the frame reception (Figure 3). After successful frame reception, the receiver node piggybacks its own schedule to the respective frame acknowledgement. Received schedule offsets of all neighbor nodes are subsequently kept in a table and are periodically updated. Based on this table, a node can determine the wake-up intervals of all its neighbors and minimize the preamble length for upcoming transmissions.

In previous work we have derived a similar scheme that offers a better protection against systematic overhearing and does not rely on full-cycle preamble for the neighborhood discovery [19]. We propose to implement so-called moving wake periods. Figure 4 shows the approach where a wake period is moving forward and backward within a fixed interval equal to the average time interval between two wakeups of a node in WiseMAC. Nodes just need to select the same fixed interval value, but do not need to synchronize further. The moving wake periods scheme ensures that two nodes can detect each other by periodic transmission of HELLO messages after a limited time, because their wake periods will sooner or later overlap. If two nodes

have detected each other and learned about their schedule they calculate when the other node becomes active again in order to schedule pending transmissions. This scheme proved to avoid overhearing and fairness problems of WiseMAC's fixed static wake-up pattern, in particular when two neighbour nodes share a similar wake pattern. Moving intervals proved to help reducing end-to-end latency over minimum-hop paths by intelligently choosing gateway nodes to forward packets.

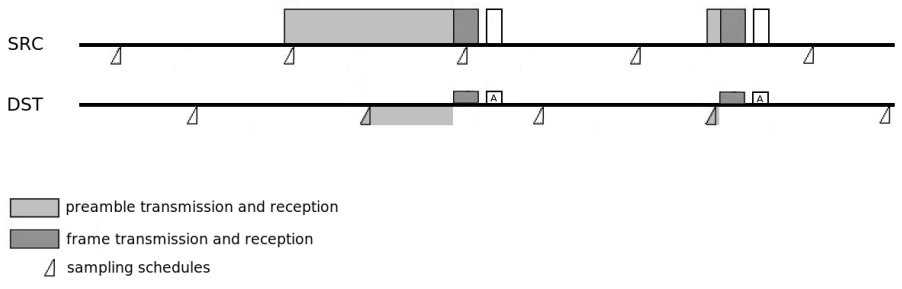


Fig. 3. WiseMAC

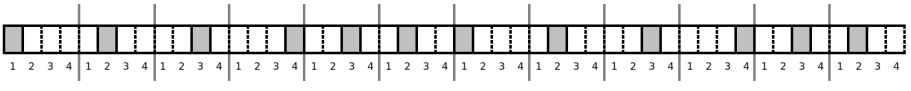


Fig. 4. MAC with moving wake periods

3 Energy-Efficient Multi-path Routing for Wireless Sensor Networks

3.1 AOMDV Inspired Multi-path Routing

Our energy-efficient multi-path routing approach is based on AOMDV, because the path construction algorithm of AODVM depends on overhearing neighboring nodes' transmissions. Permanent overhearing requires to keep the receiver constantly in the receive state, which is contrary to the scope of the energy-efficient MAC. Moreover, we found in initial experiments that redundant paths detected by multi-path routing schemes were often much longer than the optimal paths. Long detours of redundant paths have negative impact on the lifetime, because more transmissions become necessary when paths are suboptimal, and each transmission may influence other nodes in the carrier sensing range.

AODV is tailored to the use in mobile ad hoc networks and always keeps the freshest route to every destination. A node receiving a path advertisement for a given destination node checks whether the advertisement provides a higher destination sequence number, or if it provides an equal destination sequence number and a shorter path to the destination. If it does, the current entry for this destination is deleted and the packet source is taken as new next node towards the destination node. As AODV has been designed for use in mobile ad-hoc networks, in which nodes move in and out

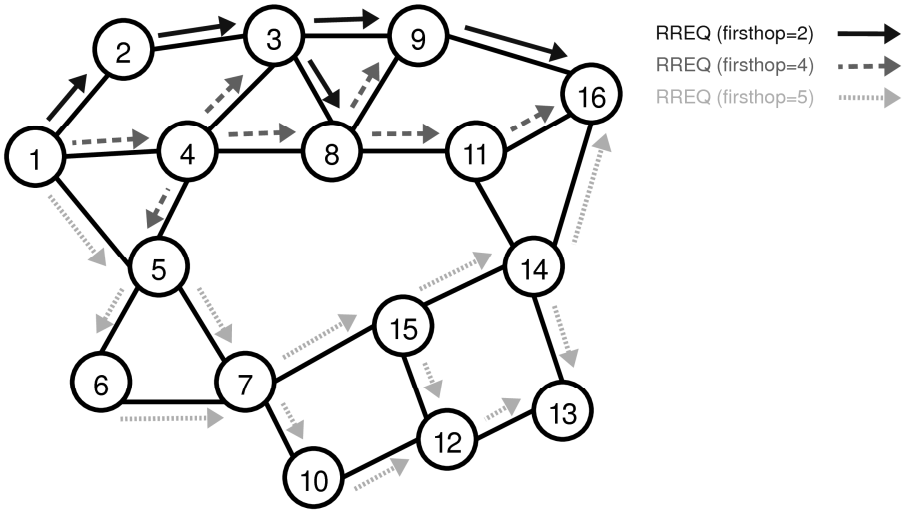
of the transmission range of each other, the sequence number condition ensures that a node always uses the path known to be the freshest one. However, most wireless sensor networks can be assumed to be rather static and node mobility does not play a major role. We therefore weakened the condition of prioritizing route advertisements with the highest sequence number. Our approach considers route advertisements to a destination with higher sequence number only, if the route is not longer than the current one. The approach incorporates the basic mechanism of the AOMDV protocol to find node-disjoint paths, but adds such paths only, if they advertise the same hop count. Incoming RREQ duplicates are treated as in AOMDV: they are answered, if they advertise a node-disjoint path to a destination and if they advertise the same hop count. To summarize, we add an additional path entry to the same destination to which a path is already known if it meets all of the following criteria:

a) The sequence number is equal or higher,
b) The first-hop is different from all already known paths to the same destination
c) The hop count is equal.

When a path advertisement arrives with lower hop count, all existing routes are deleted and the new route is added. When receiving a duplicate that fulfils the condition of a node-disjoint path and is optimal in terms of hop count, the routing table is extended to contain more than one path entry. The modification of the routing table entry update rule compared to AOMDV and AODV can be explained by Figure 5, where the dissemination of a RREQ from node 1 searching a path to node 16 is depicted. After flooding the whole network, the destination node receives path advertisements to node 1 from its neighbours 9, 11 and 14. With AODV, the destination node only answers to the first incoming RREQ with a corresponding RREP, e.g., from neighbour 9. The duplicate RREQ from neighbour 11 is simply discarded and left unanswered, as it advertises the same sequence number. Although it took another route and would provide path redundancy, AODV discards the request and leaves it unanswered. In contrast, AOMDV considers all routes that are advertised by neighbours 9, 11 and 14, as the respective RREQs all took another first hop. We changed the table update policy such that only the optimal routes in terms of hop count are added to the table and answered with a RREP. In the previous case, the RREQs received via neighbours 9 and 11 are answered with a RREP, but not the one received via neighbour 14. The resulting routing tables for source node 1 and destination node 16 are depicted in Figure 5. With AODV, only one path entry is considered, whereas AOMDV adds all paths to its table. With our approach, only the hop-count optimal routes via nodes 9 and 11 are added to the table.

AOMDV only addresses the question how to establish multiple routes, but not how to spread the load over them. There are probabilistic schemes that assign a certain probability to a route and choose the route for each packet in a random manner. We suggest exploiting information provided by the MAC layer to achieve some performance gains in respect to the latency. As all redundant path entries to a destination advertise an optimal route in terms of hop count, the next soonest wake-up of the gateway leading to the destination shall be the only selection criterion, also in each intermediate node. For a transmission of a packet from source to destination, each

intermediate node shall forward the packet to the node with the soonest wake-up. A lower latency as well as the desired load balancing among the intermediate nodes can thus be expected.



AODV

routing table node #1			
dest	next	hops	seq
16	2	4	11

AOMDV

routing table node #1			
dest	next	hops	seq
16	2	4	11
16	4	4	11
16	5	6	11

only optimal routes

routing table node #1			
dest	next	hops	seq
16	2	4	11
16	4	4	11

routing table node #16			
dest	next	hops	seq
1	9	4	23

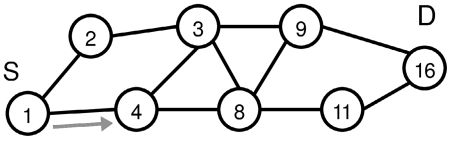
routing table node #16			
dest	next	hops	seq
1	9	4	23
1	11	4	23
1	14	6	23

routing table node #16			
dest	next	hops	seq
1	9	4	23
1	11	4	23

Fig. 5. RREQ and different table update policies

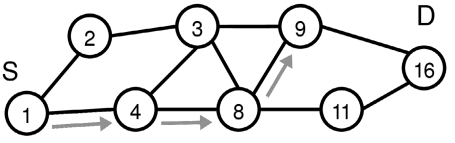
As the source knows two paths towards node 16, it chooses the path according to the delay to the next-wake-up of the gateway node. In the left part of Figure 6, we can see that the time remaining to the next wake-up of node 2 is 132 ms, and the next wake-up of node 4 is in 54 ms. Therefore, the source node chooses to send the packet via node 4, because it can deliver the packet and empty its buffer earlier. The packet is routed in every intermediate node accordingly. Since we only added hop-count-optimal routes, packets are never routed away from the destination.

If we would apply WiseMAC with its simple periodic wake-up pattern, nodes would always forward over the same gateways, because the time shift between two



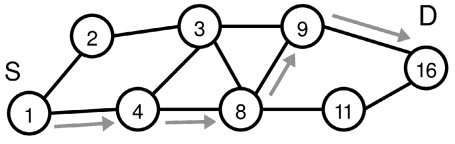
dest	next	hops	Δt [ms]
16	2	4	132
16	4	4	54

routing table node #1



dest	next	hops	Δt [ms]
16	3	3	213
16	8	3	72

routing table node #4



dest	next	hops	Δt [ms]
16	9	2	13
16	11	2	47

routing table node #8

Fig. 6. Packet forwarded from 1 to 16 choosing nodes with the soonest wake-up

node's wake-ups remains constant. With the moving wake-intervals MAC (Figure 4) nodes will always choose their gateway according to the shortest delay to the next wake-up. The choice of the gateways may change, because the offset and minimum delays dynamically change with the moving intervals.

4 Evaluation

4.1 Simulation Parameters and Scenarios

We performed our evaluation using the OMNeT++ network simulator [16] and the mobility framework [17]. The energy consumption model is based on the amount of energy that is used by the transceiver unit. We do not take processing costs of the CPU into account. Each node's energy consumption is calculated in respect to the time and input current that the node spends in the respective operation modes idle/receive, transmit and sleep. Furthermore, state transition delays are taken into account. The simulation parameters are summarized in Table 4. As the choice of the network topology may have an impact on the results, we considered the following three network topologies:

- uniformly distributed network topology of 90 nodes in an area of 300x300 m
- 7x7 nodes lattice square topology (Figure 7)
- 3x10 nodes grid topology (Figure 8)

We defined the lifetime of the network as the time until 10% of the nodes deplete or the network becomes partitioned. For each topology, we measured two different traffic patterns.

- Evenly distributed traffic: Every node starts reporting data according to the Poisson model with $\lambda = 0.01$. When every node generates the same amount of traffic, multi-path routing might not pay off, because the load is already balanced. As common single path routing protocols establish source-sink trees with some nodes having the burden to forward traffic of large sub-trees, multi-path routing still might help to redistribute the load over more hops.
- Neuralgic spots traffic: If there are neuralgic spots in the network that generate much traffic, whereas other parts stay more or less inactive, multi-path routing can pay off more. We assume that the three most distant nodes from the sink generate 20 times more traffic ($\lambda = 0.05$) than all other nodes ($\lambda = 0.0025$).

Table 4. Simulation Parameters

carrier frequency	868 MHz
bit rate	19.2 kbps
packet size including header	160 bits
transmitter power	0.1 mW
SNR threshold	4 dB
sensitivity	-101.2 dBm
sensitivity carrier sensing	-112 dBm
communication range	50 m
packet loss coefficient α	3.5
carrier sensing range	100 m
node energy	20 J
supply voltage	3V
current	
transmit	12 mA
receive	4.5 mA
sleep	5 µA
state transition delays	
receive to transmit	12 µs
transmit to receive	12 µs
sleep to receive	518 µs
receive to sleep	10 µs
transmit to sleep	10 µs

4.2 Lifetime and Delay Results

The results in Figure 9 and Figure 10 show an overall performance gain when applying the AOMDV-related scheme coupled with the next-soonest-wake-up routing paradigm of ~10-15% concerning network lifetime and one-way delay. When consid-

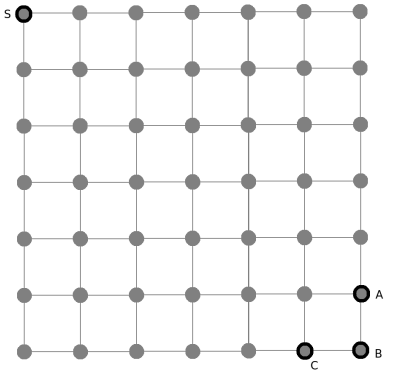


Fig. 7. 7x7 nodes lattice square topology

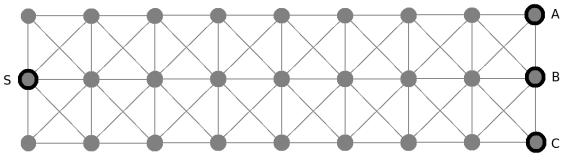


Fig. 8. 3x10 nodes grid topology

ering the low cost of some additional RREP messages in the initial route discovery phase, the results show that on-demand multi-path routing may provide limited but valuable contributions to extend network operability. In our simulation, the mechanism paid off when sticking to the hop count optimal routes only. The performance improvements are in a similar range as in [18] although the authors focused on wireless ad-hoc networks and on throughput optimization.

The exploitation of the MAC layer information about the next-soonest wake-up of the neighboring nodes paid off in respect to the one-way delay. This might be in conflict with the layered design paradigm, but in wireless sensor networks with scarce energy resources, such cross-layer approaches are acceptable, if higher efficiency can be achieved. Neither the different traffic patterns nor the topology has a big impact on the results.

In a second series of experiments, we performed all the experiments with another lifetime metric. In that case, we measured the first node depletion. The results differed only slightly from the results using the first lifetime metric. The overall gain also reached 10-15% in respect to lifetime and to one-way delay for all topologies and both traffic patterns.

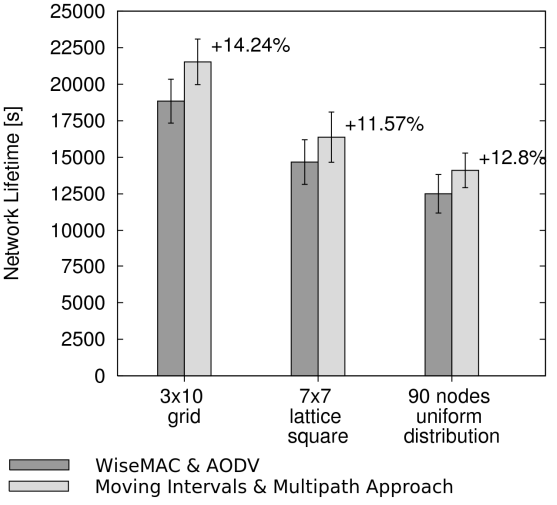


Fig. 9. Network Lifetime

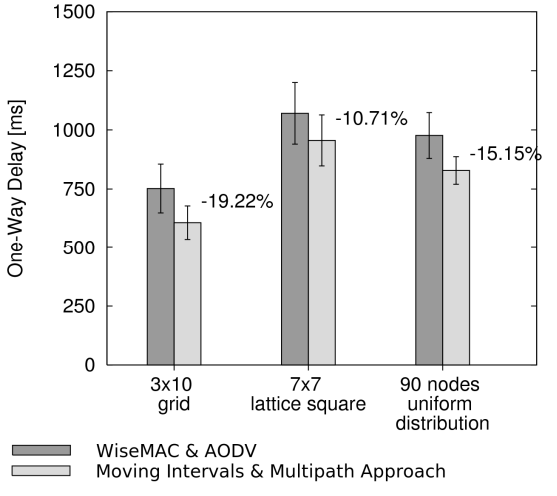


Fig. 10. One-Way Delay

5 Conclusions

The paper proposed to integrate a multi-path routing protocol and appropriate MAC protocols with periodic wake-up to balance load in a wireless sensor network and achieve higher network lifetime. The proposed concept achieves this by exploiting cross-layer optimizations between MAC and routing protocol as well as by altering path update policies. The evaluation showed that the potential to achieve higher network lifetimes is limited when applying preamble sampling low-power MAC protocols such as WiseMAC. WiseMAC amplifies the performance degrading route

coupling effect, because it increases its carrier-sensing range with a more prohibitive carrier access policy. This increased the probability that transmissions along multiple paths interfere with each other. Load balancing of multipath-routing was deteriorated by the additional cost of coping with path interference. The mechanism exhibited performance gains of 10-15% in respect to throughput and average end-to-end delay.

References

[1] El-Hoiydi, A., Decotignie, J.-D.: WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks. ALGOSENSORS (2004)

[2] Tsai, J., Moors, T.: A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks. ACoRN Early Career Researcher Workshop on Wireless Multihop Networking (2006)

[3] Perkins, C.E., Belding-Royer, M., Ad, E.: hoc On-Demand Distance Vector (AODV) Routing, RFC 3561 (2003)

[4] Johnson, D.B.: Routing in Ad Hoc Networks of Mobile Hosts. In: Workshop on Mobile Computing Systems and Applications. IEEE Computer Society, Los Alamitos (1994)

[5] Dulman, S., Nieberg, T., Wu, J., Havinga, P.: Trade-off between Traffic Overhead and Reliability in Multi-path Routing for Wireless Sensor Networks. In: WCNC Workshop (2003)

[6] Mueller, S., Ghosal, D.: Multi-path Routing in Mobile Ad Hoc Networks: Issues and Challenges. MASCOTS Tutorials (2003)

[7] Pearlman, M.R., Haas, Z.J., Sholander, P., Tabrizi, S.S.: On the impact of alternate path routing for load balancing in mobile ad hoc networks. In: ACM international symposium on Mobile ad hoc networking & computing, Boston (2000)

[8] Voigt Th., Dunkels A., Braun, T.: On-demand Construction of Non-interfering Multiple Paths in Wireless Sensor Networks 2nd Workshop on Sensor Networks, Informatik (2005)

[9] Waharte, S., Boutaba, R.: Totally Disjoint Multi-path Routing in Multihop Wireless Networks. IEEE International Conference on Communications (2006)

[10] Ganjali, Y., Keshavarzian, A.: Load Balancing in Ad hoc Networks: Single-path Routing vs. Multi-path Routing. IEEE Infocom (2004)

[11] Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly Resilient, Energy-Efficient Multi-path Routing in Wireless Sensor Networks. Mobile Computing and Communications Review (2001)

[12] Lee, S.-J., Gerla, M.: Split Multi-path Routing with Maximally Disjoint Paths in Ad Hoc Networks. In: IEEE International Conference on Communications (2001)

[13] Marina, M.K., Das, S.R.: On Demand Multi-path Distance Vector Routing in Ad hoc Networks. In: IEEE International Conference on Network Protocols (2001)

[14] Ye, Z., Krishnamurthy, S.V., Tripathi, S.K.: A Framework for Reliable Routing in Mobile Ad Hoc Networks. IEEE Infocom (2003)

[15] Shah, R.C., Rabaey, J.: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. In: IEEE Wireless Communications and Networking Conference WCNC (2002)

[16] Varga, A.: The OMNeT++ Discrete Event Simulation System European Simulation Multiconference (ESM 2001), Prague, Czech Republic (June 2001), `http://www.omnetpp.org`

[17] Mobility Framework for OMNeT++, `http://mobility-fw.sourceforge.net`

[18] Bononi, L., Di Felice, M.: Performance Analysis of Cross-Layered Multipath Routing and MAC Layer Solutions for Multi-hop Ad Hoc Networks. ACM MobiWAC (2006)

[19] Hurni, P., Braun, T.: Improving Unsynchronized MAC Mechanisms in Wireless Sensor Networks. In: 1st ERCIM Workshop on eMobility, Coimbra, Portugal, May 21 (2007)

Approximating Minimum-Power k-Connectivity

Zeev Nutov

The Open University of Israel, Raanana, Israel
nutov@openu.ac.il

Abstract. The Minimum-Power k-Connected Subgraph (MPkCS) problem seeks a power (range) assignment to the nodes of a given wireless network such that the resulting communication (sub)network is k-connected and the total power is minimum. We give a new very simple approximation algorithm for this problem that significantly improves the previously best known approximation ratios. Specifically, the approximation ratios of our algorithm are:

- 3 (improving $(3+2/3)$) for $k=2$,
- 4 (improving $(5+2/3)$) for $k=3$,
- $k+3$ for $k \in \{4,5\}$ and $k+5$ for $k \in \{6,7\}$ (improving $k+2\lceil (k+1)/2 \rceil$),
- $3(k-1)$ (improving $3k$) for any constant k.

Our results are based on a $(k+1)$-approximation algorithm (improving the ratio $k+4$) for the problem of finding a Min-Power k-Inconnected Subgraph, which is of independent interest.

1 Introduction

1.1 Preliminaries

Wireless networks are studied extensively due to their wide applications. The power consumption of a station determines its transmission range, and thus also the stations it can send messages to; the power typically increases at least quadratically in the transmission range. Assigning power levels to the stations (nodes) determines the resulting communication network. Conversely, given a communication network, the cost required at v only depends on the furthest node that is reached directly by v. This is in contrast with wired networks, in which every pair of stations that need to communicate directly incurs a cost.

In network design problems one seeks to design a "cheap" communication (sub)network that satisfies some prescribed properties. An important network property is fault-tolerance, often measured by node-connectivity of the network. Node-connectivity is much more central here than edge-connectivity, as it models stations failures. Such problems were vastly studied; see [1,3,4,9,11,12,17,21] for only a small sample of papers in this area. We consider the Min-Power k-Connected Subgraph (MPkCS) problem which is the power variant of the classic Min-Cost k-Connected Subgraph (MCkCS) problem. We give an approximation algorithm for MPkCS that significantly improves the previously best known ones.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 86–93, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Definition 1. *Let $H = (V, I)$ be a graph with edge-costs $\{c(e) : e \in I\}$. For $v \in V$, the* power $p(v) = p_H(v)$ of v *in H (w.r.t. c) is the maximum cost of an edge in I leaving v, i.e., $p(v) = p_I(v) = \max_{vu \in I} c(vu)$. The power of the graph is the sum of the powers of its nodes.*

Note that $p(H)$ differs from the cost $c(H) = \sum_{e \in I} c(e)$ of H even for unit costs; for unit costs, if H is undirected, then $c(H) = |I|$ and (if H has no isolated nodes) $p(H) = |V|$. For example, if I is a perfect matching on V then $p(H) = 2c(H)$. If H is a clique then $p(H)$ is roughly $c(H)/\sqrt{|I|/2}$. For directed graphs, the ratio of the cost over the power can be equal to the maximum outdegree, e.g., for stars with unit costs. The following statement, parts of which appeared in various papers, c.f., [9,11], shows that these are the extremal cases for general edge costs.

Proposition 1. *$c(H)/\sqrt{|I|/2} \leq p(H) \leq 2c(H)$ for any undirected graph $H = (V, I)$, and if H is a forest then $c(H) \leq p(H) \leq 2c(H)$. For any directed graph D holds: $c(D)/\Delta(D) \leq p(D) \leq c(D)$, where $\Delta(D)$ is the maximum outdegree of a node in D.*

Minimum-power problems are usually harder than their minimum-cost versions. The Minimum-Power Spanning Tree problem is APX-hard. The problem of finding minimum-cost k pairwise edge-disjoint paths is in P (this is the Minimum-Cost k-Flow problem, c.f., [22]) while both directed and undirected minimum-power variants are unlikely to have even a polylogarithmic approximation [9,17]. Another example is finding an arborescence rooted at s, that is, a subgraph that contains an sv-path for every node v. The minimum-cost case is in P (c.f., [22]), while the minimum-power variant is at least as hard as the Set-Cover problem. For more examples see [1,21].

A *network* is a (possibly directed) graph with edge costs. For a graph $H = (V, I)$ and $X \subseteq V$, let $d_I(X) = d_H(X)$ denote the degree of X in H, that is the number of edges from X to $V - X$. All the graphs in the paper are assumed to be simple, and, unless stated otherwise, undirected.

A graph $H = (V, I)$ is *k-connected* if it contains k internally-disjoint uv-paths for all $u, v \in V$. We consider the min-power variant of the extensively studied classic Min-Cost k-Connected Subgraph (MCkCS) problem.

Minimum-Power k-Connected Subgraph (MPkCS)
Instance: A graph $G = (V, E)$ with edge costs $\{c(e) : e \in E)$, and an integer k.
Objective: Find a minimum-power k-connected spanning subgraph H of G.

1.2 Previous and Related Work

We now introduce some additional related problems, that will also play an important role later. The first problem is the min-power variant of the Min-Cost k-Flow problem (with unit node capacities).

Min-Power k Disjoint Paths (MPkDP)
Instance: A graph $G = (V, E)$, edge-costs $\{c(e) : e \in E\}$, $u, v \in V$, an integer k.
Objective: Find a min-power subgraph H of G with k internally-disjoint uv-paths.

A (possibly directed) graph $H = (V, I)$ is *k-inconnected to s* if it contains k internally-disjoint vs-paths for all $v \in V - s$.

Min-Power k-Inconnected Subgraph (MPkIS)
Instance: A graph $G = (V, E)$, edge-costs $\{c(e) : e \in E\}$, $s \in V$, an integer k.
Objective: Find a min-power k-inconnected to s spanning subgraph H of G.

Min-Power k-Edge-Cover (MPkEC)
Instance: A graph $G = (V, E)$, edge-costs $\{c(e) : e \in E\}$, an integer k.
Objective: Find a min-power edge set $I \subseteq E$ so that $d_I(v) \geq k$ for all $v \in V$.

It is easy to see (c.f., [11,9]) that the simplest heuristic for MPkEC that for every node $v \in V$ takes the k cheapest edges incident to v is a $(k+1)$-approximation algorithm for MPkEC. In [12] the approximation ratio $O(\log n)$ was derived. For $k = 1$ a 3/2-approximation algorithm is given in [13].

It turns out that approximating MPkCS is closely related to approximating MCkCS and MPkEC as shows the following observation from [9], which first part was implicitly observed independently in [11].

Theorem 1 ([9,11])

(i) *An α-approximation for* MCkCS *and a β-approximation for* MP$(k-1)$EC *implies a $(2\alpha+\beta)$-approximation for* MPkCS.
(ii) *A ρ-approximation for* MPkCS *implies a $(2\rho+1)$-approximation for* MCkCS.

One can combine various values of α, β with Theorem 1(i) to get approximation algorithms for MPkCS. As was mentioned, currently $\beta = \min\{k, O(\log n)\}$ [9], and $\beta = 3/2$ for $k = 2$ [13] (note that here β is the ratio for MP$(k-1)$EC and not for MPkEC). The best known values for α are: $\alpha = \lceil (k+1)/2 \rceil$ for $2 \leq k \leq 7$ (see [2] for $k = 2, 3$, [6] for $k = 4, 5$, and [14] for $k = 6, 7$), $\alpha = k$ for other small values of k [14], and $\alpha = O\left(\log k \cdot \log \frac{n}{n-k}\right)$ otherwise [20]. Thus for undirected MPkCS the following ratios follow: $3k$ for any k, $k + 2\lceil (k+1)/2 \rceil$ for $2 \leq k \leq 7$, $O(\log n)$ unless $k = n - o(n)$, and $O(\log^2 n)$ if $k = n - o(n)$.

Improvements over the above ratios for MPkCS are known only for $k \leq 5$: $(2k - 1/3)$ for $k \in \{2, 3\}$ [13], and 9 for $k = 4$ [11].

For further results on other min-power connectivity problems, among them problems on directed graphs see [9,21,17]. For results on min-cost k-connectivity problems see [2,6,14,5,15,7,20,18]; see also a recent survey in [16] on various min-cost connectivity problems.

1.3 Results

The previously best known ratio for MPkIS was $\min\{k + 4, O(\log n)\}$ [17]. We improve the ratio $k + 4$ for $k = O(\log n)$ as follows:

Theorem 2. *Undirected* MPkIS *admits a $(k+1)$-approximation algorithm.*

Combining Theorem 2 with a direct analysis of the algorithms in [2,6,14] for MCkCS, we obtain the following result:

Theorem 3. *Suppose that* MPkIS *admits a* γ*-approximation algorithm and that* MPkDP *admits a* θ*-approximation algorithm. Then* MPkCS *admits the following approximation ratios:* $\gamma + \theta(k-2)$ *for any constant* k *and* $\gamma + \theta(\lfloor k/2 \rfloor - 1)$ *for* $k \leq 7$*. In particular, for* $k \leq 7$ *the ratios are:* γ *for* $k \in \{2,3\}$*,* $\gamma + \theta$ *for* $k \in \{4,5\}$*, and* $\gamma + 2\theta$ *for* $k \in \{6,7\}$*.*

As MPkDP admits a 2-approximation algorithm (c.f., [9,17]), then by combining Theorems 2 and 3 we obtain:

Theorem 4. MPkCS *admits the following approximation ratios:* $k+1$ *for* $k \in \{2,3\}$*,* $k+3$ *for* $k \in \{4,5\}$*,* $k+5$ *for* $k \in \{6,7\}$*, and* $3(k-1)$ *for any constant* k*.*

Theorem 4 significantly improves the previously best known ratios for MPkCS with $2 \leq k \leq 7$, as summarized in the following table:

Table 1. Approximation ratios for MPkCS

k	Prior art	This paper
1	$(5/3+\varepsilon)$ [1]	–
2	$(3+2/3)$ [13]	3
3	$(5+2/3)$ [13]	4
4	9 [11]	7
5	11 [11,9]	8
6	14 [11,9]	11
7	15 [11,9]	12
constant k	$3k$ [11,9]	$3k-3$

Theorems 2 and 3 are proved in Sections 2 and 3, respectively.

2 Algorithm for MPkIS (Proof of Theorem 2)

A *bi-direction* of an undirected network H is a directed network obtained by replacing every edge $e = uv$ of H by two opposite directed edges uv, vu each having the same cost as e. Clearly, if D is a bi-direction of H, then $p(H) = p(D)$. The *underlying network* of a directed network D is a network H obtained from D by ignoring the directions (but keeping costs) of the edges, and then keeping one (arbitrary) edge from every maximal set of parallel edges, if non-empty. If H is the underlying network of a directed star D with unit costs, then $p(H) = (\Delta(D)+1)p(D)$. The following statement shows that this is the extremal case for general costs.

Lemma 1. $p(H) \leq (\Delta(D)+1)p(D)$ *for the underlying network* H *of a directed network* D*.*

Proof. By induction on the number m of edges in D. For $m = 1$ the statement is obvious. Assume that the statement is true for digraphs with at most $m-1$

edges. Let $v \in V$ be a node in D of maximum power $c_{\max}$. Let D' be obtained from D by removing the edges leaving v, and let H' be the underlying graph of D'. Clearly, $p(D') = p(D) - c_{\max}$ and $p(H') \geq p(H) - (\Delta(D)+1)c_{\max}$. Combining with the induction hypothesis gives:

$$\begin{aligned} p(H) &\leq p(H') + (\Delta(D)+1)c_{\max} \\ &\leq (\Delta(D)+1)(p(D) + c_{\max}) \\ &= (\Delta(D)+1)p(D) \ . \end{aligned}$$

We need several results from [17].

Theorem 5 ([17]). *Directed* MPkIS *can be solved in polynomial time.*

Definition 2. *An edge e of a k-inconnected to s graph J is* critical *if $J - e$ is not k-inconnected to s. A graph is minimally k-inconnected to s if all its edges are critical.*

Theorem 6 ([17]). *Let uv' and uv'' be two distinct critical edges of a k-inconnected to s directed graph J. Then $d_J(u) = k$. In particular, $d_J(u) = k$ for every node $u \neq s$ if J is minimallyk-inconnected to s.*

The $(k+1)$-approximation algorithm for MPkIS is as follows:

1. Let D be the bi-direction of G.
2. Compute a min-power k-inconnected to s spanning subgraph J of D.
3. Return the underlying graph H of J.

Step 2 can be implemented in polynomial time using the algorithm of [17] (Theorem 5). We now show that the approximation ratio of the algorithm is $k+1$. Let H^* be an optimal solution to MPkIS instance (so $p(H^*) = \mathsf{opt}$) and let J^* be the bi-direction of H^*. Let H and J be as in the algorithm. Combining Theorem 6 with Lemma 1 we get:

$$\begin{aligned} p(H) &\leq (\Delta(J)+1)p(J) \\ &\leq (k+1)p(J) \\ &\leq (k+1)p(J^*) \\ &\leq (k+1)p(H^*) \\ &= (k+1)\mathsf{opt} \ . \end{aligned}$$

The proof of Theorem 2 is complete.

3 Algorithm for MPkCS (Proof of Theorem 3)

We need the following summary of several statements from [2,6,14].

Lemma 2 ([2,6,14]). *Let $H = (V, I)$ be k-inconnected to s graph with $d_H(s) = k$. Then one can find in polynomial time a set F of at most $k-2$ new edges on the neighbors of s in H so that $H + F$ is k-connected. Furthermore, $|F| \leq \lfloor k/2 \rfloor - 1$ for $k \leq 7$.*

Halin [10] proved that any minimally k-connected graph has a node of degree k. A stronger statement was proved by Mader [19]:

Theorem 7 ([19]). *A minimally k-connected graph contains at least $\frac{(k-1)n+2}{2k-1}$ nodes of degree k.*

This motivates the following auxiliary problem, which min-cost variant is the basis for the algorithms in [2,6,14].

Restricted MPkIS
Instance: A graph $G=(V,E)$, edge costs $\{c(e) : e \in E\}$, $s \in V$, an integer k.
Objective: Find a min-power k-inconnected to s spanning subgraph H of G with $d_H(s) = k$.

Lemma 3. *If* MPkIS *admits a γ-approximation algorithm then* Restricted MPkIS *admits a γ-approximation algorithm for any constant k.*

Proof. The algorithm for Restricted MPkIS is derived from the algorithm for MPkIS by "guessing" the k edges incident to s in some optimal solution for Restricted MPkIS. For any subset $K \subseteq E$ of k edges incident to s, we remove the other edges incident to s, and compute a γ-approximate solution H_K to MPkIS (or declare that the resulting graph is not k-inconnected to s). Then, among the subgraphs H_K computed, we output one H of the minimum power. The running time is $\binom{n}{k} = O(n^k)$ times the running time of the γ-approximation algorithm for MPkIS, hence polynomial for any constant k.

Remark: In [6], it was shown that the min-cost version of directed Restricted MPkIS is solvable in polynomial time; this was done by using the algorithm of [8] for the min-cost version of directed MPkIS and penalty methods. Although MPkIS is solvable in polynomial time [17], it seems that the penalty method used in [6] does not work for directed Restricted MPkIS.

We now finish the proof of Theorem 3. The algorithm is as follows:

1. For every $s \in V$, compute a γ-approximate solution H_s to Restricted MPkIS with G, s.
   Among the subgraphs H_s computed, let H be one of the minimum power.
2. Compute an edge set F as in Lemma 2.
3. For every $uv \in F$ compute a 2-approximate solution for MPkDP in $G, \{u, v\}$.
4. Return $H + \bigcup\{F_{uv} : uv \in F\}$.

The fact that the returned graph is k-connected was already established in [2,14], and easily follows from the definition of F. For any constant k, Step 1 can be implemented in polynomial time, by Lemma 3. All the other steps can be implemented in polynomial time for any k. Thus the running time is polynomial for any constant k, as claimed.

We prove the approximation ratio. Note that a k-connected graph is also k-inconnected to s for every $s \in V$. Let H^* be some optimal solution to MPkCS;

clearly, we may assume that H^* is minimally k-connected. From Theorem 7 it follows that there is a node $s \in V$ so that the degree of s in H is k. Thus for the graph H computed at Step 1 we have $p(H) \leq \gamma p(H^*) = \gamma\mathsf{opt}$. Also, H^* contains k internally disjoint uv for all $u, v \in V$. Thus $F_{uv} \leq \theta\mathsf{opt}$ for all $uv \in F$. Consequently,

$$\begin{aligned} p\left(H + \bigcup\{F_{uv} : uv \in F\}\right) &\leq p(H) + \sum_{uv \in F} p(F_{uv}) \\ &\leq \gamma\mathsf{opt} + \theta|F|\mathsf{opt} \\ &= (\gamma + \theta|F|)\mathsf{opt} \ . \end{aligned}$$

Substituting the sizes of F from Lemma 2 we obtain the following. For any k we have $|F| \leq k-2$, and thus in this case the approximation ratio is $\gamma + \theta|F| = \gamma + \theta(k-2)$. For $k \leq 7$ we have $|F| \leq \lfloor k/2 \rfloor - 1$, and thus in this case the approximation ratio is $\gamma + \theta|F| = \gamma + \theta(\lfloor k/2 \rfloor - 1)$. Substituting the specific values of k, we obtain the last statement of the Theorem.

The proof of Theorem 3 is complete.

4 Open Problems

The main open problem in the context of this paper is to determine whether the *undirected* MPkDP is in P or is NP-hard (the directed MPkDP is in P, c.f., [9]). If MPkDP is in P, then we can substitute $\theta = 1$ in Theorem 3 and obtain the following ratios for MPkCS: $2k-1$ (instead of $3k-3$) for any constant k, and $k + \lfloor k/2 \rfloor$ (improving $k - 1 + 2\lfloor k/2 \rfloor$) for $k \leq 7$.

We note that we do not know the answer even to the following "easier" question. Let MPkDP Augmentation be the restriction of MPkDP to instances where $E_0 = \{e \in E : c(e) = 0\}$ contains $k-1$ pairwise internally disjoint paths. We do not know if (undirected) MPkDP Augmentation is in P, but we conjecture this is so. A polynomial algorithm for MPkDP Augmentation can be used to improve the ratios for MPkCS for $k = 4, 5$: from 7 to 6 for $k = 4$ and from 8 to 7 for $k = 5$. This is since in [2,6] it is shown that if H is k-inconnected to s and $d_H(s) = k$ then H is $(\lceil k/2 \rceil + 1)$-connected. Thus for $k = 4, 5$, H is $k-1$-connected, and, by Lemma 2, H contains two nodes u, v so that increasing the connectivity between them by one results in a k-connected graph.

Except directed MPkDP and MPkIS that are in P, there is still a large gap between upper and lower bounds of approximation for many other min-power node connectivity problems, for both directed and undirected graphs, see [21,17,12].

References

1. Althaus, E., Calinescu, G., Mandoiu, I., Prasad, S., Tchervenski, N., Zelikovsky, A.: Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks. Wireless Networks 12(3), 287–299 (2006)

2. Auletta, V., Dinitz, Y., Nutov, Z., Parente, D.: A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph. J. of Algorithms 32(1), 21–30 (1999)
3. Calinescu, G., Kapoor, S., Olshevsky, A., Zelikovsky, A.: Network lifetime and power assignment in ad hoc wireless networks. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 114–126. Springer, Heidelberg (2003)
4. Calinescu, G., Wan, P.J.: Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks. Mobile Networks and Applications 11(2), 121–128 (2006)
5. Cheriyan, J., Vempala, S., Vetta, A.: An approximation algorithm for the minimum-cost k-vertex connected subgraph. SIAM J. on Computing 32(4), 1050–1055 (2003)
6. Dinitz, Y., Nutov, Z.: A 3-approximation algorithm for finding optimum 4,5-vertex-connected spanning subgraphs. J. of Algorithms 32(1), 31–40 (1999)
7. Fackharoenphol, J., Laekhanukit, B.: An $O(\log^2 k)$-approximation algorithm for the k-vertex connected spanning subgraph problem. In: STOC, pp. 153–158 (2008)
8. Frank, A., Tardos, E.: An application of submodular flows. Linear Algebra and its Applications 114/115, 329–348 (1989)
9. Hajiaghayi, M.T., Kortsarz, G., Mirrokni, V.S., Nutov, Z.: Power optimization for connectivity problems. Math. Program. 110(1), 195–208 (2007); Preliminary version in IPCO 2005
10. Halin, R.: A theorem on n-connected graphs. J. Combinatorial Theory 7, 150–154 (1969)
11. Jia, X., Kim, D., Makki, S., Wan, P.J., Yi, C.W.: Power assignment for k-connectivity in wireless ad hoc networks. J. Comb. Optim. 9(2), 213–222 (2005); Preliminary version in INFOCOM 2005
12. Kortsarz, G., Mirrokni, V.S., Nutov, Z., Tsanko, E.: Approximating minimum-power degree and connectivity problems. In: LATIN, pp. 423–435 (2008)
13. Kortsarz, G., Nutov, Z.: Approximating minimum-power edge-covers and 2,3-connectivity. Manuscript (submitted for journal publication)
14. Kortsarz, G., Nutov, Z.: Approximating node-connectivity problems via set covers. Algorithmica 37, 75–92 (2003)
15. Kortsarz, G., Nutov, Z.: Approximating k-node connected subgraphs via critical graphs. SIAM J. on Computing 35(1), 247–257 (2005)
16. Kortsarz, G., Nutov, Z.: Approximating minimum-cost connectivity problems. In: Gonzalez, T.F. (ed.) Approximation algorithms and Metaheuristics, ch. 58. Chapman & Hall/CRC, Boca Raton (2007)
17. Lando, Y., Nutov, Z.: On minimum power connectivity problems. Preliminary version in ESA 2007, pp. 87–98 (manuscript, submitted for journal publication)
18. Lando, Y., Nutov, Z.: Inapproximability of survivable networks. In: APPROX (to appear 2008)
19. Mader, W.: Ecken vom grad n in minimalen n-fach zusammenhängenden graphen. Archive der Mathematik 23, 219–224 (1972)
20. Nutov, Z.: An almost O(logk)-approximation for k-connected subgraphs (manuscript)
21. Nutov, Z.: Approximating minimum power covers of intersecting families and directed connectivity problems. In: Díaz, J., Jansen, K., Rolim, J., Zwick, U. (eds.) APPROX 2006. LNCS, vol. 4110, pp. 236–247. Springer, Heidelberg (2006)
22. Schrijver, A.: Combinatorial Optimization, Polyhedra and Efficiency. Springer, Berlin (2004)

A Secure Cross-Layer Protocol for Multi-hop Wireless Body Area Networks

Dave Singelée[1], Benoît Latré[2], Bart Braem[3], Michael Peeters[4], Marijke De Soete[4], Peter De Cleyn[3], Bart Preneel[1], Ingrid Moerman[2], and Chris Blondia[3]

[1] ESAT–SCD–COSIC, Katholieke Universiteit Leuven — IBBT, Kasteelpark Arenberg 10, 3001 Heverlee-Leuven, Belgium
dave.singelee@esat.kuleuven.be
[2] IBCN, Dept. of Information Technology (INTEC), Ghent University — IBBT, Gaston Crommenlaan 8, bus 201, 9050 Gent, Belgium
[3] PATS, Dept. of Mathematics and Computer Sc., University of Antwerp — IBBT, Middelheimlaan 1, B-2020, Antwerp, Belgium
[4] NXP Semiconductors, Competence Center System Security & DRM, A&I Innovation & Development Center Leuven, Interleuvenlaan 74–82, 3001 Leuven, Belgium

Abstract. The development of Wireless Body Area Networks (WBANs) for wireless sensing and monitoring of a person's vital functions, is an enabler in providing better personal health care whilst enhancing the quality of life. A critical factor in the acceptance of WBANs is providing appropriate security and privacy protection of the wireless communication. This paper first describes a general health care platform and pinpoints the security challenges and requirements. Further it proposes and analyzes the CICADA-S protocol, a secure cross-layer protocol for WBANs. It is an extension of CICADA, which is a cross-layer protocol that handles both medium access and the routing of data in WBANs. The CICADA-S protocol is the first integrated solution that copes with threats that occur in this mobile medical monitoring scenario. It is shown that the integration of key management and secure, privacy preserving communication techniques within the CICADA-S protocol has low impact on the power consumption and throughput.

1 Introduction

Recent progress in wireless sensing and monitoring, and the development of small wearable or implantable biosensors, have led to the use of Wireless Body Area Networks (WBANs). The research on communication within a WBAN is still in its early stages. Only few protocols designed specifically for multi-hop communication in WBANs exist. They try to minimize the thermal effects of the implanted devices by balancing the traffic over the network [1] or by forming clusters [2,3] or a tree network [4].

Wireless Body Area Networks can be seen as an enabling technology for mobile health care [5]. Medical readings from sensors on the body are sent to servers at

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 94–107, 2008.
© Springer-Verlag Berlin Heidelberg 2008

the hospital or medical centers where the data can be analyzed by professionals. These systems reduce the enormous costs associated to ambulant patients in hospitals as monitoring can take place even at home in real-time and over a longer period.

In this paper, we propose and analyze CICADA-S, a secure protocol for WBANs. It is based on an existing multi-hop protocol for WBANs, called CICADA [4]. This is a cross-layer protocol that sets up a data gathering tree in a reliable manner, offering low delay and high energy efficiency. The communication of health related information between sensors in a WBAN and over the Internet to servers is strictly private and confidential and should therefore be encrypted to protect the patient's privacy. Furthermore, the medical staff who collects the data must be confident that the data is not tampered with, and indeed originates from that patient.

The CICADA-S protocol is designed within the scope of the IBBT IM3-project (Interactive Mobile Medical Monitoring), which focuses on the research and implementation of a wearable system for health monitoring [6]. Patient data is collected using a WBAN and analyzed at the gateway (also called medical hub) worn by the patient. If an event (e.g., heart rhythm problems) is detected, a signal is sent to a health care practitioner who can view and analyze the patient data remotely.

The remainder of this paper is organized as follows. Section 2 gives an overview of related work. The general architecture and the necessary security assumptions are described in section 3. A short description of CICADA is given, followed by the integration of the security mechanisms in the protocol and a description of the key management aspects in section 4. The analysis of the integration in terms of performance overhead and security properties are dealt with in section 5. Finally, section 6 provides a final conclusion on the paper.

2 Related Work

Security is essential for broad acceptance and further growth of Wireless Sensor Networks. These networks pose unique challenges as security techniques used in traditional networks cannot be directly applied. Indeed, to make sensor networks economically viable, sensor devices should be limited in their energy consumption, computation, and communication capabilities. Since most of the existing security mechanisms have major drawbacks in that respect, new ideas are needed to address these requirements in an appropriate way [7].

One of the most crucial components to support the security architecture of a Wireless Sensor Network is its key management. During the last years, a number of pairwise key establishment schemes have been proposed. Zhou and Haas propose to secure ad-hoc networks using asymmetric cryptography [8]. They use threshold cryptography to distribute trust among a set of servers. This scheme achieves a high level of security, but is too energy consuming to be used in practice in a Wireless Sensor Network. Eschenauer and Gligor introduce a key management scheme for distributed sensor networks [9]. It relies on probabilistic

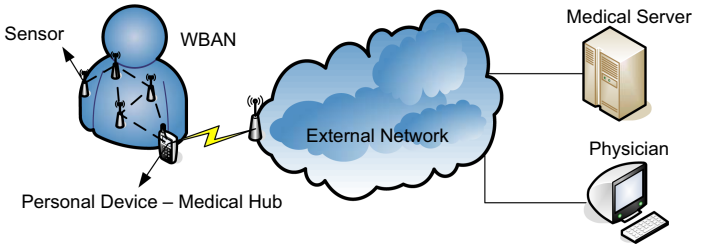


Fig. 1. General overview of the IM3 health care architecture

key sharing among the nodes of a random graph. Perrig et al. present SPINS, a suite of security building blocks optimized for resource-constrained environments and wireless communication [10]. It has two secure building blocks: SNEP and μTESLA. SNEP provides data confidentiality, two-party data authentication and data freshness, while μTESLA offers authenticated broadcast in constrained environments.

The security mechanisms employed in Wireless Sensor Networks do generally not offer the best solutions to be used in Wireless Body Area Networks for the latter have specific features that should be taken into account when designing the security architecture. The number of sensors on the human body, and the range between the different nodes, is typically quite limited. Furthermore, the sensors deployed in a WBAN are under surveillance of the person carrying these devices. This means that it is difficult for an attacker to physically access the nodes without this being detected. When designing security protocols for WBANs, these characteristics should be taken into account in order to define optimized solutions with respect to the available resources in this specific environment.

Although providing adequate security is a crucial factor in the acceptance of WBANs, little research has been done in this specific field [11]. In [12] an algorithm based on biometric data is described that can be employed to ensure the authenticity, confidentiality and integrity of the data transmission between the personal device and all the other nodes. Another method is presented in [13] where body-coupled communication (BCC) is used to associate new sensors in a WBAN.

None of the current protocols offer a solution where appropriate security mechanisms are incorporated into the communication protocol while addressing the lifecycle of the sensors. Further, security and privacy protection mechanisms use a significant part of the available resources and should therefore be energy efficient and lightweight. The mechanisms proposed in this paper aim to cover these challenges.

3 Architecture

3.1 General Overview

Fig. 1 shows the health care architecture used by the IM3 project. There are three main components: the Wireless Body Area Network (WBAN), the external

network and the back-end server. In this scenario, the WBAN contains several sensors that measure medical data such as ECG, body movement etc. These sensors send their measurements, directly or via several hops, to the gateway. Each WBAN (and hence every patient) has its unique gateway. In other words, the sensors shall only send their data to the unique gateway they are linked with and this needs to be enforced by specific security mechanisms. The gateway processes the medical data, and sends the result via the external network to the back-end server at the hospital, where it can be observed and analyzed by medical staff.

Although the architecture was originally designed for and is fully adapted to a medical environment, it may also be used in other applications. Indeed, as long as the (security) relations between the different devices remain valid, the protocol remains applicable, which increases the generality of our solution. In the remainder of this paper, the medical scenario will be further used to explain the architecture and the secure cross-layer protocol for multi-hop WBANs.

3.2 Security Assumptions

This section aims to address the security of the entire system, and the WBAN in particular.

The most security critical device in the entire architecture is the back-end server. This server, which is managed by the hospital or medical center, will receive the medical data sent by all active WBANs. It is assumed that this server is physically protected (e.g., put in a secure place in the hospital where it can not be stolen or tampered with), and that an adequate access control system is implemented (i.e. only authorized medical personnel has (partial) access to the server through appropriate identification/authentication mechanisms). The back-end server is considered to be a trusted third party, which means that it is known and trusted by all other devices in the network after a successful authentication.

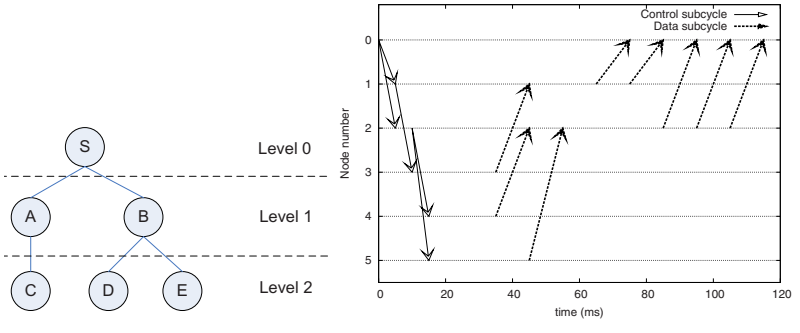
Since potentially security critical data will be transferred through the external network, end-to-end security between the gateway and the back-end server is required. For efficiency reasons, it is assumed that both devices share a symmetric session key to secure their communication. This symmetric session key can be manually installed (e.g., pre-installed during manufacturing), or (preferably) established via a symmetric key establishment protocol. The description of such protocols can be found in the ISO 9798–2 standard, and is out of scope of this article. The symmetric session key is updated regularly. The end-to-end channel between gateway and back-end server should also be anonymized using temporary pseudonyms. This avoids privacy problems like (location) tracking. In the remainder of the paper, it is assumed that the secure end-to-end channel between gateway and back-end server is already established after a successful mutual authentication. As mentioned before, each gateway belongs to a specific WBAN (i.e. a patient, who is carrying this device). To enforce this, the gateway is registered in advance at the back-end server.

It is assumed that it is impossible to alter or read the memory of a (securely initialized) node that is put on the patient's body, or to modify the behavior of a node without this being detected. This is not a strong assumption, since the patient is carrying the nodes on its body, and an attacker is not able to access the nodes without this being detected. It is also assumed that the attacker has no access to the sensors that yet have to be securely initialized (e.g., because they are stored in a safe place). However, an attacker can put a malicious node in the presence of a WBAN, and try to join the network. He can also eavesdrop on all data transmitted in the WBAN, and insert/delete/modify (malicious) data into the network. The attacker is hence assumed to be active.

4 Protocol Design

4.1 CICADA

CICADA is a cross-layer protocol as it handles both medium access and the routing of data [4]. The protocol sets up a spanning tree in a distributed manner, which is subsequently used to guarantee collision free access to the medium and to route data toward the gateway. The time axis is divided in slots grouped in cycles, to lower the interference and avoid idle listening. Slot assignment is done in a distributed way where each node informs its children when they are allowed to send their data using a SCHEME. Slot synchronization is possible because a node knows the length of each cycle. During a cycle, a node is allowed to send all of its data to its parent node. CICADA is designed in such a way that all packets arrive at the source in only one cycle. Routing itself is not complicated in CICADA anyway as data packets are routed up the tree which is set up to control the medium access, no special control packets are needed.



(a) Sample topology

(b) Packet streams in this network. Notice the down- and upstream traffic

Fig. 2. Communication in CICADA for a sample network of 5 nodes

A cycle is divided in a control subcycle consisting of control slots, and a data subcycle consisting of data slots. The former is used to broadcast a SCHEME

message from parent to child, i.e. to let the children know when they are allowed to send in the data subcycle. In the data subcycle, data is forwarded from the nodes to the gateway. In each data subcycle, a contention slot is included to allow nodes to join the tree. New children hear the SCHEME message of the desired parent and send a JOIN-REQUEST message in the contention slot. When the parent hears the JOIN-REQUEST message, it will include the node in the next cycle. Each node will send at least two packets per cycle: a data packet or HELLO packet (if no data is sent) and a SCHEME packet. If a parent does not receive a packet from a child for N or more consecutive cycles, the parent will consider the child to be lost. If a child does not receive packets from its parent for N or more consecutive cycles, the child will assume that the parent is gone and will try to join another node. An example of communication in CICADA is given in Fig. 2, for a network of 5 nodes. The control and data subcycles can be seen clearly.

A node informs its parent node of the number of slots it needs to send its own data and forward data coming from its children, by calculating two parameters: α and β. The former gives the number of slots needed for sending data (including forwarded data) to its parent, the latter gives the number of slots the node has to wait until it has received all data from its children. Based on the α and β from its children, a node can calculate the slot allocation for the next cycle.

4.2 CICADA-S

The CICADA protocol, as described in the previous section, does not guarantee any form of security and privacy. Unauthorized nodes can easily join the WBAN, and all communication in the network is sent in plain text and is not integrity protected. The fixed identity of the sensors is not kept confidential, and can hence be used to track sensors (and patients carrying these sensors). To counter these problems, appropriate security mechanisms have to be added to the CICADA protocol. The result is the CICADA-S protocol, the secure version of the CICADA protocol.

From a security point of view, there are four main states which take place during the lifetime of a sensor: the secure initialization phase, the sensor (re)joining the WBAN, a key update procedure in the WBAN, and the sensor leaving the WBAN. The security mechanisms used in these phases and their integration into the CICADA-S protocol, based on the results of [6], will now be described.

Secure Initialization Phase: Initially, each sensor has to be securely initialized by the back-end server before it can join the WBAN in a later stage. During this initialization phase, the sensor and the back-end server will agree on a shared symmetric key. This can be done via asymmetric cryptographic techniques, but this is typically too energy (and computation) consuming for a regular sensor. Another way of establishing a shared key, is by using a private and authentic out-of-band channel. Such a channel is typically cheap to setup. It has the interesting property that all data transmitted on the channel remains confidential for eavesdroppers, and that the integrity and authenticity is protected too.

A private and authentic channel can be created in several ways, depending on the exact hardware and (physical) characteristics of the sensors. It can be established by connecting the sensor directly to the back-end server, via an extra electrical contact available on both devices. Other techniques to create such a secure out-of-band channel is by employing distance bounding protocols, by having the user manually enter the data on both devices etc. More information on these and other techniques to establish a private and authentic out-of-band channel can be found in the literature [14,15,16].

Let us assume that sensor A has to be initialized. The data transfer via the secure out-of-band channel takes place in two steps. First, the sensor sends its fixed identity to the back-end server. This can be done explicitly or implicitly (the identity of the sensor can be implicitly known because of the specific characteristics of the out-of-band channel). In the second step of the protocol, the back-end server generates a random secret key (k_A), and sends this key securely to the sensor. The sensor and the back-end server store this secret key in their memory. The key is (conceptually) composed out of 2 subkeys: the encryption key k_{A_encr} and the integrity key k_{A_int}. Note that each new node is assigned a new and unique secret key.

Each sensor i is also assigned a unique counter CTR_i, which is initialized to 0 and stored in the sensor's memory. The value of this counter is included in all key management messages, and is used to avoid replay attacks and assure freshness. Every time the counter is used, the value gets incremented by 1.

Sensor (Re)joining the WBAN: After the initialization procedure, the sensor is ready to be put on the patient's body. It will detect the WBAN, and start the join procedure, which will now be discussed.

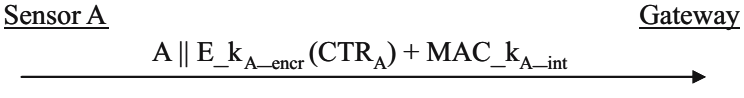


Fig. 3. Secure JOIN-REQUEST originating from sensor A

When the sensor (with fixed identity A) hears the SCHEME of the desired parent, it sends a secure JOIN-REQUEST message, as shown in Fig. 3, in the contention slot. This message is forwarded to the gateway. It is basically a HELLO message containing the unique (global) identity of the sensor and the value of its unique counter CTR_A. The counter is encrypted for privacy reasons (since it is used in all key management messages). The gateway stores (and updates) this value of the counter. The integrity and authenticity of the entire secure JOIN-REQUEST message is protected by a message authentication code (MAC) [17], computed with the key k_{A_int}.

When the gateway receives the secure JOIN-REQUEST message of sensor A, it forwards this request to the back-end server via the secure end-to-end channel. This triggers a protocol in which the key k_A is securely transported from the

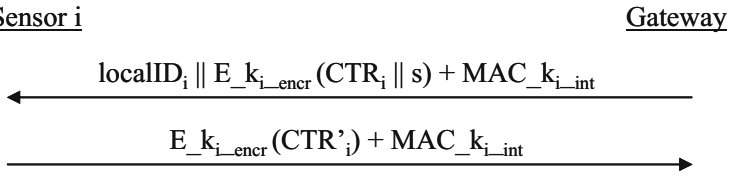


Fig. 4. Secure key transport to all the sensors in the WBAN

back-end server to the gateway. More information on how to accomplish this, can be found in the ISO 9798–2 standard [18]. In some scenarios, and this is often the case in a medical environment, it is known in advance (e.g., already during the initialization procedure) in which WBAN the sensor will be deployed. In this case, the back-end server can already transport the key k_A to the correct gateway, and does not have to wait until it receives the secure JOIN-REQUEST message. This makes the join procedure faster. In the case a sensor leaves the network, and (not much) later rejoins it, the gateway may still have the key k_A in its memory and does not have to forward the request to the back-end server. From the moment the gateway has access to the key, it can check the validity of the JOIN-REQUEST by verifying the message authentication code, and in case of a rejoin, also the value of the counter CTR_A (the new value should be higher than the current value shared by sensor and gateway). If this verification is successful, the sensor is allowed to join the WBAN and is assigned a temporary identity $localID_A$. This temporary identity, which is chosen by the gateway, is established in order to preserve the privacy. It is only unique within the environment of the WBAN. Other networks can reuse the same identifier. Since the bitlength of such a local identifier can be smaller than the full identity of the sensor (A), it also improves the efficiency. A joining sensor in the WBAN is informed about its temporary identity during the key transport procedure, which takes place immediately after the approval of the secure JOIN-REQUEST message.

Key Update Procedure in the WBAN: Except for the key management messages, the data traveling in the WBAN consists of schemes sent during the control subcycle, and medical data sent during the data subcycle from the sensors to the gateway. The former is only integrity protected (to allow a new node to inform itself about the contention slot), while the latter is both integrity protected and encrypted. All these operations are performed by employing a secret group key s, that is shared between all the sensors in the WBAN. Every time a node joins or leaves the network, the group key is updated in order to avoid an attacker recovering the key. Even when the topology of the network remains constant for a long time, the group key should still be updated at regular intervals. The exact period is determined by the cryptographic strength of the encryption and integrity algorithms used to protect the data in the WBAN, and the length of the key. We will briefly come back to this in section. 5.1.

The update process works as follows. First, the gateway randomly generates a new group key s. Next, it performs a secure key transport procedure with all the nodes in the WBAN, as shown in Fig. 4. The gateway constructs a key

update message, unique for every sensor, which contains the encrypted value of the updated group key s. For each node i, the message also contains the new value of the counter CTR_i (which is the current value of the counter incremented by 1), in order to avoid replay attacks, and the local identifier $localID_i$. The authenticity and the integrity of the message is protected by a message authentication code. Nodes that have been excluded from the WBAN, can not decrypt the key transport messages anymore, and are hence not able to obtain the new group key s.

The key update message is uniquely constructed for every sensor, and forwarded from the gateway to the correct node during the control subcycle. Each node takes the message containing its local identifier, checks the validity of the message (by verifying the value of the counter and the message authentication code) and decrypts the encrypted part in order to recover the new value of the group key s. It also forwards all other key update messages to its children, who perform the same procedure. A new joining node A does not yet know its local identifier $localID_A$, and therefore has to check the message authentication code (and the counter) of all the key update messages using its key k_{A_int} until the test succeeds. This only has to be done once, and is easily feasible since computing a message authentication code can be done very efficiently. The joining sensor stores its local identifier $localID_A$ in its memory, and recovers the group key s from the encrypted part of the key update message. Finally, all sensors send a secure acknowledgement back to the gateway during the next data subcycle, to inform that they received the key well. This key confirmation message only contains the encrypted value of the updated counter CTR_i, concatenated with a message authentication code. After having received the key confirmation message, the gateway knows it can definitively update the group key. When a node does not send its key confirmation message within a certain period, e.g., because it did not receive the new group key s due to packet loss, the gateway retransmits the key transport message to that particular node.

Sensor leaving the WBAN: When a node detects that a particular sensor A is not part anymore of the WBAN, it forwards this information to the gateway. This automatically triggers a group key update procedure. This has to be done in order to avoid that an attacker stealing a sensor from the network, would be able to read or modify the data in the WBAN. After a certain interval (or even immediately, depending on the policy), the gateway deletes the key k_A and the identifier $localID_A$ from its memory. If the medical staff removes sensor A from the patient, or if the sensor is reported lost or stolen, the key k_A should also be deleted from the memory of the back-end server. This way, the sensor can not rejoin any network anymore in a later stage, until it has been securely reinitialized by the back-end server.

5 Analysis

5.1 Performance Evaluation

The addition of these security mechanisms to CICADA undoubtedly influences the performance as it leads to an increased overhead and higher delay. The

exact impact strongly depends on the choice of the cryptographic algorithms that are deployed in the WBAN, and it is hence difficult to formulate results that are generally applicable. That is why a worst case analysis will be given, in which we assume that a secure block cipher, such as the Advanced Encryption Standard (AES) [19], is employed in an authenticated encryption mode (e.g., CCM or GCM mode of operation). The numbers used below are based on the guidelines of the National Institute of Standards and Technology (NIST) [20,21]. In practice, it would be better to employ a low-cost encryption and integrity algorithm, which has a slightly lower security level, but is more efficient.

The combined encryption and authentication algorithm uses a symmetric key of 16 bytes (the group key s or the shared key k_i). The output of this method are encrypted blocks of 16 bytes, and a message authentication code of at least 8 bytes. Furthermore, the unique hardware address of the sensor is assumed to be 6 bytes (e.g., as in Bluetooth), and a counter of 4 bytes is employed to avoid replay attacks. Note that encrypting the counter results in an encryption block of 16 bytes. Using these parameters offers a high level of security as long as the keys are updated regularly, which depends on the strength of the cryptographic algorithm that is being used. E.g., when AES is used in the GCM mode of operation, the group key s should be updated at least at every 2^{32}th invocation of the encryption algorithm [21]. In this section, we will now briefly discuss the (worst case) impact of the security mechanisms on the CICADA protocol, using the numbers stated above.

In the (re)joining phase, additional information is sent to the gateway in the JOIN-REQUEST message. The original CICADA-message only contains $localID_A$ and $localID_P$ (i.e. the local ID of node A joining the network and the local ID of the desired parent P respectively). The length of these IDs is 1 byte, which is sufficient for a WBAN. In CICADA-S the unique hardware address of the sensor is sent, together with the encrypted synchronized counter and a message authentication code. The length of the JOIN-REQUEST message thus is longer, but still only 30 bytes. As this information is sent in a contention slot with fixed size, this will not influence the throughput of the system. However, this secure JOIN-REQUEST message needs to be forwarded to the gateway. As the contention slot of a node is in the beginning of a data subcycle, the message can be sent to the gateway directly. E.g., the JOIN-REQUEST message can be piggybacked on a data packet that is sent to the gateway. As the length of the message is small, this may not influence the overall throughput significantly. The number of bytes that can be sent in one slot depends on the size of the slot and the raw bit rate of the radio technology used. If the number of bytes in the data packet and the secure JOIN-REQUEST message is too large, the slot size will have to be altered. This will lower the throughput of the network. A better solution is to send the JOIN-REQUEST message in a separate data slot. This will hardly impact the throughput of the network. If the key is already present at the gateway, the gateway can immediately start the key update procedure. If not, the gateway has to wait for a response from the back-end server. This will add extra delay to the joining procedure.

In the key update procedure, the gateway sends a new key to all the nodes in the control subcycle. This message contains $localID_A$, the new key group key s concatenated with an increased counter (both encrypted), and a message authentication code. For each node, this is an additional 41 bytes. Due to the broadcast mechanism in the control subcycle, these messages all need to be broadcasted by every node sending its SCHEME in the control subcycle. This will lead to a larger slot length in the control subcycle, and subsequently a lower throughput. In CICADA, the slot length in the control subcycle is smaller than the data slot length as the SCHEME-messages sent in the control subcycle are very short. The slot length can be up to ten times smaller. This improves the energy throughput of CICADA. As the key is only updated after several cycles, we opt to change the control slot dynamically. When the key is updated, the control slot length has the same length as the data slot. At any other time, the control slot has its shorter length. When the key is about to be updated, the gateway broadcasts a warning in the previous cycle by setting a bit in the header. The nodes receive this warning and adapt their control slot lengths for one cycle.

When a node leaves the network or is no longer attached to it, the (former) parent node sends a message to the gateway. This can be added to a data packet and will not influence the throughput.

It is very important to note that the key management messages are sent rarely (only when a node (re)joins the network, or when the group key has to be updated), and hardly affect the global throughput in the network. Most data traveling in the WBAN is medical data, sent by the sensors to the gateway. These messages are protected by employing the group key s. The data is encrypted in blocks of 16 bytes, and a message authentication code of 8 bytes is added. The SCHEME packets sent during the control subcycle are not encrypted, but integrity protected. For both types of data, the length of the messages is hardly influenced. Overall, the security mechanisms will have a minor impact on the performance of CICADA-S.

5.2 Security Properties

One of the design goals of the CICADA-S protocol is to secure the wireless communication in the WBAN while preserving privacy. The most interesting security properties of our protocol will now be briefly discussed (without formal proof). It has to be stressed that the following statements are based on the assumptions stated in section 3.2, and that all devices in the network, including the attacker, are computationally bounded.

- The CICADA-S protocol provides forward security. A node that leaves the network can not successfully read/modify/insert/delete data in the WBAN, since the group key s is always updated in case the topology of the network changes.
- Nodes that are not securely initialized, can not join the WBAN. Only nodes that share a symmetric key with the back-end server, can construct a valid secure JOIN-REQUEST message, which is needed to join the WBAN.

- Since the group key is transported in an encrypted format from the gateway to the nodes in the WBAN, it is practically not feasible for an eavesdropper to recover the key. Only an attacker that can break the encryption scheme used to protect data in the WBAN, is able to find the group key s.
- The CICADA-S protocol offers key confirmation, which is important for security and performance reasons. After receiving the new group key s, a node sends a key confirmation message to the gateway, to inform that the key was received well. This avoids certain Denial-of-Service attacks (e.g., blocking key update messages). Due to packet loss and bit errors, key confirmation is also an important and necessary property of network protocols for wireless media.
- A sensor that is a member of a WBAN can not join another WBAN at the same time. The second secure JOIN-REQUEST message sent by the sensor will be refused by the back-end server, because this device will detect that the sensor already belongs to another network.
- Nodes that are part of a particular WBAN, are not able to read, modify, insert or delete encrypted data in other WBANs without this being detected, since these other networks do not share the same group key s.
- Since the confidentiality and integrity of data traveling in the WBAN is cryptographically protected, a device that does not possess the group key will not succeed in decrypting the enciphered communication, nor successfully modifying/inserting/deleting data into the network without this being detected.
- Replay attacks are detected because of the use of the synchronized counter, that is shared between sensor and gateway.
- Location privacy has been taken into account during the design of the CICADA-S protocol. The communication between gateway and back-end server is assumed to be completely secured (end-to-end) and anonymized. Using the data in the WBAN to trace a patient is not possible, because it only contains local identifiers, and these are not unique across WBANs. Only in the first message of the join procedure, the exact identity of the sensor is exposed. It is however not used in the other key management messages. Neither is it possible to link other messages to the initial key management message of the join procedure (since the synchronized counter is encrypted). As a result, the data in the WBAN can not be used to trace patients.

6 Conclusion

Wireless Body Area Networks are an enabling technology for mobile health care. These systems reduce the enormous costs associated to patients in hospitals as monitoring can take place even at home in real-time and over a longer period. A critical factor in the acceptance of WBANs is the provision of appropriate security and privacy protection of the wireless communication medium. The data traveling between the sensors should be kept confidential and integrity protected. Certainly in the mobile monitoring scenario, this is of uttermost importance.

In this paper we have presented CICADA-S, a security enabled cross-layer multi-hop protocol for Wireless Body Area Networks. It is a secure extension of the CICADA protocol, and was designed within the scope of the IM3-project (Interactive Mobile Medical Monitoring), which focuses on the research and implementation of a wearable system for health monitoring. The CICADA-S protocol is the first integrated solution to cope with the threats of interactive mobile monitoring and the life cycle of the sensors. It combines key management and secure privacy preserving communication techniques. We have presented the main security properties of CICADA-S, and shown that the addition of security mechanisms to the CICADA-S protocol has low impact on the power consumption and throughput. The security mechanisms integrated in the protocol are simple, yet very effective. The CICADA-S protocol can be implemented on today's devices as it only requires low-cost and minimal hardware changes.

The authors strongly believe that adding sufficient security mechanisms to Wireless Body Area Networks will work as a trigger in the acceptance of this technology for health care purposes.

Acknowledgments. This work is partially funded by a research grant of the Katholieke Universiteit Leuven for D. Singelée, by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), by the Fund for Scientific Research — Flanders (F.W.O.-V.,Belgium) project G.0531.05 (FWO-BAN) and by the Flemish IBBT project IM3.

References

1. Takahashi, D., Xiao, Y., Hu, F.: LTRT: Least total-route temperature routing for embedded biomedical sensor networks. In: Proceedings of the 50th IEEE Global Telecommunications Conference, GLOBECOM 2007 (November 2007)
2. Moh, M., Culpepper, B.J., Lan, D., Teng-Sheng, M., Hamada, T., Ching-Fong, S.: On data gathering protocols for in-body biomedical sensor networks. In: Proceedings of the 48th IEEE Global Telecommunications Conference, GLOBECOM 2005 (November/December 2005)
3. Ruzzelli, A.G., Jurdak, R., O'Hare, G.M.P., Van Der Stok, P.: Energy-efficient multi-hop medical sensor networking. In: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, HealthNet 2007, New York, NY, USA, pp. 37–42 (2007)
4. Latré, B., Braem, B., Moerman, I., Blondia, C., Reusens, E., Joseph, W., Demeester, P.: A low-delay protocol for multihop wireless body area networks. In: Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, Philadelphia, PA, USA (August 2007)
5. Otto, C., Milenkovic, A., Sanders, C., Jovanov, E.: System architecture of a wireless body area sensor network for ubiquitous health monitoring. Journal of Mobile Multimedia 1(4), 307–326 (2006)
6. IBBT IM3-project, http://projects.ibbt.be/im3

7. Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. Communications of the ACM 47(6), 53–57 (2004)
8. Zhou, L., Haas, Z.J.: Securing ad hoc networks. IEEE Network 13(6), 24–30 (1999)
9. Eschenauer, L., Gligor, V.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, pp. 41–47 (November 2002)
10. Perrig, A., Szewczyk, R., Wen, V., Culler, D.E., Tygar, J.D.: SPINS: Security protocols for sensor networks. In: Mobile Computing and Networking, pp. 189–199 (2001)
11. Baldus, H., Klabunde, K., Msch, G.: Reliable set-up of medical body-sensor networks. In: Karl, H., Wolisz, A., Willig, A. (eds.) EWSN 2004. LNCS, vol. 2920, pp. 353–363. Springer, Heidelberg (2004)
12. Poon, C.C.Y., Yuan-Ting, Z., Shu-Di, B.: A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. IEEE Communications Magazine 44(4), 73–81 (2006)
13. Falck, T., Baldus, H., Espina, J., Klabunde, K.: Plug 'n play simplicity for wireless medical body sensors. Mobile Networks and Applications 12(2-3), 143–153 (2007)
14. Gehrmann, C., Mitchell, C., Nyberg, K.: Manual authentication for wireless devices. RSA Cryptobytes 7(1), 29–37 (2004)
15. Singelée, D., Preneel, B.: Key establishment using secure distance bounding protocols. In: Proceedings of the first Workshop on the Security and Privacy of Emerging Ubiquitous Communication Systems, SPEUCS 2007, Philadelphia, PA, USA, August 2007. IEEE Computer Society Press, Los Alamitos (2007)
16. Stajano, F., Anderson, R.: The resurrecting duckling: Security issues in ad–hoc wireless networks. In: Malcolm, J.A., Christianson, B., Crispo, B., Roe, M. (eds.) Security Protocols 1999. LNCS, vol. 1796, pp. 172–182. Springer, Heidelberg (2000)
17. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
18. ISO/IEC 9798-2. Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms (1999)
19. Daemen, J., Rijmen, V.: The Design of Rijndael – AES – The Advanced Encryption Standard. Springer, Heidelberg (2002)
20. NIST Special Publication 800-38C. Recommendation for block cipher modes of operation – the CCM mode for authentication and confidentiality. U.S. DoC/NIST (May 2004), http://csrc.nist.gov/publications/
21. NIST Special Publication 800-38D. Recommendation for block cipher modes of operation – galois/counter mode (GCM) and GMAC. U.S. DoC/NIST (November 2007), http://csrc.nist.gov/publications/

Communication in Random Geometric Radio Networks with Positively Correlated Random Faults

Evangelos Kranakis[1], Michel Paquette[1], and Andrzej Pelc[2]

[1] School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada
kranakis@scs.carleton.ca, michel.paquette@polymtl.ca
[2] Département d'informatique et d'ingénierie, Université du Québec en Outaouais Gatineau, Québec, J8X 3X7, Canada
pelc@uqo.ca

Abstract. We study the feasibility and time of communication in random geometric radio networks, where nodes fail randomly with positive correlation. We consider a set of radio stations with the same communication range, distributed in a random uniform way on a unit square region. In order to capture fault dependencies, we introduce the *ranged spot* model in which damaging events, called *spots*, occur randomly and *independently* on the region, causing faults in all nodes located within distance s from them. Node faults within distance $2\mathsf{s}$ become dependent in this model and are positively correlated. We investigate the impact of the spot arrival rate on the feasibility and the time of communication in the fault-free part of the network. We provide an algorithm which broadcasts correctly with probability $1 -$ in faulty random geometric radio networks of diameter D in time $\mathsf{O}(\mathsf{D} + \log 1/\ )$.

Keywords: Fault-tolerance, dependent faults, broadcast, crash faults, random, geometric radio network.

1 Introduction

Wireless networks have received much attention in recent years because of applications where wired networks are impractical or impossible to deploy. These networks are now so common that the idea of large scale wireless networks has become natural. However, as they grow in size, complexity, and area, wireless networks become increasingly vulnerable to component failures and damaging environmental phenomena. Nodes of a network may fail and the communication medium may become too noisy to support correct message transmissions. These failures often result in delaying, blocking, or even distorting transmitted messages. Hence, it becomes important that the desired tasks may be accomplished efficiently in spite of these faults, usually without knowing their location ahead of time. Networks with this property are called fault-tolerant.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 108–121, 2008.
© Springer-Verlag Berlin Heidelberg 2008

An important type of wireless network is obtained from a set of stations in the plane where each station u has communication range r_u. The resulting network is modeled as a directed graph in which stations are nodes and a directed edge from u to v exists if v is at distance at most r_u from u. Such networks are called geometric radio networks (GRN).

One of the most important communication tasks is broadcasting. In this process, a source node attempts to transmit a message to all other nodes of the network. This process is successful if, upon termination, all functional nodes, connected to the source by a fault-free path, have received the source message. Although the question of fault-tolerant broadcasting has been widely studied for faulty point-to-point networks, few results are known about this process in geometric radio networks. To the best of our knowledge, all existing analytic results examine the general problem of broadcasting in networks where the number of faults is bounded above (cf., e.g., [8]), or faults are distributed randomly and independently (cf., e.g., [9,14]). Hence, the present paper is the first to address the problem of broadcasting in GRNs in the presence of positively correlated faults.

1.1 Model and Problem Definitions

We seek to model a network composed of mobile stations moving under the *Random Waypoint* mobility model [7] inside an open region, e.g., a train station or a plaza. Under this mobility model, mobile stations alternately move and pause for random amounts of time, choosing a direction, distance and speed at random at every movement phase. Here, we assume that the mobile stations move at low, pedestrian-like speeds, making the network appear static for the short duration of communication processes; the distance and directions are chosen in some uniform way. We further assume that neither the boundaries of the open region nor the other mobile stations have any effect on the mobile station movements. Hence, any snapshot of the graph is a set of stations distributed on a plane by a Poisson process. Due to the short duration of the communication processes, we consider that the faults are permanent. The proposed static model also applies to networks of sensors spread randomly in hostile environments where individual placement and replacement of units is not possible.

We focus attention on a unit square region of the plane. Node locations occur with Poisson arrival rate n. We fix a parameter r, called the communication range. Any two nodes at Euclidean distance at most r from one another can communicate directly. We now define the *ranged spot* fault model. Damaging phenomena, called *spots*, occur on the plane with Poisson arrival rate λ. Some examples of damaging phenomena are lightning strikes (and other electrostatic discharges), electromagnetic pulses and explosions. We fix a parameter $s > 0$, called the *spot range*. Each spot causes permanent crash faults in all nodes within distance s of it, i.e., inside the disk of radius s centered at it, which we call the *spot disk*. For a fixed spot i, we denote the corresponding spot disk by D_i. Faulty nodes can neither send nor receive messages for the entire communication process. More formally, consider the undirected fault-free graph $G(V, E)$, where

V is the random set of nodes occurring on the unit square with Poisson arrival rate n, and E is the set of all node pairs $\{u, v\}$ for which the Euclidean distance is at most r. Let S be the set of spots which occur on the unit square with Poisson arrival rate λ. Let F be the set of faulty nodes, i.e., all the nodes in V whose location is within distance s from at least one spot in S. We consider the graph $G[V']$ induced on G by the set $V' = V \setminus F$ of all functional nodes. To remind the reader how it is built, throughout this paper, we will denote the graph $G[V']$ by $U(n, r, \lambda, s)$.

As usual in wireless network algorithms, communication in $U(n, r, \lambda, s)$ is assumed synchronous; nodes have synchronized clocks and the communication process is executed in fixed time steps, called *rounds*. All communication is done using the same base frequency, modulation and encoding, hence using a single channel. In each round, each node either sends a message or listens to the channel. In the first case, we say that the node is a sender, otherwise, it is a receiver. In a fixed round t, a node v receives a message if and only if it is a receiver and precisely one of its neighbors is a sender. If no neighbor of v is a sender, then there is no message on the channel which v can receive. If more than one neighbor of v sends a message, we say that a *collision* occurs at v and v can only perceive noise on the channel. Nodes do not have collision detection abilities, i.e., they cannot distinguish collision noise from background noise (which is apparent when no messages are heard).

We say that an event occurs in the graph with high probability (w.h.p.) if its probability converges to 1 as the node arrival rate n grows to infinity. We say that an event occurs on the graph with constant (positive) probability if its probability p is bounded away from 0 and from 1 as n grows to infinity, i.e., if there exist positive constants ϵ_1, ϵ_2 such that $0 < \epsilon_1 < p < \epsilon_2 < 1$ for all n. Specifically, we say that a graph is connected w.h.p. when the event that it is connected occurs w.h.p. On the other hand, we say that a graph is not connected w.h.p. when the event that it is disconnected occurs at least with constant probability.

In this paper, we study the question of feasibility and efficiency of communication in the fault-free graph $U(n, r, \lambda, s)$.

1.2 Our Results

We first give answers to the question for which parameters $s = s(n)$ and $\lambda = \lambda(n, s)$ there exist any fault-free nodes in the unit square, i.e., when the fault-free graph $U(n, r, \lambda, s)$ is non-empty, w.h.p. For $s \in o(1)$, we find a threshold function $l(n, s)$ and constants L_1, L_2 such that, for $\lambda \geq L_1 \cdot l(n, s)$ fault-free nodes do not exist, w.h.p., while for $\lambda \leq L_2 \cdot l(n, s)$ they do exist w.h.p. For $s \in \Omega(1)$, we show that, for $\lambda \in \omega(1/s^2)$ fault-free nodes do not exist, w.h.p., and for $\lambda \in o(1/s^2)$ they do exist, w.h.p.

We then give answers to the question for which parameters $s = s(n)$, $r = r(n)$ and $\lambda = \lambda(n, s, r)$, the fault-free graph $U(n, r, \lambda, s)$ is connected, w.h.p. Connectivity is equivalent to feasibility of communication in our setting. We restrict attention to the case of small spot range, more precisely, we work under the

assumption $s \in o(r)$. In the case $r \in o(1)$, we find a threshold function $c(n, s, r)$ and constants C_1, C_2 such that, for $\lambda \geq C_1 \cdot c(n, s, r)$ the graph $U(n, r, \lambda, s)$ is not connected w.h.p., and for $\lambda \leq C_2 \cdot c(n, s, r)$ it is connected, w.h.p. Then, in the case $r \in \Omega(1)$, and for $\lambda \in o(1/s^2)$, we show that for the values of λ for which the graph $U(n, r, \lambda, s)$ contains at least one node w.h.p., it is also connected w.h.p.

Finally, under the additional restriction on spot range, when $s \in o(1/\sqrt{n})$, we show an algorithm which accomplishes broadcast with probability at least $1 - \epsilon$ in time $O(D + \log 1/\epsilon)$ in the graph $U(n, r, \lambda, s)$ of diameter D.

Due to lack of space, the proofs of several lemmas and theorems are omitted.

1.3 Related Work

The fundamental questions of network reliability have received much attention in the context of point-to-point networks, under the assumption that components fail randomly and independently (cf., e.g. [1,2,3,11] and the survey [12]). On the other hand, empirical work has shown that positive correlation of faults is a more reasonable assumption for networks [6,15,16]. In particular, in [16], the authors provide empirical evidence that data packets losses are spatially correlated in networks. Moreover, in [6], the authors simulate failures in a sensor network using a model similar to that of the present paper; according to these authors, the environment provides many spatially correlated phenomena resulting in such fault patterns. More recently, in [10], a gap was demonstrated between the fault-tolerance of networks when faults occur independently as opposed to when they occur with positive correlation. To the best of our knowledge, this was the first paper to provide analytic results concerning network fault-tolerant communication in the presence of positively correlated faults.

In contrast, few results are known about fault-tolerant communication in geometric radio networks. To the best of our knowledge, all existing analytic results examine the problem of broadcasting in networks where, either the number of faults is bounded above (cf., e.g., [8]), or faults occur randomly and independently (cf., e.g., [9,14]). In particular, in [14], the authors consider the problem of connectivity of a square grid of n sensors with communication range r on a unit square when faults occur at the nodes randomly and independently with probability $1-p$. They show that if $pr^2 \approx \frac{\log n}{n}$, then the functional nodes are all part of a connected component w.h.p. In [8], the authors consider the problem of broadcasting in a fault-free connected component of a radio network whose nodes are located at grid points of square grids and can communicate within a square of size r. For an upper bound t on the number of faulty nodes, in worst-case location, the authors propose a $\Theta(D+t)$-time oblivious broadcast algorithm and a $\Theta(D + \log(\min(r, t)))$-time adaptive broadcast algorithm, both operating on a connected fault-free component of diameter D.

The question of communication in networks of unknown topology has been widely studied in recent years. In fact, in [4], the authors state that broadcasting algorithms which function in unknown GRNs also function in the resulting fault-free connected components of faulty GRNs. A basic performance evaluation criterion of broadcasting algorithms is the time necessary for the algorithm

to terminate; in synchronous networks, this time is measured as the number of communication rounds. For networks whose fault-free part has a diameter D, $\Omega(D)$ is a trivial lower bound on broadcast time, but optimal running time is a function of the information available to the algorithms (cf., e.g., [5]). For instance, in [5], an algorithm was obtained which accomplishes broadcast in arbitrary GRNs in time $O(D)$ under the assumption that nodes have a large amount of knowledge about the network, i.e. given that all nodes have a *knowledge radius* larger than R, the largest communication radius. The authors also show that algorithms broadcasting in time $O(D + \log n)$ are asymptotically optimal, for unknown GRNs when nodes can communicate spontaneously (before receiving the source message) and either can detect collisions or have knowledge of node locations at some positive distance δ, arbitrarily small. In the present paper, we assume that nodes communicate spontaneously, but know nothing of the network, other than their own location, and cannot detect collisions. Under these assumptions, we show an $O(D + \log 1/\epsilon)$-time algorithm which correctly broadcasts in the random graph $U(n, r, \lambda, s)$ with probability at least $1 - \epsilon$.

2 Liveness of the Graph

In this section, we show bounds on the spot arrival rate λ for which functional nodes exist in the unit square, i.e., the graph $U(n, r, \lambda, s)$ contains at least one node, w.h.p. We say that the graph $U(n, r, \lambda, s)$ is *alive* if it contains at least one node; otherwise, we say that it is *dead*.

Theorem 1. *For $s \in o(1)$, there exist two positive constants, L_1 and L_2, such that if the spot arrival rate $\lambda \geq \frac{L_1 \ln(\min\{n, 1/s^2\})}{s^2}$, then the graph $U(n, r, \lambda, s)$ is dead, w.h.p., and if $\lambda \leq \frac{L_2 \ln(\min\{n, 1/s^2\})}{s^2}$, then $U(n, r, \lambda, s)$ is alive, w.h.p.*

Theorem 2. *For $s \in \Omega(1)$, the graph $U(n, r, \lambda, s)$ is dead w.h.p. if $\lambda \in \omega(1/s^2)$ and alive w.h.p. if $\lambda \in o(1/s^2)$.*

Remark 1. For $s \in o(1/\sqrt{n})$ and $\lambda = \frac{\ln(cn)}{\pi s^2}$, where c is a positive constant, the graph $U(n, r, \lambda, s)$ is dead with constant probability.

3 Connectivity of U (n, r, , s)

In the preceding section, we gave a threshold for the spot arrival rate for which the graph $U(n, r, \lambda, s)$ is non-empty w.h.p. We now answer the next natural question: for which spot arrival rate is the graph $U(n, r, \lambda, s)$ connected w.h.p.?

It has been shown, in [13], that for any real number c, if $r \geq \sqrt{\frac{\ln n + c}{\pi n}}$, then the probability that the graph $U(n, r, \lambda, s)$, with $\lambda = 0$, is connected is at least $e^{-e^{-c}}$, as $n \to \infty$. If we substitute $e^{-c} = f(n)$, assume that $f(n) \in o(1)$ and recall that $e^{-f(n)} = 1 - f(n) + f(n)^2/2 - \ldots \geq 1 - f(n)$, then we see that if

$$r \geq \sqrt{\frac{\ln n + \ln 1/f(n)}{\pi n}}$$

then

$$\Pr[U(n,r,0,s) \text{ is connected}] \geq 1 - f(n).$$

Hence, it is natural to investigate the connectivity of the graph $U(n,r,\lambda,s)$ under the assumption $r^2 \geq \frac{\ln n + \ln 1/f(n)}{\pi n}$, for some $f(n) \in o(1)$, when we know that connectivity is guaranteed w.h.p. without faults. In what follows we make this assumption.

The main results of this section are Theorems 3 and 4. In Theorem 3, for spot range s of lower order of magnitude than the communication range r and for $r \in o(1)$, we show a threshold for the spot arrival rate λ below which the graph $U(n,r,\lambda,s)$ is connected w.h.p. and above which it is not. For the case $r \in \Omega(1)$, the separation is different: in Theorem 4, we show thresholds for the spot arrival rate λ below which the graph $U(n,r,\lambda,s)$ is connected w.h.p. and above which it is *dead* w.h.p.

Theorem 3. *For $s \in o(r)$ and $r \in o(1)$, there exist two positive constants, C_1 and C_2, such that if spots appear with arrival rate $\lambda \geq C_1 \ln\left(\frac{r^2 \min\{n,1/s^2\}}{\ln(1/r^2)}\right)/s^2$, then the graph $U(n,r,\lambda,s)$ is not connected w.h.p., and if the spot arrival rate $\lambda \leq C_2 \ln\left(\frac{r^2 \min\{n,1/s^2\}}{\ln(1/r^2)}\right)/s^2$, then the graph $U(n,r,\lambda,s)$ is connected, w.h.p.*

Theorem 3 will follow from Lemmas 2, 3, 4, and 5.

Theorem 4. *For $s \in o(r)$ and $r \in \Omega(1)$,*

1. *if $s \in o(1)$, then there exist two positive constants, C_3 and C_4, such that*
   (a) *for $\lambda \leq C_3 \ln(\min\{n, 1/s^2\})/s^2$, $U(n,r,\lambda,s)$ is connected, w.h.p.,*
   (b) *for $\lambda \geq C_4 \ln(\min\{n, 1/s^2\})/s^2$, the graph $U(n,r,\lambda,s)$ is dead w.h.p.,*
2. *if $s \in \Omega(1)$, then*
   (a) *for $\lambda \in o(1/s^2)$, the graph $U(n,r,\lambda,s)$ is connected, w.h.p.*
   (b) *for $\lambda \in \omega(1/s^2)$, the graph $U(n,r,\lambda,s)$ is dead w.h.p.,*

Theorem 4 will follow from Theorems 1 and 2 and from Lemmas 6 and 7.

3.1 Non-connectivity Results

In this section, we show conditions on spot arrival rate implying, w.h.p., non-connectivity of the graph $U(n,r,\lambda,s)$ by the existence of two functional nodes which cannot communicate with one another in the unit square.

Denote by $\mathcal{P}_{left}$ and $\mathcal{P}_{right}$ the two rectangular halves of the unit square. Partition $\mathcal{P}_{left}$ and $\mathcal{P}_{right}$ respectively into meshes of $r \times r$ squares. Group these squares in matrices of 5×5 squares, called blocks; let $\mathcal{B}_{left}$ and $\mathcal{B}_{right}$ be the sets of these blocks. For each block b, denote by c_b the central square in this block and by p_b the union of 8 squares adjacent to c_b. Let $alive_b$ be the event that c_b contains at least one functional node. Let $surround_b$ be the event that p_b contains no functional node. Let $isolation_b$ be the intersection of events $surround_b$ and $alive_b$. If $isolation_b$ occurs, and there is at least one functional node outside b, then nodes in c_b have no functional path to this external functional node,

and then, the graph $U(n, r, \lambda, s)$ is disconnected. In particular, we show non-connectivity w.h.p. by proving that events $isolation_{b_1}$ and $isolation_{b_2}$, $b_1 \in \mathcal{B}_{left}$ and $b_2 \in \mathcal{B}_{right}$, occur w.h.p. Note that, for distinct blocks b_1 and b_2, events $surround_{b_1}$ and $surround_{b_2}$ are independent.

We first examine non-connectivity in the case when $r \in o(1)$ and $s \in o(1/\sqrt{n})$, in Lemma 2. Non-connectivity for $r \in o(1)$ and for larger values of $s \in o(r)$ will be addressed in Lemma 3. The case $s \in \Omega(1)$ is treated in the next section. We show that for these values of r, the graph is connected w.h.p. for those spot arrival rates for which it is alive w.h.p.

Let F_v be the event that a fixed node v is faulty, i.e., that there exists at least one spot within distance s of it. Then, for spot arrival rate λ we have

$$\Pr[F_v] = 1 - e^{-\lambda \pi s^2}.$$

While distant faults are independent, the presence of a faulty node within distance $2s$ from a fixed node v implies that there is a spot which might be close enough to v to make it faulty, i.e., the occurrence of a fault at a node can never decrease the probability of a fault on another node. This is why faults are positively correlated. Hence, the following fact applies to the events F_v.

Fact 1. *For any set Z of nodes,*

$$\Pr[\bigcap_{v \in Z} F_v] \geq \prod_{v \in Z} \Pr[F_v].$$

A set S of nodes whose elements have a distance greater than $2s$ from one another is called *sparse*. Such a set has the property that the events F_v, for $v \in S$, are independent. The following lemma states that there exist large sparse sets, w.h.p.

Lemma 1. *A square A with area $|A|$ contains a sparse set S of size at least $k|A| \min\{n, 1/s^2\}$, for some positive constant k, w.h.p., if $|A| \min\{n, 1/s^2\} \to \infty$ as $n \to \infty$.*

Lemma 2. *Fix any constants $\alpha > 8$ and $\beta > 1$. For $s \in o(1/\sqrt{n})$ and $r \in o(1)$, the graph $U(n, r, \lambda, s)$ is not connected w.h.p. when $\lambda = \beta \ln\left(\frac{\alpha n r^2}{\ln(1/r^2)}\right)/\pi s^2$.*

Proof. Consider $f(n) \in \omega(1)$ and the set Λ of spot arrival rates of the form $\lambda = \ln\left(\frac{\alpha r^2 n}{\ln(1/(r^2 f(n)))}\right)/(\pi s^2)$. Consider two subsets of Λ: Λ_1 consisting of these λ of the form $\lambda = \ln(g(n) r^2 n)/(\pi s^2)$, with $g(n) \in O(1)$ and Λ_2 consisting of these λ of the same form with $g(n) \in \Omega(1)$. In each case, we show that there exists at least one occurrence of the event $isolation_b$ in each set $\mathcal{B}_{left}$ and $\mathcal{B}_{right}$ and thus, that the graph $U(n, r, \lambda, s)$ is disconnected.

Case 1: $\lambda = \ln(g(n) r^2 n)/(\pi s^2)$, with $g(n) \in O(1)$

Fix a block b and consider the event $alive_b$. Consider the subsquare $c'_b \subset c_b$ whose points are at distance greater than $2s$ from p_b, i.e. for which the contained

nodes become faulty independently from nodes in p_b. For $s/r \to 0$ as $n \to \infty$, $|c'_b| > 0.9r^2$, for large n. From Lemma 1, since $0.9nr^2 > 0.9 \log n \in \omega(1)$, it follows that, w.h.p., there is a sparse set of nodes S_b, in c'_b, of size at least knr^2, for some positive constant k. Events F_v, $v \in S_b$, occur independently. Let A be the event that the above lower bound on the size of the sparse set S_b holds. Assume A. Then,

$$\begin{aligned}
\Pr[alive_b] &= 1 - \Pr[\forall v \in c_b \ F_v] \geq 1 - \Pr[\forall v \in c'_b \ F_v] \\
&\geq 1 - \Pr[\forall v \in S_b \ F_v] \geq 1 - (\Pr[F_v])^{knr^2} = 1 - (1 - e^{-\lambda \pi s^2})^{knr^2} \\
&= 1 - (1 - e^{-\ln(g(n)r^2 n)})^{knr^2} \\
&= 1 - (1 - 1/g(n)r^2 n)^{knr^2} \geq c' \in \Theta(1)
\end{aligned}$$

since $g(n) \in O(1)$. Since $\Pr[A] \to 1$ for large n, this implies that the probability of event $alive_b$ is at least a positive constant c. Let $\mathcal{A}_{left}$ be the set of all blocks b in $\mathcal{B}_{left}$ for which the event $alive_b$ occurs. Since the probability of the event $alive_b$ is a constant, the expected size of the set $\mathcal{A}_{left}$ is a constant fraction of $|\mathcal{B}_{left}|$. The number of blocks in $\mathcal{B}_{left}$ is $|\mathcal{B}_{left}| = 1/50r^2$. For $r \in o(1)$, $|\mathcal{B}_{left}|$ grows to infinity as $n \to \infty$ and thus, under the preceding assumptions, we use Chernoff bounds to show that $|\mathcal{A}_{left}| \geq (0.9)c/(50r^2)$ w.h.p. Assume this bound on $|\mathcal{A}_{left}|$ and let $k/r^2 = (0.9)c/(50r^2)$ for the remainder of the proof.

Fix a block b and consider the event $surround_b$. Using Chernoff Bounds adapted to Poisson distributions, we can show that, w.h.p., at most αnr^2 nodes are in p_b; let E be the event that this bound holds. Assume E. Then, we have, by Fact 1,

$$\Pr[surround_b] = \Pr[\bigcap_{v \in p_b} F_v] \geq \prod_{v \in p_b} \Pr[F_v] \geq (1 - e^{-\lambda \pi s^2})^{\alpha nr^2}$$

and since $\Pr[E] \to 1$ for large n, we have $\Pr[surround_b] \geq (0.9)(1 - e^{-\lambda \pi s^2})^{\alpha nr^2}$, for large n. Then, the probability that there exists a block $b \in \mathcal{B}_{left}$ for which event $isolation_b$ occurs is

$$\begin{aligned}
\Pr[\exists b \in \mathcal{B}_{left} \ isolation_b] &= \Pr[\exists b \in \mathcal{A}_{left} \ surround_b] \\
&= 1 - \Pr[\forall b \in \mathcal{A}_{left} \neg surround_b] \\
&= 1 - (\Pr[\neg surround_b])^{|\mathcal{A}_{left}|} \\
&\geq 1 - \left(1 - (0.9)\left(1 - \frac{\ln(1/(r^2 f(n)))}{\alpha r^2 n}\right)^{\alpha r^2 n}\right)^{k/r^2} \\
&= 1 - (1 - (0.9)r^2 f(n))^{k/r^2} \to 1 \text{ as } n \to \infty.
\end{aligned}$$

The same calculations apply to the second half of the unit square, thus showing the occurrence of at least 2 events $isolation_b$ w.h.p. This concludes the argument in the first case.

Case 2: $\lambda = \ln(g(n)r^2n)/(\pi s^2)$, with $g(n) \in \Omega(1)$

Consider again the event $surround_b$. For $\lambda = \ln(g(n)r^2n)/(\pi s^2)$, with $g(n) \in \Omega(1)$, the same argument as in case 1 implies

$$\begin{aligned}\Pr[surround_b] &\geq (0.9)(1-e^{-\lambda\pi s^2})^{\alpha nr^2} = (0.9)(1-1/(g(n)r^2n))^{\alpha nr^2} \\ &\geq c' \in \Theta(1).\end{aligned}$$

Let $\mathcal{S}_{left}$ be the set of all blocks in $\mathcal{B}_{left}$ for which the event $surround_b$ occurs. Since the probability of $surround_b$ is constant, the expected size of the set $\mathcal{S}_{left}$ is a constant fraction of $|\mathcal{B}_{left}|$. The number of blocks in $\mathcal{B}_{left}$ is $|\mathcal{B}_{left}| = 1/50r^2$. For $r \in o(1)$, $|\mathcal{B}_{left}|$ grows to infinity as $n \to \infty$ and thus, under the preceding assumptions, we use Chernoff bounds to show that $|\mathcal{S}_{left}| \geq (0.9)c'/(50r^2)$ w.h.p. Assume this bound on $|\mathcal{S}_{left}|$ and let $k/r^2 = (0.9)c'/(50r^2)$ for the remainder of the proof.

From Remark 1, if the spot arrival rate is $\lambda = \ln(nh(n))/(\pi s^2)$, $h(n) \in \Omega(1)$, we find a positive constant probability that the graph $U(n, r, \lambda, s)$ is dead. Hence, consider the subset of spot arrival rates of the form $\lambda = \ln(nh(n))/(\pi s^2)$, $h(n) \in o(1)$. Then, for these values of λ, the probability that there exists a block $b \in \mathcal{B}_{left}$ for which event $isolation_b$ occurs is

$$\begin{aligned}\Pr[\exists b \in \mathcal{B}_{left}\ isolation_b] &= \Pr[\exists b \in \mathcal{S}_{left}\ alive_b] = 1 - \Pr[\forall b \in \mathcal{S}_{left} \neg alive_b] \\ &= 1 - (\Pr[\neg alive_b])^{|\mathcal{S}_{left}|} \\ &\geq 1 - (1-(1-(1-e^{-\lambda\pi s^2})^{k'nr^2}))^{k/r^2} \\ &= 1 - ((1-e^{-\ln(nh(n))})^{k'nr^2})^{k/r^2} \\ &= 1 - (1-1/(nh(n)))^{k'kn} \to 1 \text{ as } n \to \infty.\end{aligned}$$

The same calculations apply to the second half of the unit square, thus showing the occurrence of at least 2 events $isolation_b$ w.h.p. This concludes the argument in the second case.

To conclude the proof, fix the function $f(n) = 1/r$. Since $r \in o(1)$, we have $f(n) \in \omega(1)$. Hence the corresponding $\tilde{\lambda} = \ln\left(\frac{\alpha r^2 n}{\ln(1/(r^2 f(n)))}\right)/(\pi s^2) = \ln\left(\frac{\alpha r^2 n}{\ln(1/r)}\right)/(\pi s^2)$ is in Λ. We show that $\tilde{\lambda} < \beta \ln\left(\frac{\alpha r^2 n}{\ln(1/r^2)}\right)/(\pi s^2)$, for any constant $\beta > 1$. Indeed,

$$\begin{aligned}\tilde{\lambda} &= \ln\left(\frac{\alpha r^2 n}{\ln(1/r)}\right)/(\pi s^2) = \ln\left(\frac{\alpha r^2 n}{0.5\ln(1/r^2)}\right)/(\pi s^2) \\ &= \left(\ln\left(\frac{\alpha r^2 n}{\ln(1/r^2)}\right) + \ln 2\right)/(\pi s^2) \leq \beta \ln\left(\frac{\alpha r^2 n}{\ln(1/r^2)}\right)/(\pi s^2).\end{aligned}$$

It follows that all $\lambda = \beta \ln\left(\frac{\alpha r^2 n}{\ln(1/r^2)}\right)/(\pi s^2)$, for any constant $\beta > 1$, are also in Λ which proves the lemma. Note that, under the assumption $s \in o(1/\sqrt{n})$), we have $\min\{n, 1/s^2\} = n$. ∎

The proof of Lemma 3 differs from the proof of Lemma 2 only in the use of a partition to obtain the probability of the event $surround_b$.

Lemma 3. *Fix any constant $\beta > 1$. For $s \in o(r)$ and $r \in o(1)$, the graph $U(n, r, \lambda, s)$ is disconnected w.h.p. when $\lambda = 4\beta \ln\left(\frac{8r^2/s^2}{\ln(1/r^2)}\right)/\pi s^2$.*

The preceding lemmas concern only the case when $r \in o(1)$. As stated in Theorem 4, for $r \in \Omega(1)$, thresholds on spot arrival rate separate the case of connected $U(n, r, \lambda, s)$ from the case when it is dead. Hence, we do not provide any non-connectivity result for $r \in \Omega(1)$ and defer this case to the next section.

3.2 Connectivity Results

In this section, we show conditions on spot arrival rate guaranteeing connectivity of the graph $U(n, r, \lambda, s)$ w.h.p. We show connectivity of $U(n, r, \lambda, s)$ by proving the existence of a fault-free node in each square of a sufficiently fine partition of the unit square w.h.p. This implies the existence of a fault-free path between any pair of nodes of the graph $U(n, r, \lambda, s)$ and hence this graph is connected.

Partition the unit square into a mesh of $r/\sqrt{5} \times r/\sqrt{5}$ squares, called blocks. Let $\mathcal{B}$ be the set of all blocks. The distance between any two points in blocks which are adjacent by an edge (edge-adjacent) is at most r. Hence, functional nodes in adjacent blocks can communicate with each other.

Partition each block $b \in \mathcal{B}$ into a mesh of $3s \times 3s$ squares called tiles. Let T_b be the set of all these $r^2/(45s^2)$ tiles for the block b. For a fixed tile $t \in T_b$, let $free_t$ be the event that it contains no spot. Under the event $free_t$, the central $s \times s$ square $c_t \subset t$ is at distance greater than s from all spots. Let a_t be the event that c_t contains at least one node. Since node arrivals and spot arrivals are independent, the events $free_t$ and a_t are independent. Moreover, for all $t \neq s \in T_b$, the events a_t, a_s $(free_t, free_s)$ are independent since they are respectively the result of arrivals inside non-overlapping tiles t and s.

Consider the event $alive_b$ that a fixed block b contains at least one functional node. The event $alive_b$ is implied by the existence of a tile $t \in T_b$ where both the events $free_t$ and a_t occur. Let $alive'_b = \{\exists t \in T_b \text{ s.t. } free_t \cap a_t\}$ be this sub-event of $alive_b$. Hence, the probability of event $alive_b$ that a fixed block b contains at least one functional node is

$$\begin{aligned}\Pr[alive_b] \geq \Pr[alive'_b] &= \Pr[\exists t \in T_b\ free_t \cap a_t] = 1 - \Pr[\forall t \in T_b\ \neg free_t \cup \neg a_t] \\ &= 1 - (\Pr[\neg free_t \cup \neg a_t])^{|T_b|} = 1 - (1 - \Pr[free_t \cap a_t])^{|T_b|} \\ &= 1 - (1 - \Pr[free_t]\Pr[a_t])^{|T_b|} = 1 - (1 - e^{-\lambda 9s^2}(1 - e^{-ns^2}))^{r^2/(45s^2)}.\end{aligned}$$

Let *connect* be the event that each block b in $\mathcal{B}$ contains at least one functional node. We have $\Pr[connect] \geq \Pr[\forall b \in \mathcal{B}\ alive'_b]$. The next two lemmas are easily derived from the above estimates.

Lemma 4. *For $s \in o(1/\sqrt{n})$ and any constant $\alpha < 1$, the graph $U(n, r, \lambda, s)$ is connected, w.h.p., when the spot arrival rate is $\lambda = \alpha \ln\left(\frac{nr^2}{45\ln(1/r^2)}\right)/9s^2$.*

Lemma 5. *For $s \in \Omega(1/\sqrt{n}) \cap o(r)$ and any constant $\alpha < 1$, $U(n, r, \lambda, s)$ is connected, w.h.p., when the spot arrival rate is $\lambda = \alpha \ln\left(\frac{r^2/s^2}{45\ln(1/r^2)}\right)/9s^2$.*

For large values of r, we show connectivity for the same range of λ for which we have shown the graph $U(n, r, \lambda, s)$ to be alive w.h.p.

Lemma 6. *For $r \in \Omega(1)$ and $s \in o(1)$, the graph $U(n, r, \lambda, s)$ is connected, w.h.p., when the spot arrival rate is $\lambda = \alpha \frac{\ln(\min\{n, 1/s^2\})}{\pi s^2}$, for any constant $\alpha < 1$.*

For $r \in \Omega(1)$ and $s \in \Omega(1) \cap o(r)$, we observe that if $r \in \Theta(1)$, then the condition $s \in o(r)$ is impossible. Hence, necessarily, $r \in \omega(1)$. Since the unit square has a diameter of $\sqrt{2}$, if it is alive, then it is also connected for $r \in \omega(1)$ and sufficiently large n. Hence Lemma 7 follows from Theorem 2.

Lemma 7. *For $r \in \omega(1)$ and $s \in \Omega(1) \cap o(r)$, the graph $U(n, r, \lambda, s)$ is connected, w.h.p., when the spot arrival rate is $\lambda \in o(1/s^2)$.*

4 Broadcasting Algorithm

We propose a deterministic algorithm which completes broadcast with probability $1 - \epsilon$ in time $O(D + \log 1/\epsilon)$, in the fault-free graph $U(n, r, \lambda, s)$ for $s \in o(1/\sqrt{n})$. The algorithm consists of two parts: a preprocessing part called *spokesman election*, and a message transmission part. In the spokesman election part a unique node, called the *spokesman*, is selected in each square of a partition defined below. Only the spokesman of a square relays messages in the following part.

Partition the unit square into a mesh of $r/\sqrt{5} \times r/\sqrt{5}$ squares called boxes and let S be the set of these boxes. Group the boxes in 5×5 matrices, called blocks and let B be the set of all these blocks. For all blocks, label its boxes 1 through 25, row by row. Further partition each box into a mesh of $1/\sqrt{n} \times 1/\sqrt{n}$ squares, called tiles. Let T_i be the set of all tiles in a box i. For all boxes, label the tiles 1 through $t = r^2 n/5$, row by row.

Algorithm $\mathcal{A}^*$

Spokesman Election Part

Nodes know their location and hence, they can compute the labels i, j of their box, and tile, respectively. Nodes label themselves (i, j) accordingly.

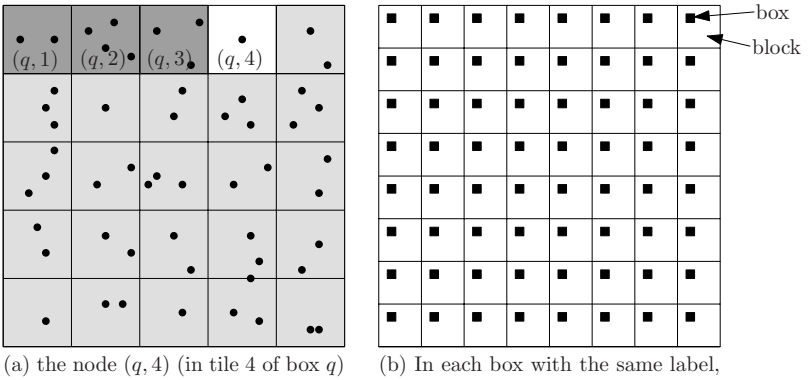
In parallel for all blocks, the algorithm executes rounds $i = 1, 2, \ldots, 25$. In a round i, the algorithm sequentially goes through steps $j = 1, 2, \ldots, t$. In a round i, at step j, all nodes with label (i, j) (in box i and tile j) transmit their label and the list of labels heard from adjacent boxes. At any given step j', when only one node transmits its label (i, j'), the message is heard by all other nodes in the box i and all edge-adjacent boxes; The first node whose message is heard is chosen as the spokesman for box i by all other nodes in the box i (the node itself does not know it yet) and in edge-adjacent boxes. In subsequent steps in round i, nodes in the box i containing this node (i, j') are silent. The node (i, j') will learn that it is the spokesman for the box i when, in an edge adjacent box, a unique node transmits its own label and the list of labels heard from adjacent boxes. Since all boxes, except box 25, are edge-adjacent to a box with a larger label, by the end of round 25, if a spokesman is chosen for each box, then all

spokesmen, with the exception of the spokesman in box 25 are confirmed, i.e., they know that they are spokesmen. Hence, after round 25, a single transmission from the spokesman in box 24 is sufficient to confirm the spokesman of box 25. This transmission is done in parallel by all spokesmen in boxes labeled 24, right after the end of round 25.

Hence, the spokesman election part chooses and confirms one spokesman in every box if there is, in every box, a tile which contains exactly one functional node.

Message Transmission Part

In the first step of this part, the source transmits its message. Then, in parallel for all blocks, the algorithm is executed in identical phases $\rho = 1, 2, \ldots$. In phase ρ, steps $j = 1, 2, \ldots, 25$ are executed sequentially. In a step j, a spokesman of box j which has received the source message but has not relayed it yet, transmits the message. This completes the description of algorithm $\mathcal{A}^*$. See Figure 1.



(a) the node $(q, 4)$ (in tile 4 of box q) is elected spokesman in box q

(b) In each box with the same label, spokesmen transmit the message in parallel, for all blocks

Fig. 1. Algorithm $\mathcal{A}^*$: (a) Spokesman Election part (b) Message Transmission part

Let ϵ be the tolerated error probability for the algorithm, i.e., we wish to broadcast with probability at least $1 - \epsilon$. Let $\mathcal{A}$ be the algorithm $\mathcal{A}^*$ modified so that the spokesman election part uses only the first $\frac{\ln(2D^2/\epsilon)}{\ln(1/(1-(0.9)e^{-(c+1)}))}$ tiles of each box.

Theorem 5. *Let c be a positive constant and $d = \ln(1/(1-(0.9)e^{-(c+1)}))$. For $s \in o(1/\sqrt{n})$, $r^2 \geq \frac{5\ln(5D^2/\epsilon)}{dn}$, and $\lambda \leq c/(\pi s^2)$, the algorithm $\mathcal{A}$ broadcasts a message in time $O(D + \log 1/\epsilon)$, with probability at least $1 - \epsilon$.*

Proof. Consider a tile t. There exists a subsquare a of t of area $(1/\sqrt{n} - s)^2 = 1/n - 2s/\sqrt{n} + s^2$ whose nodes are not affected by spots in other tiles; the remaining subset a' of the tile has area $2s/\sqrt{n} - s^2$. Let $good_t$ be the event that there exists exactly one node in a, no node in a', and that the node in a is not within distance s of a spot. We have

$$\begin{aligned}\Pr[good_t] &= e^{-(1/n-2s/\sqrt{n}+s^2)n}((1/n-2s/\sqrt{n}+s^2)n)\cdot e^{-(2s/\sqrt{n}-s^2)n}\cdot e^{-\lambda\pi s^2}\\ &\geq (1-2s\sqrt{n}+s^2n)e^{-1+(2s\sqrt{n}-s^2n)-(2s\sqrt{n}-s^2n)-\frac{c}{\pi s^2}\pi s^2} \geq 0.9e^{-(c+1)}\end{aligned}$$

for large n. Let $spokesman_q$ be the event that the spokesman election part is successful in a fixed box q. Since $r^2 \geq \frac{5\ln(5D^2/\epsilon)}{dn}$, there are at least $nr^2/5 = \frac{\ln(5D^2/\epsilon)}{d}$ tiles in each box. Hence, the algorithm $\mathcal{A}$ can execute its spokesman election part. Then, we have

$$\begin{aligned}\Pr[spokesman_q] &= 1-(1-\Pr[good_t])^{\ln(5D^2/\epsilon)/d}\\ &\geq 1-(1-0.9e^{-(c+1)})^{\ln(5D^2/\epsilon)/d}\\ &= 1-\left(\frac{5D^2}{\epsilon}\right)^{\ln(1-0.9e^{-(c+1)})/d}\\ &= 1-\left(\frac{5D^2}{\epsilon}\right)^{-\frac{\ln(1/(1-0.9e^{-(c+1)}))}{\ln(1/(1-0.9e^{-(c+1)}))}} = 1-\frac{\epsilon}{5D^2}.\end{aligned}$$

There are at most $5D^2$ boxes.Hence, the event $spokesmen$ that each box contains one spokesman occurs with probability

$$\Pr[spokesmen] \geq \left(1-\frac{\epsilon}{5D^2}\right)^{5D^2} \geq 1-\epsilon.$$

We now show that, assuming the event $spokesmen$, all functional nodes are informed and we estimate the total time of the algorithm. Consider the Message Transmission part. For each phase, 25 time steps are elapsed. We say that a box with label j is active if the algorithm step is j, i.e., when its spokesman may transmit. All boxes with the same label are located at distance at least $4r/\sqrt{5}$ from each other. Only spokesmen in active boxes (with the same label j, at a step j) transmit. Hence all nodes in boxes adjacent to active boxes will receive the message correctly at every time step when a spokesman transmits in this active box (due to large distances between boxes with the same label, there are no collisions in adjacent boxes). It follows that if a message is received by any box in a block i at time t, then there exists a positive constant δ such that at time $t+\delta$ all nodes in the block i will know the message. Moreover, at time $t+\delta$, the nodes in boxes outside the block i, but adjacent to the boxes in block i also have received the message. Consider two nodes in different blocks i and j such that there is a sequence of edge-adjacent blocks of length $k-1$ between them. If all nodes in block i have received the message by time t, it follows from the above that, at the time $t+k\delta$, the message will also be received by all nodes in block j. Since the unit square is partitioned in rows and columns of $\sqrt{5}/(5r)$ blocks, there is a sequence of, at most, $2\sqrt{5}/(5r)$ blocks between any two blocks i and j, so that consecutive blocks are edge-adjacent. Hence, the total broadcast time is at most $2\delta\sqrt{5}/(5r)$. Since the diameter of the graph is at least $1/r$, the message transmission part is completed in time $O(D)$. The spokesman election

part of the algorithm terminates in $O(\log(5D^2/\epsilon)) = O(1 + \log D + \log 1/\epsilon)$ time steps. Hence, the total execution time of the algorithm is $O(D + \log 1/\epsilon)$, and the algorithm is correct with probability at least $1 - \epsilon$. ■

Acknowledgements. Evangelos Kranakis and Michel Paquette were supported by MITACS and NSERC. Andrzej Pelc was supported by the Research Chair in Distributed Computing of the Université du Québec en Outaouais and NSERC.

References

1. Bienstock, D.: Broadcasting with random faults. Discr. Appl. Math. 20, 1–7 (1988)
2. Chlebus, B.S., Diks, K., Pelc, A.: Sparse networks supporting efficient reliable broadcasting. Nordic Journal of Computing 1, 332–345 (1994)
3. Chlebus, B.S., Diks, K., Pelc, A.: Reliable broadcasting in hypercubes with random link and node failures. Comb. Prob. and Computing 5, 337–350 (1996)
4. Clementi, A.E.F., Monti, A., Silvestri, R.: Round robin is optimal for fault-tolerant broadcasting on wireless networks. J. Par. Distrib. Comp. 64, 89–96 (2004)
5. Dessmark, A., Pelc, A.: Broadcasting in geometric radio networks. Journal of Discrete Algorithms 5, 187–201 (2007)
6. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review 5(4), 11–25 (2001)
7. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks, ch. 5, pp. 153–181. Kluwer Academic Publishers, Dordrecht (1996)
8. Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks. Journal of Algorithms 39, 47–67 (2001)
9. Kranakis, E., Krizanc, D., Urrutia, J.: Coverage and connectivity in networks with directional sensors. In: Danelutto, M., Vanneschi, M., Laforenza, D. (eds.) Euro-Par 2004. LNCS, vol. 3149, pp. 917–924. Springer, Heidelberg (2004)
10. Kranakis, E., Paquette, M., Pelc, A.: Communication in networks with random dependent faults. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 418–429. Springer, Heidelberg (2007)
11. Paquette, M., Pelc, A.: Fast broadcasting with byzantine faults. International Journal of Foundations of Computer Science 17(6), 1423–1439 (2006)
12. Pelc, A.: Fault-tolerant broadcasting and gossiping in communication networks. Networks 28(6), 143–156 (1996)
13. Penrose, M.D.: On k-connectivity for a geometric random graph. Random Struct. Alg. 15, 145–164 (1999)
14. Shakkottai, S., Srikant, R., Shroff, N.: Unreliable sensor grids: Coverage, connectivity and diameter. In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 1073–1083 (2003)
15. Thottan, M., Ji, C.: Using network fault predictions to enable IP traffic management. J. Network Syst. Manage 9(3), 327–346 (2001)
16. Yajnik, M., Kurose, J., Towsley, D.: Packet loss correlation in the MBone multicast network. In: Proceedings of IEEE Global Internet, pp. 94–99 (1996)

The Mathematics of Routing in Massively Dense Ad-Hoc Networks

Eitan Altman[1], Pierre Bernhard[2], and Alonso Silva[1]

[1] INRIA, 2004 Route des Lucioles - BP 93, 06902 – Sophia Antipolis, France
{eitan.altman,alonso.silva}@sophia.inria.fr
[2] I3S, University of Nice-Sophia Antipolis and CNRS, France
Pierre.Bernhard@polytech.unice.fr

Abstract. Computing optimal routes in massively dense adhoc networks becomes intractable as the number of nodes becomes very large. One recent approach to solve this problem is to use a fluid type approximation in which the whole network is replaced by a continuum plain. Various paradigms from physics have been used recently in order to solve the continuum model. We propose in this paper an alternative modeling and solution approach similar to a model by Beckmann [3] developed more than fifty years ago from the area of road traffic.

1 Introduction

An important approach to routing in ad-hoc network has been to to design traffic dependent adaptive protocols that send packets along paths that have smallest delays. This metrics goes back to an early paper by Gupta and Kumar [8] who show that by doing so, resequensing delays (that are undesirable in real time traffic and that are very harmful in data transfers using the TCP protocol) are minimized. A recent line of research has been to study the such protocols in massively dense static ad-hoc networks that are characterized by the property that each node has many other nodes in its transmission range. We are interested here in the recent fluid limit approach in which the nodes are modeled as a continuum, and where the discrete graph describing the links and their costs is replaced by a cost density (which depends on the traffic intensity) over the plain. The rational of using such fluid limit approximations is that whereas the complexity of finding optimal routes grows with the number n of nodes, the fluid limit does not depend on n and hence the complexity of finding optimal routes in the fluid approximation does not grow with n.

Various approaches inspired by physics have been proposed starting with the pioneering work of Jacquet (see [10]) who used ideas from geometrical optics[1]. Approaches based on electrostatics have been designed in [20,21,18,17,9] (see the survey [19] and references therein).

The physics-inspired paradigms allow one to minimize various metrics related to the routing. In contrast, Hyytia and Virtamo propose in [15] an approach based on load balancing arguing that if shortest path (or cost minimization) arguments were used, then some parts of the network would carry more traffic than others and may use more

[1] We note that this approach is restricted to costs that do not depend on the congestion.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 122–134, 2008.
© Springer-Verlag Berlin Heidelberg 2008

energy than others. This would result in a shorter lifetime of the network since some parts would be out of energy earlier than others and earlier than any part in a load balanced network.

The development of the original theory of routing in massively dense ad-hoc networks has emerged in a complete independent way of the related theory developed within the community of road traffic engineers, introduced in 1952 by Wardrop [22] and by Beckmann [3][2], and which is still an active research area among that community, see [5,6,13,14,24] and references therein.

This community further developed numerical approaches to solve the continuous approximation model through discretization[3].

Inspired by Dafermos [5] who considered routing over two possible directions (North to South and West to East), we have studied in [1] routing in static ad-hoc networks (e.g. sensor networks) where the limitation to two directions can be justified by the use of directional antennas. In the present work, we study the case where any general direction can be chosen at any point.

Two types of objectives are sought in the research on routing in the road traffic context. The first is to maximize the global utility for the whole society, and the second is to find a routing configuration (called "traffic assignment") such that each transmission uses only paths with minimum costs. Configurations satisfying this property are known as "Wardrop Equilibrium", and they coincide with the solution concept used by Gupta and Kumar [8]. We study the two types of objectives in this paper in the context of massively dense ad-hoc networks. For the first objective (which corresponds to a cooperation between nodes) we use and strengthen results of Beckmann by using tools from optimization and control theory that have not been available at the middle of the last century. We further study the Wardrop equilibrium and establish conditions under which it coincides with the global optimization.

The paper is structured as follows. After describing the model in the next section, we provide in Section 3 the mathematical foundations for globally optimizing the fluid model. The mathematical foundation for describing and solving the non-cooperative case (i.e. the Wardrop equilibrium) are introduced in Section 4. This is followed by Section 5 with two examples for congestion cost. We end with a concluding section that summarizes our contributions.

2 The Problem

2.1 Routing in a Dense Network

We consider a routing problem in a dense ad-hoc network. A domain Ω of the plane (x, y) is densely covered by potential routers. Messages have to flow from a region $\mathcal{S}$ of the boundary Γ of Ω to a disjoint region $\mathcal{R}$ of Γ. The intensity $\sigma(x, y)$ of message

[2] See also [3, p 644, footnote 3] for the abundant literature of the early 50's.

[3] Although it may seem that one is back to the starting point with yet another discrete problem to solve, the new discrete problem is simpler, each node in it has only a small number of neighbors, and the number of nodes in the new discrete model is independent of the number of nods in the original system.

generation on $\mathcal{S}$ given, while the intensity $\rho(x, y)$ of signal sink on $\mathcal{R}$ is not. It is only assumed that these are consistent: the total flow of messages emitted and received are equal. On the rest $\mathcal{T}$ of the boundary of Ω, no message should enter nor leave Ω.

The congestion cost per packet transmitted (say in terms of delays, or energy use) at each point in Ω is a function $c(x, y, \varphi)$ of the point and of the intensity φ of the flow of messages through that point.

We wish to investigate the optimal routing policy and its relationship with a Wardrop kind of optimality.

2.2 A Mathematical Model

Formal Equations. We shall use the notation $\mathbf{x} = (x, y)$ to denote a point of $\mathbb{R}^2$. Let Ω be an open domain of $\mathbb{R}^2$ with a smooth boundary Γ, Ω being at every point of Γ on a single side of Γ, so that an exterior normal to Ω, say $n(\mathbf{x})$ is well defined and smooth on Γ.

We model the flow of messages as a vector field $f : \Omega \to \mathbb{R}^2$, and we let $\varphi(\mathbf{x}) = \|f(\mathbf{x})\|$ be its intensity. The flux of messages through $\mathcal{S}$ is given as a $\mathcal{C}^1$ function $\sigma(\cdot) : \mathcal{S} \to \mathbb{R}_+$. The consistency assumption now reads

$$\int_{\mathcal{R}} \rho(\mathbf{x})\, \mathrm{d}s = \int_{\mathcal{S}} \sigma(\mathbf{x})\, \mathrm{d}s\,. \tag{1}$$

Let $\mathcal{Q} = \mathcal{S} \cup \mathcal{T}$ and extend the function σ to the whole of $\mathcal{Q}$ by $\sigma(\mathbf{x}) = 0$ on $\mathcal{T}$. We model the conditions on the boundary as

$$\forall \mathbf{x} \in \mathcal{Q}\,, \quad \langle n(\mathbf{x}), f(\mathbf{x})\rangle = -\sigma(\mathbf{x}) \tag{2}$$

There is no source nor sink of messages in Ω, which we model as a constraint

$$\forall \mathbf{x} \in \Omega\,, \quad \mathrm{div} f(\mathbf{x}) = 0\,. \tag{3}$$

It follows that

$$\int_{\Gamma} \langle n(\mathbf{x}), f(\mathbf{x})\rangle \mathrm{d}s = 0\,,$$

which suffices to insure the consistency condition (1).

The congestion cost per packet c is supposed to be a strictly positive $\mathcal{C}^1$ function $c(\mathbf{x}, \varphi) : \Omega \times \mathbb{R}_+ \to \mathbb{R}_+$, increasing and convex in φ for each $\mathbf{x}$. The total cost of congestion will be taken as

$$G(f(\cdot)) = \int_{\Omega} c(\mathbf{x}, \|f(\mathbf{x})\|)\|f(\mathbf{x})\|\, \mathrm{d}\mathbf{x}\,. \tag{4}$$

The path followed by a packet is specifed by its direction of travel $e_\theta = (\cos\theta,\ \sin\theta)$ along its path, according to $\dot{\mathbf{x}} = e_\theta$. The cost incurred by one packet traveling from $\mathbf{x}_0 \in \mathcal{S}$ at time t_0 to $\mathbf{x}_1 \in \mathcal{R}$ reached at time t_1 is

$$J(e_\theta(\cdot)) = \int_{\mathbf{x}_0}^{\mathbf{x}_1} c(\mathbf{x}, \|f(\mathbf{x})\|)\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2} = \int_{t_0}^{t_1} c(\mathbf{x}(t), \|f(\mathbf{x}(t))\|)\, \mathrm{d}t\,. \tag{5}$$

Notice that this "time" t may be a fictitious time, related to physical time, say τ, by $\mathrm{d}\tau = c\,\mathrm{d}t$ for instance. Then c is the inverse of a speed of travel, a delay due to congestion, and J is the time taken by the message to go from source to destination.

Regularity and Function Spaces. We shall seek $f(\cdot)$ in a space we call V. We next discuss the choice of function spaces. A non mathematical oriented reader may skip the description of the funcntion spaces we introduce.

We may choose $V = (H^1(\Omega))^2$, but this will require $\sigma(\cdot)$ to be slightly more regular than necessary, viz. $H^{1/2}(\Gamma)$. To keep with the classical hypothesis in fluid dynamics, we may choose $V = (H_{\mathrm{div}}(\Omega))^2$, the space of L^2 functions whose divergence is in L^2. Then we my choose $\sigma(\cdot)$ in $L^2(\Gamma)$.

The above Sobolev spaces have been introduced by the modern theory of PDEs [7]. An extensive extensive Theory of PDEs and their numerical approximations is now available in these spaces.

This choice of spaces allows one to have complete spaces for functions and for their derivatives along with a scalar product of L^2. The completeness is needed to have existence of minima. The scalar product allows to have duality. The completeness together with the duality allows KKT Theorem to hold, which we make use of in this paper.

Let V_0 be the closure in V of the set of C^∞ functions with compact support in Ω. Let $V_{\mathcal{R}}$ and $V_{\mathcal{Q}}$ be the closures in V of the set of $\mathcal{C}^\infty$ functions that are null in a neighborhood of $\mathcal{R}$ and $\mathcal{Q}$ respectively. They are vector spaces, supersets of V_0. Let also $\tilde{f}(\cdot) : \Omega \to \mathbb{R}^2$ be a vector field in V satisfying the constraint (2) (for instance a smooth extension of $\sigma(\mathbf{x})n(\mathbf{x})$). Let $\mathcal{V}$ be the affine space $\tilde{f} + V_{\mathcal{Q}}$.

We shall also need the space $H^1_{\mathcal{R}}$ of functions of $H^1(\Omega)$ whose trace on $\mathcal{R}$ is zero.

Finally, we let $\Omega_0 = \{\mathbf{x} \mid f^\star(\mathbf{x}) = 0\}$, or more precisely, since $f^\star$ is not necessarily continuous, the largest open subset of Ω over which $\int_{\Omega_0} \|f^\star(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x} = 0$.

2.3 The Case of Elastic Traffic

Let's assume that we do not have to ship the whole demand $\sigma(x)$ to the destination. We shall send less if there is congestion. The standard way to model that is first to define a utility $u(s)$ for having s units of information shipped; we take $s(x) \leq \sigma(x)$. The new objective is to minimize the sum of $C(f) - U(s)$ where $U(s)$ is the integral of $u(s(x))$ over x.

One way to solve the problem is to define a new sink S. Then add an alternative route from each source to S; the cost to ship f units from a source x to S is $-u(\sigma(x) - f)$. Thus instead of directly adding utilities to the optimization problem, they appear through costs of new routes that are added. The elastic routing problem is thus transformed into an equivalent routing problem with fixed demand. This transformation is standard, see [12,16], and we shall not pursue it here.

3 Global Optimum

3.1 The Completely Differentiable Case

We seek here the vector field $f^\star \in (L^2(\Omega))^2$ satisfying the constraints (2) and (3) and minimizing $G(f)$.

Let $C(\mathbf{x}, \varphi) = c(\mathbf{x}, \varphi)\varphi$. It is convex in φ and coercive (i.e. goes to infinity with φ). As a consequence, $f(\cdot) \mapsto G(f(\cdot))$ is continuous, convex and coercive. Moreover,

the constraints are linear. Therefore an optimum exists, and we may apply the theorem "KKT" (Karush, Kuhn and Tucker).

We dualize only the constraint (3) and look for $f^\star$ in $\mathcal{V}$. Let therefore $p(\cdot) \in L^2(\Omega)$ be the dual variable, we let

$$\mathcal{L}(f,p) = \int_\Omega \Big(C(\mathbf{x}, \|f(\mathbf{x})\|) + p(\mathbf{x}) \mathrm{div} f(\mathbf{x}) \Big) \mathrm{d}\mathbf{x} \,.$$

Using Green's formula, we may also write

$$\mathcal{L}(f,p) = \int_\Omega \Big(C(\mathbf{x}, \|f(\mathbf{x})\|) - \langle \nabla p(\mathbf{x}), f(\mathbf{x}) \rangle \Big) \mathrm{d}\mathbf{x} + \int_\Gamma p(\mathbf{x}) \langle n(\mathbf{x}), f(\mathbf{x}) \rangle \mathrm{d}s \,.$$

The optimal vector field $f^\star$ should minimize $\mathcal{L}$ over $\mathcal{V}$, for some p. Therefore, 0 must belong to the subdifferential with respect to f of the restriction of $\mathcal{L}$ to $\mathcal{V}$.

Wherever $f^\star \neq 0$, $\mathcal{L}$ is actually differentiable, so that the subdifferential contains only the derivative. Actually, we only need the restriction of the derivative to $V_\mathcal{Q}$. which

$$\mathrm{D}\mathcal{L}\cdot g = \int_\Omega \Big(\mathrm{D}_2 C(\mathbf{x}, \|f^\star(\mathbf{x})\|) \frac{\langle f^\star(\mathbf{x}), g(\mathbf{x}) \rangle}{\|f^\star(\mathbf{x})\|} - \langle \nabla p(\mathbf{x}), g(\mathbf{x}) \rangle \Big) \mathrm{d}\mathbf{x} + \int_\mathcal{R} p(\mathbf{x}) \langle n(\mathbf{x}), g(\mathbf{x}) \rangle \mathrm{d}s \,,$$

should be zero for every $g \in V_\mathcal{Q}$. Pick first g in V_0. The last integral vanishes. It follows that necessarily

$$\forall \mathbf{x} : f^\star(\mathbf{x}) \neq 0 \,, \quad \mathrm{D}_2 C(\mathbf{x}, \|f^\star(\mathbf{x})\|) \frac{f^\star(\mathbf{x})}{\|f^\star(\mathbf{x})\|} = \nabla p(\mathbf{x}) \,. \tag{6}$$

It follows from this equation that $p(\cdot) \in H^1(\Omega)$, and also that the first integral in the r.h.s. must be zero for every g in $V_\mathcal{Q}$. Picking now $g \in V_\mathcal{Q}$. It follows that

$$p(\cdot) \in H^1_\mathcal{R} \tag{7}$$

Wherever $\|f^\star(\mathbf{x})\| = 0$, a discussion arises. If $D_2 C(\mathbf{x}, \varphi)/\varphi$ remains bounded as $\varphi \to 0$, there is nothing to add to equations (6) and (7) above. (We shall see the typical example $C(\mathbf{x}, \varphi) = (1/2) c(\mathbf{x}) \varphi^2$ below.) Otherwise the situation is more complicated.

3.2 Lack of Differentiability

We investigate now the case where $\mathrm{D}_2 C(\mathbf{x}, \varphi)/\varphi \to \infty$ as $\varphi \to 0$. This typically arises, e.g. if $\mathrm{D}_2 C(\mathbf{x}, 0) \neq 0$. We shall see the typical example $C(\mathbf{x}, \varphi) = c(\mathbf{x}) \varphi$ below.

Then $f \mapsto C(\mathbf{x}, \|f\|)$ is not differentiable (with respect to f) at 0. Its subdifferential is the set

$$\partial_f C(\mathbf{x}, 0) = \{ q \in \mathbb{R}^2 \mid \forall g \in \mathbb{R}^2 \,,\, C(\mathbf{x}, \|g\|) - C(\mathbf{x}, 0) \geq \langle q, g \rangle \} \,.$$

Since C is assumed differentiable and convex in its second argument, this is equivalent to

$$\partial_f C(\mathbf{x}, 0) = \{ q \mid \forall g \in \mathbb{R}^2 \,,\, \mathrm{D}_2 C(\mathbf{x}, 0) \|g\| \geq \langle q, g \rangle \} \,,$$

which in turn is equivalent to $\|q\| \leq |\mathrm{D}_2 C(\mathbf{x}, 0)|$. Now, since C is assumed increasing in φ, $\mathrm{D}_2 C \geq 0$. Placing this back into the subdifferential of $\mathcal{L}$, we get, for $\mathbf{x} \in \Omega_0$,

$$\exists q(\mathbf{x}) \text{ such that } \|q(\mathbf{x})\| \leq \mathrm{D}_2 C(\mathbf{x}, 0) \text{ and } \forall g \in V_{\mathcal{Q}}\,, \int_{\Omega_0} (q(\mathbf{x}) - \nabla p(\mathbf{x})) g(\mathbf{x})\, \mathrm{d}\mathbf{x} = 0\,.$$

Combining both cases, we conclude that, for a function $f^\star(\cdot) \in V$ with null set Ω_0 to be optimal, there must exist a $p(\cdot) \in H^1_{\mathcal{R}}$ such that

$$\begin{aligned} &\forall \mathbf{x} \in \Omega\,, \qquad \|\nabla p(\mathbf{x})\| \leq \mathrm{D}_2 C(\mathbf{x}, 0)\,, \\ &\forall \mathbf{x} \in \Omega - \Omega_0\,, \nabla p(\mathbf{x}) = \mathrm{D}_2 C(\mathbf{x}, \|f(\mathbf{x})^\star\|) \tfrac{1}{\|f^\star(\mathbf{x})\|} f^\star(\mathbf{x})\,. \end{aligned} \tag{8}$$

We may notice that the first condition above also yields

$$\forall \mathbf{x} : f^\star(\mathbf{x}) \neq 0\,, \quad \|\nabla p(\mathbf{x})\| = D_2 C(\mathbf{x}, \|f^\star(\mathbf{x})\|)\,,$$

Overall, the problem of determining the optimum $f^\star$ is equivalent (if that system has a single solution) to determining simultaneously $f^\star$ and p satisfying (2),(3) and (8).

This system certainly has at least one solution, since our problem is convex coercive with affine constraints, and thus has a minimum. Uniqueness on the other hand, is by no means simple. It may be noticed that one might look for the two scalar functions φ and p, satisfying

$$\begin{aligned} &\forall \mathbf{x} : \varphi(\mathbf{x}) \neq 0\,, \|\nabla p(\mathbf{x})\| = \mathrm{D}_2 C(\mathbf{x}, \varphi(\mathbf{x}))\,, \\ &\forall \mathbf{x} : \varphi(\mathbf{x}) = 0\,, \|\nabla p(\mathbf{x})\| \leq \mathrm{D}_2 C(\mathbf{x}, 0)\,, \\ &\forall \mathbf{x} \in \mathcal{R}\,, \qquad p(\mathbf{x}) = 0\,, \end{aligned}$$

and impose furthermore the constraints (2) and (3) on

$$f^\star(x) = \frac{\varphi(\mathbf{x})}{\mathrm{D}_2 C(\mathbf{x}, \varphi(\mathbf{x}))} \nabla p(\mathbf{x})\,.$$

We shall investigate a typical case hereafter.

4 Wardrop Equilibrium

Assume the message flow obeys the above necessary conditions. We want to investigate whether it is optimal for a single message to follow the route prescribed by $f^\star$, i.e. an integral line of that field, assuming that its lone deviation from that scheme would have no effect on the overall congestion map. (This is the so called "atomicity" assumption.)

We investigate the optimization of the criterion (5) via its Hamilton-Jacobi-Bellman equation. Let $V(\mathbf{x})$ be the return function, it must be a viscosity solution of

$$\begin{aligned} &\forall \mathbf{x} \in \Omega\,, \min_\theta \langle e_\theta, \nabla V(\mathbf{x}) \rangle + c(\mathbf{x}, \|f^\star(\mathbf{x})\|) = 0\,, \\ &\forall \mathbf{x} \in \mathcal{R}\,, V(\mathbf{x}) = 0\,. \end{aligned}$$

hence

$$\begin{aligned} &\forall \mathbf{x} \in \Omega\,, -\|\nabla V(\mathbf{x})\| + c(\mathbf{x}, \|f^\star(\mathbf{x})\|) = 0\,, \\ &\forall \mathbf{x} \in \mathcal{R}\,, V(\mathbf{x}) = 0\,. \end{aligned} \tag{9}$$

And the optimal direction of travel is opposite to $\nabla V(\mathbf{x})$, i.e. $e_\theta = -\nabla V(\mathbf{x}) / \|\nabla V(\mathbf{x})\|$.

Clearly, this is the same system of equations as previously, upon replacing $p(\mathbf{x})$ by $-V(\mathbf{x})$, and $\mathrm{D}_2 C(\mathbf{x},\varphi)$ by $c(\mathbf{x},\varphi)$. We thus conclude that the Wardrop equilibrium can be obtained by solving the globally optimal problem in which the cost density is replaced by $\int_0^\varphi c(\mathbf{x},\varphi)d\varphi$. This is the continuous version of the potential function approach of Beckmann et al. [4]. This transformation has been frequently used in the road traffic context but only for one particular cost structure [23,24,25,26] the equivalence was shown to hold in [23,25].

Monomial cost. In the case where $c(\mathbf{x},\varphi) = c(\mathbf{x})\varphi^\alpha$, then $C(\mathbf{x},\varphi) = \alpha c(\mathbf{x},\varphi)$, and therefore the two systems of equations coincide, or more precisely, they coincide in the domain $\{\mathbf{x} \mid f^\star(\mathbf{x}) \neq 0\}$. We shall show that for a given $\varphi(\cdot)$, p is uniquely defined. We therefore have the following property :

Proposition 1. *For a monomial cost, any global equilibrium where Ω_0=$\emptyset$ is a Wardrop equilibrium.*

5 Two Examples

5.1 Linear Congestion Cost

We investigate here the simple typical case, where the cost of congestion is linear : $c(\mathbf{x},\varphi) = \frac{1}{2}c(\mathbf{x})\varphi$, so that

$$C(\mathbf{x},\varphi) = \frac{1}{2}c(\mathbf{x})\varphi^2\,.$$

Then, $\mathcal{L}$ is differentiable everywhere, and the necessary condition of optimality is just that there should exist $p : \Omega \to \mathbb{R}^2$ such that $\nabla p(\mathbf{x}) = c(\mathbf{x}) f^\star(\mathbf{x})$. Placing this into (3) and (2), we see that we end up with a simple elliptic equation with mixed Dirichlet - (non-homogeneous) Neuman boundary conditions :

$$\left.\begin{array}{ll} \forall \mathbf{x} \in \Omega\,, & \mathrm{div}(\frac{1}{c(\mathbf{x})}\nabla p(\mathbf{x})) = 0\,, \\ \forall \mathbf{x} \in \mathcal{Q}\,, & \dfrac{\partial p}{\partial n}(\mathbf{x}) = c(\mathbf{x})\sigma(\mathbf{x})\,, \\ \forall \mathbf{x} \in \mathcal{R}\,, & p(\mathbf{x}) = 0\,, \end{array}\right\} \tag{10}$$

for which we easily get existence and uniqueness of the solution.

A more or less explicit solution can then be given in terms of the Green function $\mathcal{G}(\mathbf{x},\xi)$ of the domain

$$f^\star(\mathbf{x}) = \int_{\mathcal{Q}} \frac{1}{c(\mathbf{x})}\nabla_1 \mathcal{G}(\mathbf{x},\xi)\sigma(\xi)\mathrm{d}s(\xi)\,.$$

If he Green function is not available, according to a classical approach, we may derive a finite element method from the variational form : Find $p \in H^1_{\mathcal{R}}$ such that, for any $q \in H^1_{\mathcal{R}}$,

$$\int_\Omega \frac{1}{c(\mathbf{x})}\langle \nabla p(\mathbf{x}), \nabla q(\mathbf{x})\rangle \mathrm{d}x - \int_{\mathcal{Q}} \sigma(\mathbf{x})q(\mathbf{x})\,\mathrm{d}s = 0\,.$$

This can be read as $\mathrm{D}K(p) = 0$ where $K : H^1_{\mathcal{R}} \to \mathbb{R}$ is given by

$$K(p) = \frac{1}{2}\int_{\Omega} \frac{1}{c(\mathbf{x})}\|\nabla p(\mathbf{x})\|^2 - \int_{\mathcal{Q}} \sigma(\mathbf{x})p(\mathbf{x})\,\mathrm{d}s\,.$$

Thanks to Poincaré's inequality, it is convex coercive. We therefore obtain:

Proposition 2. *Equations (10) have a unique solution* $p \in H^1_{\mathcal{R}}$.

5.2 Uncongested Network

An Algorithm. We consider now a situation where the network operates far from congestion. The "cost" $c(\mathbf{x})$ may be regarded as a delay, then the cost of any trajectory is just the time it takes, or an energy expenditure. In any case, it is related to the state of the infrastructure, not to its load. Then, c is independent of $\|f(\mathbf{x})\|$, and we get $C(\mathbf{x}, \varphi) = c(\mathbf{x})\varphi$. Then, (8) simplifies into

$$\begin{aligned} &\forall \mathbf{x} \in \Omega\,, \quad \|\nabla p(\mathbf{x})\| \le c(\mathbf{x})\,,\\ &\forall \mathbf{x} : f^\star(\mathbf{x}) \neq 0\,, \quad \nabla p(\mathbf{x}) = c(\mathbf{x})\frac{f^\star(\mathbf{x})}{\|f^\star(\mathbf{x})\|}\,. \end{aligned}$$

Let

$$\varphi(\mathbf{x}) = \|f^\star(\mathbf{x})\|\,, \quad \psi(\mathbf{x}) = \frac{\varphi(\mathbf{x})}{c(\mathbf{x})}\,.$$

The above system yields

$$\forall \mathbf{x} \in \Omega\,, \quad \psi(\mathbf{x}) \ge 0\,, \quad \|\nabla p(\mathbf{x})\| \le c(\mathbf{x})\,, \quad \psi(\mathbf{x})[\|\nabla p(\mathbf{x})\| - c(\mathbf{x})] = 0\,, \tag{11}$$

and also $f^\star(\mathbf{x}) = \psi(\mathbf{x})\nabla p(\mathbf{x})$, which placed in (3) and (2) yields

$$\begin{aligned} &\forall \mathbf{x} \in \Omega\,, \quad \psi(\mathbf{x})\Delta p(\mathbf{x}) + \langle \nabla\psi(\mathbf{x}), \nabla p(\mathbf{x})\rangle = 0\,,\\ &\forall \mathbf{x} \in \Gamma\,, \quad \psi(\mathbf{x})\langle n(\mathbf{x}), \nabla p(\mathbf{x})\rangle = \sigma(\mathbf{x})\,. \end{aligned} \tag{12}$$

We do not have a satisfactory theory of this equation. If, as we noticed, existence is guaranteed, we do not know whether that solution is unique. It should be noticed that the uniqueness proof given for a very similar equation in [3] does nor carry over here, because it relies critically on the strict convexity of the cost in $\|f\|$.

As an attempt, we provide here an iterative algorithm which, if it converges, converges toward a solution of the system. It provides us with a uniqueness result under a strong hypothesis. We suspect that a more general result is true, and also that the algorithm converges even without that hypothesis.

We seek ψ in $H^1(\Omega)$, and p in $H^1_{\mathcal{R}}$.

Using the classical variational trick, we may reformulate system (12) as $\forall q \in V_{\mathcal{R}}$,

$$\int_{\Omega}[\psi(\mathbf{x})\Delta p(\mathbf{x}) + \langle \nabla\psi(\mathbf{x}), \nabla p(\mathbf{x})\rangle]q(\mathbf{x})\,\mathrm{d}\mathbf{x} - \int_{\mathcal{Q}}[\psi(\mathbf{x})\langle n(\mathbf{x}), \nabla p(\mathbf{x})\rangle - \sigma(\mathbf{x})]q(\mathbf{x})\mathrm{d}s = 0\,.$$

Using Green's formula for $q \in H^1(\Omega)$:

$$\int_\Omega [\psi(\mathbf{x})\Delta p(\mathbf{x}) + \langle \nabla\psi(\mathbf{x}), \nabla p(\mathbf{x})\rangle] q(\mathbf{x})\,\mathrm{d}\mathbf{x} = \\ -\int_\Omega \psi(\mathbf{x})\langle \nabla p(\mathbf{x}), \nabla q(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x} + \int_\Gamma \psi(\mathbf{x})\langle n(\mathbf{x}), \nabla p(\mathbf{x})\rangle q(\mathbf{x})\,\mathrm{d}s\,,$$

system (12) can therefore be stated as:

$$\forall q \in V_{\mathcal{R}}\,, \quad \int_\Omega \psi(\mathbf{x})\langle \nabla p(\mathbf{x}), \nabla q(\mathbf{x})\rangle\,\mathrm{d}\mathbf{x} - \int_{\mathcal{Q}} \sigma(\mathbf{x}) q(\mathbf{x})\,\mathrm{d}s = 0\,. \tag{13}$$

This equality may also be interpreted as $\mathrm{D}_1 J(p,\psi)q = 0$ where $J : V_{\mathcal{R}} \to \mathbb{R}$ is defined by

$$J(p,\psi) = \frac{1}{2}\int_\Omega \psi(\mathbf{x})\|\nabla p(\mathbf{x})\|^2\,\mathrm{d}\mathbf{x} - \int_{\mathcal{Q}} \sigma(\mathbf{x}) p(\mathbf{x})\,\mathrm{d}s\,.$$

Poincaré's inequality states that there exists $C > 0$ such that,

$$\forall p \in V_{\mathcal{R}}, \|p\|^2 \le C\|\nabla p\|^2\,. \tag{14}$$

Thus the functional J above is coercive and has a single minimum.

One may guess the following algorithm: fix $\psi^0(\mathbf{x})$ (say $= 1$). Given ψ^n, minimize J with respect to p, say solving the finite element equations corresponding to (13). Call p^n the solution, and do

$$\psi^{n+1}(\mathbf{x}) = \max\{0, \psi^n(\mathbf{x}) + \theta(\|\nabla p^n(\mathbf{x})\|^2 - c(\mathbf{x})^2)\} \tag{15}$$

for some positive θ. We shall prove the following theorem :

Proposition 3. *If there exists a solutin of equations (11)(12) such that $\|f^\star\|$ is essentially bounded away from 0 in Ω, it is unique and for θ small enough algorithm (15) converges toward that solution.*

Analysis of the Algorithm. Let $\psi^\star$, $p^\star$ be a soltion of our system of equations. Notice first that indeed, for any $\theta > 0$,

$$\forall \mathbf{x} \in \Omega\,, \quad \psi^\star(\mathbf{x}) = \max\{0, \psi^\star(\mathbf{x}) + \theta(\|\nabla p^\star(\mathbf{x})\|^2 - c(\mathbf{x})^2)\} \tag{16}$$

And any limit of he above algorithm has to satisfy this equation, which says that $\|\nabla p(\mathbf{x})\| = c(\mathbf{x})$ for every $\mathbf{x}$ where $\psi(\mathbf{x}) \neq 0$. Together with the condition that p minimizes J for ψ, this is exactly the conditions (11) and (12).

Substract (16) from (15). It results that

$$|\psi^{n+1}(\mathbf{x}) - \psi^\star(\mathbf{x})| \le |\psi^n(\mathbf{x}) - \psi^\star(\mathbf{x}) + \theta(\|\nabla p^n(\mathbf{x})\|^2 - \|\nabla p^\star(\mathbf{x})\|^2)|\,.$$

Take the square, and integrate over Ω :

$$\int_\Omega |\psi^{n+1}(\mathbf{x}) - \psi^\star(\mathbf{x})|^2\,\mathrm{d}\mathbf{x} \le \int_\Omega |\psi^n(\mathbf{x}) - \psi^\star(\mathbf{x})|^2\,\mathrm{d}\mathbf{x} \\ +2\theta\int_\Omega (\psi^n(\mathbf{x}) - \psi^\star(\mathbf{x}))(\|\nabla p^n(\mathbf{x})\|^2 - \|\nabla p^\star(\mathbf{x}))\|^2)\,\mathrm{d}\mathbf{x} \\ +\theta^2\int_\Omega (\|\nabla p^n(\mathbf{x})\|^2 - \|\nabla p^\star(\mathbf{x})\|^2)^2\,\mathrm{d}\mathbf{x}\,. \tag{17}$$

Using Cauchy-Schwarz inequality, the last term is bounded from above by

$$\int_\Omega (\|\nabla p^n(\mathbf{x})\|^2 - \|\nabla p^\star(\mathbf{x})\|^2)^2 \,\mathrm{d}\mathbf{x} \leq$$

$$\int_\Omega \|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x} \int_\Omega \|\nabla(p^n(\mathbf{x}) + p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x}\,.$$

Hence, assuming $\int_\Omega \|\nabla p^n(\mathbf{x})\|^2 \,\mathrm{d}\mathbf{x}$ remains bounded, there exists $a > 0$ such that

$$\int_\Omega (\|\nabla p^n(\mathbf{x})\|^2 - \|\nabla p^\star(\mathbf{x})\|^2)^2 \,\mathrm{d}\mathbf{x} \leq a \int_\Omega \|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x}\,. \tag{18}$$

Concerning the second term of the r.h.s. of (17), write

$$\|\nabla p^\star\|^2 = \|\nabla p^n + \nabla(p^\star - p^n)\|^2 = \|\nabla p^n\|^2 + 2\langle \nabla p^n, \nabla(p^\star - p^n)\rangle + \|\nabla(p^\star - p^n)\|^2\,.$$

Thus (using short notations for convenience)

$$\begin{aligned}
&\frac{1}{2}\int_\Omega \psi^n \|\nabla p^\star\|^2 \,\mathrm{d}\mathbf{x} - \int_\mathcal{Q} \sigma p^\star \,\mathrm{d}s \\
&= \frac{1}{2}\int_\Omega \psi^n \|\nabla p^n\|^2 \,\mathrm{d}\mathbf{x} - \int_\mathcal{Q} \sigma p^n \,\mathrm{d}s + \frac{1}{2}\int_\Omega \psi^n \|\nabla(p^\star - p^n)\|^2 \,\mathrm{d}\mathbf{x} \\
&\qquad + \int_\Omega \psi^n \langle \nabla p^n, \nabla(p^\star - p^n)\rangle \,\mathrm{d}\mathbf{x} - \int_\mathcal{Q} \sigma(p^\star - p^n) \,\mathrm{d}s\,.
\end{aligned}$$

By the definition of p^n as solving equation (13), the second line above is zero, leaving the first line alone. In a symmetric fashion, we also get

$$\frac{1}{2}\int_\Omega \psi^\star \|\nabla p^n\|^2 - \int_\mathcal{Q} \sigma p^n = \frac{1}{2}\int_\Omega \psi^\star \|\nabla p^\star\|^2 - \int_\mathcal{Q} \sigma p^\star + \frac{1}{2}\int_\Omega \psi^\star \|\nabla(p^n - p^\star)\|^2\,.$$

Summing the last two equalities (and multiplying by two), we obtain

$$\int_\Omega (\psi^n - \psi^\star)(\|\nabla p^n\|^2 - \|\nabla p^\star\|^2) = -\int_\Omega (\psi^n + \psi^\star)\|\nabla(p^n - p^\star)\|^2\,.$$

Placing this and (18) in (17), we may summarize the above calculations as

$$\begin{aligned}
&\int_\Omega |\psi^{n+1}(\mathbf{x}) - \psi^\star(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x} \leq \int_\Omega |\psi^n(\mathbf{x}) - \psi^\star(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x} \\
&-2\theta \int_\Omega (\psi^n(\mathbf{x}) + \psi^\star(\mathbf{x}))\|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x} \\
&+a\theta^2 \int_\Omega \|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x}\,.
\end{aligned} \tag{19}$$

Assume that, for almost all $\mathbf{x} \in \Omega$, $\psi^\star(\mathbf{x}) \geq b > 0$. It follows that

$$\int_\Omega (\psi^n(\mathbf{x}) + \psi^\star(\mathbf{x}))\|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x} \geq b \int_\Omega \|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \,\mathrm{d}\mathbf{x}\,,$$

and therefore that for any $\theta \leq b/a$,

$$\int_\Omega |\psi^{n+1}(\mathbf{x}) - \psi^\star(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} \leq \int_\Omega |\psi^n(\mathbf{x}) - \psi^\star(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} - b\theta \int_\Omega \|\nabla(p^n(\mathbf{x}) - p^\star(\mathbf{x}))\|^2 \, \mathrm{d}\mathbf{x} \, .$$

Summing these inequalities, it follows that the series of the L^2 norms $\|\nabla p^n \nabla p^\star\|^2$ converges, and according to Poincarés inequality again, $p^n \to p^\star$ in $H^1(\Omega)$. The field of optimal directions converges as well, and assuming it is regular enough for the integral curves to be unique, the optimal field converges as well.

The algorithm is independent from the choice of $p^\star$ and $\psi^\star$ who are therefore uniquely defined.

6 Concluding Comments

We present a brief comparison of our treatment with [3], called hereafter **M.B.**. In **M.B.**, one introduces both the density $u(\mathbf{x})$ of commodity to be moved, and the speed $v(\mathbf{x})$ of this motion, which is a data. And the cost of transportation is assumed to be a function of u alone. The decision variable in **M.B.** is the vector field φ of transportation where the direction of φ is that of the transportation, and $\|\varphi\|$ its density u. Hence **M.B.**'s $v\varphi$ is our f. And his equation (11) is our equation (6).

In **M.B.** there is an area source or sink of matter to be transported. It did not seem necessary in our context, but technically, it would be trivially done just adding a nonzero r.h.s. to equation (3) and its various forms, the first equation of (10) and of (12).

Now, since the early 50's, the theory of PDE's has been considerably developed, using the tools of Sobolev spaces and the variational theory of J-L. Lions, P. Lax, and others. Thus our derivation is not formal any more, and we are able to give existence and uniqueness theorems impossible to derive in 1952. Notice that our example with no congestion, where our uniqueness theorem is not very satisfactory, does not satisfy the hypotheses of the uniqueness theorem of **M.B.**, because that paper requires that the cost function be strictly convex.

Finally, we solve for the concept of Wardrop equilibrium, and we are therefore able to compare the global optimum to the Wardrop equilibrium, which was not available to Beckmann in 1952.

By casting the routing problem in dense Ad-hoc networks in the context of the road traffic framework of Beckmann, we are able to formulate and solve various optimization problems and study various cost functions, which was not the case with the physics-inspired paradigms that had been used before to study massively dense ad-hoc networks.

Acknowledgement

This work was partly supported by the INRIA grant (ARC) for promoting cooperation on Population, Game Theory and Evolution. The work of the first and third authors was partly supported by the BIONETS European contract.

References

1. Altman, E., Bernhard, P., Debbah, M., Silva, A.: Continuum Equilibria for Routing in Dense Ad-Hoc Networks. In: 45th Allerton Conference on Communication, Control and Computing, Illinois, USA, September 26 - 28 (2007)
2. Bardi, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Birkhauser, Basel (1994)
3. Beckmann, M.: A continuous model of transportation. Econometrica 20, 643–660 (1952)
4. Beckmann, M., McGuire, C.B., Winsten, C.B.: Studies in the Economics and Transportation. Yale Univ. Press (1956)
5. Dafermos, S.C.: Continuum Modeling of Transportation Networks. Transpn Res. 14B, 295–301 (1980)
6. Daniele, P., Maugeri, A.: Variational Inequalities and discrete and continuum models of network equilibrium protocols. Mathematical and Computer Modelling 35, 689–708 (2002)
7. Evans, L.C.: Partial Differentiali Equations. Graduate Studies in Mathematics, vol. 19. Ameriacn Mathematical Society (1998)
8. Gupta, P., Kumar, P.R.: A system and traffic dependent adaptive routing algorithm for ad hoc networks. In: Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, December 1997, pp. 2375–2380 (1997)
9. Gupta, G.A., Toumpis, S.: Optimal placement of nodes in large sensor networks under a general physical layer model. In: IEEE Secon, Santa Clara, CA (September 2005)
10. Jacquet, P.: Geometry of information propagation in massively dense ad hoc networks. In: MobiHoc 2004: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, pp. 157–162. ACM Press, New York (2004)
11. Fleming, W.H., Soner, H.M.: Controlled Markov Processes and Viscosity Solutions, 2nd edn. Springer, Heidelberg (2006)
12. Haurie, A., Marcott, P.: On the relationship between Nash-Cournot and Wardrop equilibria. Networks 15, 295–308 (1985)
13. Ho, H.W., Wong, S.C.: Two-Dimensional Continuum Modeling Approach to Transportation Problems. Journal of Transportation Systems Engineering and Information Technology 6(6), 53–72 (2006)
14. Idone, G.: Variational inequalities and applications to a continuum model of transportation network with capacity constraints. Journal of Global Optimization 28, 45–53 (2004)
15. Hyytia, E., Virtamo, J.: On load balancing in a dense wireless multihop network. In: Proceeding of the 2nd EuroNGI conference on Next Generation Internet Design and Engineering, Valencia, Spain (April 2006)
16. Patriksson, M.: The Traffic Assignment Problem: Models and Methods, VSP BV, The Netherlands (1994)
17. Tassiulas, L., Toumpis, S.: Packetostatics: Deployment of massively dense sensor networks as an electrostatic problem. IEEE INFOCOM 4, 2290–2301 (2005)
18. Tassiulas, L., Toumpis, S.: Optimal deployment of large wireless sensor networks. IEEE Transactions on Information Theory 52(7), 2935–2953 (2006)
19. Toumpis, S.: Mother nature knows best: A survey of recent results on wireless networks based on analogies with physics. Computer Networks 52, 360–383 (2008)
20. Toumpis, S.: Optimal design and operation of massively dense wireless networks: or how to solve 21st century problems using 19th century mathematics. In: interperf 2006: Proceedings from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications systems, ACM Press, New York (2006)
21. Toumpis, S., Catanuto, R., Morabito, G.: Optical routing in massively dense networks: Practical issues and dynamic programming interpretation. In: Proc. of IEEE ISWCS 2006 (September 2006)

22. Wardrop, J.G.: Some theoretial aspects of road traffic research. In: Proceedings of the Institution of Civil Engineers, pp. 325–378 (1952)
23. Wong, S.C.: Multi-Commodity Traffic Assignment by Continuum Approximation of Network Flow with Variable Demand. Transpn Res. -B. 32(8), 567–581 (1998)
24. Wong, S.C., Du, Y.C., Sun, J.J., Loo, B.P.Y.: Sensitivity analysis for a continuum traffic equilibrium problem. Ann. Reg. Sci. 40, 493–514 (2006)
25. Wong, S.C., Lee, C.K., Tong, C.O.: Finite element solution for the continuum traffic equilibrium problems. International Journal for Numerical Methods in Engineering 43(7), 1253–1273 (1998)
26. Yang, H., Juang, H.-J.: The multi-class, multi-criteria traffic network equilibrium and systems optimum problem. Transportation Research Part B 38, 1–15 (2004)

Localized Spanner Construction for Ad Hoc Networks with Variable Transmission Range

David Peleg* and Liam Roditty

Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel

Abstract. This paper presents an algorithm for constructing a spanner for ad hoc networks whose nodes have *variable* transmission range. Almost all previous spanner constructions for ad hoc networks assumed that all nodes in the network have the same transmission range. This allowed a succinct representation of the network as a unit disk graph, serving as the basis for the construction. In contrast, when nodes have variable transmission range, the ad hoc network must be modeled by a general disk graph. Whereas unit disk graphs are undirected, general disk graphs are directed. This complicates the construction of a spanner for the network, since currently there are no efficient constructions of low-stretch spanners for general directed graphs. Nevertheless, in this paper it is shown that the class of disk graphs enjoys (efficiently constructible) spanners of quality similar to that of unit disk graph spanners. Moreover, it is shown that the new construction can be done in a localized fashion.

1 Introduction

A wireless ad hoc network is composed of a collection S of n nodes distributed in the two dimensional plane. The nodes can communicate with each other using wireless connections. As opposed to cellular networks, there is no wire infrastructure and the connections between the nodes are restricted by their transmission energy. As nodes often receive their energy from a battery, reducing energy consumption is one of the most fundamental problems in the design of ad hoc networks. A popular approach for coping with the challenge of designing an efficient ad hoc network is to find a topology in which only a linear number of links need to be maintained, while the degradation of paths that connect any pair of nodes is restricted.

In the common wireless network model, the power needed to transmit from p to q is $|pq|^\alpha$, where $|pq|$ is the Euclidean distance between p and q and α is a constant that varies between 2 and 4. The basic assumption adopted in most of the literature on ad hoc networks is that all the nodes have the same transmission range. Consequently, the ad hoc network can be represented using a *unit disk graph*, that is, a graph in which two nodes share an edge if their

* Supported in part by grants from the Minerva Foundation and the Israel Ministry of Science.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 135–147, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Euclidean distance is at most 1. The size (in edges) of the unit disk graph can be as large as $O(n^2)$.

One fundamental object used in the design of ad hoc network topologies is a *spanner* [16,18,19]. A graph H is a t-spanner of a graph G if $\delta_H(u,v) \leq t \cdot \delta_G(u,v)$ for every two nodes u and v, where $\delta_G(u,v)$ denotes the shortest path distance between u and v in the graph G and H is a subgraph of G. The parameter t is referred to as the *stretch factor* of the spanner.

There is an extensive body of literature on spanners in both the geometric setting and the ad hoc setting. In the geometric setting, the graph G to be spanned is the complete graph over a set S of n points, where the weight of each edge of G is the distance between its endpoints in $\mathbb{R}^d$. Yao in [26] , Vaidya [23], Salowe [21] and Callahan and Kosaraju [3] showed how to compute a geometric $(1+\epsilon)$-spanner with $O(n/\epsilon^d)$ edges in $O(n \log n)$ time. In [11], Gao et. al. showed how to maintain a $(1+\epsilon)$-spanner in a distributed manner in a mobile setting, i.e., when points can move.

In the ad hoc settings, where the graph to be spanned is a unit disk graph, the most popular constructions that are used as underlying network topologies for routing are the *relative neighborhood graph* (RNG) and *Gabriel graph* (GG) which are planar subgraphs (see [2,12]). These graphs might suffer a very high stretch in the worst-case. Subsequent work by Gao et. al. [10], Wang and Yang-Li [24] and Yang-Li et. al. [14] considered the restricted Delaunay graph, whose worst-case stretch is constant (larger than $1+\epsilon$). In [25], Wang and Yang-Li showed how to construct a spanner of bounded degree which is also planar. That spanner too has constant stretch.

Spanners in ad hoc networks have crucial role. Not only do they preserve the connectivity of the network but they also guarantee that the distance between every pair of nodes is within some constant factor from the shortest possible distance. Moreover, the size of the spanner is only linear. These properties made the use of spanners an attractive approach for ad hoc networks. To learn more on the tight connection between topology control in ad hoc networks and spanners see [20].

Common to all the papers mentioned above in the ad hoc setting is the assumption that the ad hoc network is represented by a unit disk graph, that is, every node of the network is assumed to have the same transmission range. This model is of significant theoretical appeal, but its accuracy is limited due to the fact that coverage areas are assumed to be disk of equal radius, implying in particular that transmission coverage must be symmetric. The focus on the restricted model of unit disk graph is partially explained by the lack of methods for dealing with more general models on one hand and the attractive properties of unit disk graphs on the other hand. There are few papers which studied more general models than the unit disk graph, such as the Quasi-unit disk graph in [13] and [17], however, these models are still limited. Li, Song and Wang [15] considered a model similar to ours, in which every node has a different transmission range. In their model an edge connects u and v in the communication

graph only if u can transmit to v and v can transmit to u. Thus, the resulting graph is still undirected.

The current paper considers a more general and sometimes more natural case in which any node has a different transmission range, taken from the range $[1, M]$, and an edge is placed from u to v if u can transmit to v. This yields an intermediate model between the geometric setting and the usual ad hoc setting, as the transmission graph induced in this case is no longer a unit disk graph but a general disk graph. In such graphs, edges have a direction, since the fact that p can transmit to q does not necessarily imply that q can transmit to p. Thus, the resulting graph is directed and the transmission coverage is no longer assumed to be symmetric. In this respect, our work can be viewed as an intermediate step towards more general coverage models.

The main result of the current paper (in Section 2) is an algorithm for constructing a $(1+\epsilon)$-spanner for a given disk graph with $O(n/\epsilon^{-d} \log M)$ edges. The algorithm can be implemented in $O(m \log n)$ time, where m is the number of edges in the disk graph.

Our result is also of theoretical significance. Finding good spanners for directed graphs is a difficult problem. A general bound, similar to the one available for undirected graphs, cannot exist for the directed case, as indicated by considering the example of a directed bipartite graph in which all the edges are directed from one side to the other; clearly, any spanner for such a graph must contain every edge. In that sense, our spanner construction yields the first result establishing the existence of a directed spanner for a non-trivial class of directed graphs.

Many routing protocols for ad hoc network use only the local information which is stored with every node. In such algorithms a packet is routed out from a node by considering only its neighbors in the topology. See [1,2,12,22] for more information. As our topology is constructed on top of a directed network our result opens a new direction for localized routing algorithms.

In addition, the paper also presents (in Section 3) an algorithm for constructing a linear size ($O(n/\epsilon^{-d})$ edges) $(1+\epsilon)$-spanner for a given unit disk graph. In particular, we show that any geometric $(1+\epsilon)$-spanner can be turned into a $(1+\epsilon')$-spanner for a unit disk graph by applying a simple process.

2 Spanners for General Disk Graphs

Let S be a set of points in $\mathbb{R}^d$ and assume that any point $p \in S$ has a transmission radius $r(p)$, taken from the range $[1, M]$. The transmission graph of S is a disk graph $I(S, E)$, whose vertices are the points of S and whose edge set includes an edge from p to q if p can transmit to q. Obviously, the resulting graph is directed, as it might happen that p can transmit to q while q cannot transmit to p. In this section we show how to compute a $(1+\epsilon)$-spanner with $O(n/\epsilon^{-d} \log M)$ edges for a given disk graph.

The construction of the spanner is based on hierarchal partition of the points of S that takes into account the variable transmission radii.

Let ϵ be an arbitrarily small positive constant and let α and β be two small constants depending on ϵ, to be fixed later on. Assume that the transmission radii are scaled so that the smallest edge in the disk graph is of weight 1. Let i be an integer from the range $[0, \lfloor \log_{1+\alpha} M \rfloor]$ and let $M_i = M/(1+\alpha)^i$. Let $E(M_{i+1}, M_i) = \{(x,y) \mid M_{i+1} \leq |xy| \leq M_i\}$. Let $\ell(x,y)$ be the level of the edge (x,y), that is, if $(x,y) \in E(M_{i+1}, M_i)$ then $\ell(x,y) = i$. Let p be a point with a transmission radius $r(p) \in [M_{i+1}, M_i]$. It follows that level i is the first level in which p can have outgoing edges. We denote this level with $\ell(p)$.

The spanner construction algorithm receives as input a (directed) disk graph $I(S,E)$ and a desired approximation factor ϵ. It constructs the set of spanner edges $E_{\text{SP}}^{\text{DIR}}$ and returns the graph $H^{\text{DIR}}(S, E_{\text{SP}}^{\text{DIR}})$. The construction is as follows. The edges of $I(S,E)$ are partitioned into classes $E(M_{i+1}, M_i)$ for $i \in [0, \lfloor \log_{1+\alpha} M \rfloor]$. Assume that in each class the edges are sorted by their weight. For every $i \in [0, \lfloor \log_{1+\alpha} M \rfloor]$, starting from $i = 0$, the edges of the class $E(M_{i+1}, M_i)$ are considered in a non-decreasing order. On each stage of the construction we maintain a set of pivots P_i. Let $x \in S$ and let $NN(x, P_i)$ be the nearest neighbor of x among the points of P_i. For a pivot $p \in P_i$, define $\Gamma_i(p) = \{x \mid x \in S, NN(x, P_i) = p, r(x) \geq |xp|\}$, that is, all the points whose nearest neighbor from P_i is p which can transmit to p.

When considering the edge (x,y), the algorithm acts according to the following rule: If $NN(x, P_i) > \beta M_{i+1}$ then x is added to P_i and the edge (x,y) is added to $E_{\text{SP}}^{\text{DIR}}$. If $NN(x, P_i) \leq \beta M_{i+1}$ and there is no edge $(x', y) \in E_{\text{SP}}^{\text{DIR}}$ such that $x' \in \Gamma_i(NN(x, P_i))$ then the edge (x,y) is added to $E_{\text{SP}}^{\text{DIR}}$. When i reaches $\lfloor \log_{1+\alpha} M \rfloor$, the algorithm handles all the edges that belong to $E(M_{\lfloor \log_{1+\alpha} M \rfloor + 1}, M_{\lfloor \log_{1+\alpha} M \rfloor})$. This includes also edges whose weight is 1, the minimal possible weight.

The spanner construction algorithm is given in Figure 1. The algorithm returns the directed graph $H^{\text{DIR}}(S, E_{\text{SP}}^{\text{DIR}})$. In what follows we prove that $H^{\text{DIR}}(S, E_{\text{SP}}^{\text{DIR}})$ is a $(1+\epsilon)$-spanner with $O(n/\epsilon^{-d} \log M)$ edges of the directed graph $I(S,E)$.

2.1 The Stretch of the Spanner

We start by showing that the stretch of the graph $H^{\text{DIR}}(S, E_{\text{SP}}^{\text{DIR}})$ returned by the algorithm is $1+\epsilon$.

Lemma 1 (Stretch). *Let $\epsilon > 0$ and let $H^{DIR}(S, E_{SP}^{DIR})$ be the graph returned by Algorithm* **disk-spanner**. *If $(x,y) \in E$ then $\delta_G(x,y) \leq (1+\epsilon)|xy|$.*

Proof. Assume that the transmission ranges are scaled such that the shortest edge is of weight 1. Set $\alpha = \beta < \epsilon/6$. We prove that every directed edge of an arbitrary node $x \in S$ is approximated with $1+\epsilon$ stretch. Let $i \in [0, \lfloor \log_{1+\alpha} M \rfloor]$. The proof is by induction on i. For a given node x, the base of the induction is the maximal value of i in which x has an edge in $E(M_{i+1}, M_i)$. Let j be this value for x, that is, the set $E(M_{j+1}, M_j)$ contains the shortest edge that touches x. Every other node is at distance at least M_{j+1} away from x, hence x

```
Algorithm disk-spanner (I(S, E), )
E_SP^DIR ←
P_0 ←
for i ← 0 to ⌊log_{1+α} M⌋
    for each (x, y) ∈ E(M_{i+1}, M_i) do
        if |NN(x, P_i)x| > M_{i+1} then
            P_i ← P_i ∪ {x}
        if ∄(x', y) ∈ E_SP^DIR s.t. x' ∈ _i(NN(x, P_i))
            E_SP^DIR ← E_SP^DIR ∪ {(x, y)}
    P_{i+1} ← P_i
return H^DIR(S, E_SP^DIR)
```

Fig. 1. A high level implementation of the spanner construction algorithm for *general* disk graphs

is a pivot at this stage and every edge that touches x from the set $E(M_{j+1}, M_j)$ is added to $E_{\mathrm{SP}}^{\mathrm{DIR}}$.

We now turn to prove the induction hypothesis. Let $(x, y) \in E(M_{i+1}, M_i)$ for some $i < j$ and let $p = NN(x, P_i)$. If the edge (x, y) is not in the spanner, then there must be an edge $(\hat{x}, y) \in E_{\mathrm{SP}}^{\mathrm{DIR}}$, where $\hat{x} \in \Gamma_i(p)$. The crucial observation is that x has a transmission range of at least M_{i+1}. It follows from the algorithm that $|\hat{x}p| \leq \beta M_{i+1}$ and $|xp| \leq \beta M_{i+1}$.

By the choice of β, it follows that $2\beta M_{i+1} < M_{i+1}$ and $(x, \hat{x}) \in E$. Thus, there is a (directed) path from x to y of the form $\langle x, \hat{x}, y \rangle$ whose length is $2\beta M_{i+1} + M_i$. However, only the edge $(\hat{x}, y)$ is in $E_{\mathrm{SP}}^{\mathrm{DIR}}$. By the inductive hypothesis, the edge $(x, \hat{x})$ whose weight is $2\beta M_{i+1}$ is approximated with $1 + \epsilon$ stretch. Thus, there is a path in the spanner from x to y whose length is at most $(1 + \epsilon)|x\hat{x}| + M_i$, and this can be bounded by

$$(1 + \epsilon)2\beta M_{i+1} + M_i = ((1 + \epsilon)2\beta + (1 + \alpha))M_{i+1}.$$

As the edge $(x, y) \in E(M_{i+1}, M_i)$ it follows that $|xy| \geq M_{i+1}$. It remains to prove that $1 + 2\epsilon\beta + 2\beta + \alpha \leq 1 + \epsilon$, which follows directly from the choice of α and β.

2.2 The Size of the Spanner

We now prove that the size of the spanner $H^{\mathrm{DIR}}(S, E_{\mathrm{SP}}^{\mathrm{DIR}})$ is $O(n/\epsilon^d \log M)$. As a first step, we state the following well-known lemma, cf. [9].

Lemma 2. *[Packing Lemma] If all points in a set $U \in \mathbb{R}^d$ are at least r apart from each other, then there are at most $(2R/r + 1)^d$ points in U within any ball X of radius R.*

The next lemma establishes a bound on the number of incoming spanner edges that a point may be assigned on stage $i \in [0, \lfloor \log_{1+\alpha} M \rfloor]$ of the algorithm.

Lemma 3. *Let $i \in [0, \lfloor \log_{1+\alpha} M \rfloor]$ and let $y \in S$. The total number of incoming edges of y that were added to the spanner on stage i is $O(\epsilon^{-d})$.*

Proof. Let (x, y) be a spanner edge and let $NN(x, P_i) = p$. We associate (x, y) to p. From the spanner construction algorithm it follows that this is the only incoming edge of y whose source is in $\Gamma_i(p)$. Thus, this is the only incoming edge of y which is associated to p. Now consider all the incoming edges of y on stage i. The source of each of these edges is associated to a unique pivot within distance of at most $M_i + \beta M_{i+1}$ away from y and any two pivots are βM_{i+1} apart from each other. Using Lemma 2, we get that the number of edges entering y is $(\frac{M_i + \beta M_{i+1}}{\beta M_{i+1}} + 1)^d = ((1+\alpha)/\beta + 2)^d = O(\epsilon^{-d})$.

It follows from the above lemma that the total number of edges that were added to $E_{\mathrm{SP}}^{\mathrm{DIR}}$ in the main loop is $O(n/\epsilon^d \log M)$.

2.3 The Construction Time

We now describe how to efficiently implement the algorithm. Let n be the number of vertices and let m be the number of edges in the disk graph $I(S, E)$.

First, the algorithm has to partition the set E into the sets $E(M_{\lfloor \log_{1+\alpha} M \rfloor + 1}, M_{\lfloor \log_{1+\alpha} M \rfloor}), \ldots, E(M_1, M_0)$. This can be done in $O(m)$ time. The algorithm also preforms nearest neighbor queries. It is easy to see that at most $O(m)$ such queries are processed. To obtain an efficient implementation we maintain the set P_i using the dynamic nearest neighbor data structure of Cole and Gottlieb [6]. Every operation is supported in $O(\log n)$ time. However, their data structure is only capable of answering ϵ-approximate nearest neighbor queries. Luckily, it is enough for our purpose. The only effect of using an approximation is that the separation between any two pivots becomes $(1 + \epsilon')\beta M_{i+1}$ for some arbitrarily small $\epsilon' > 0$, instead of βM_{i+1}, which has a negligible effect on our bounds.

Any new pivot is inserted into the data structure in $O(\log n)$ time. The set of pivots on the $(i+1)$st stage is initiated with the set of pivots of the ith stage. Thus, any point is inserted exactly once into that data structure.

By the above discussion it follows that the total cost of the construction algorithm is $O(m \log n)$.

2.4 A Localized Algorithm

We now turn to describe a localized implementation of the algorithm. We assume a synchronous model in which a unique id is assigned to every node and that any node knows the id's of its outgoing neighbors (i.e., the nodes it can reach).

Similarly to the centralized algorithm, the localized algorithm of every node u has a main loop and in each iteration of the main loop the pivots of the current level are chosen by a simple adaption of the standard distributed algorithm for finding a maximal independent set (cf. Peleg [18]; chapter 8). More specifically, let $N_i(u)$ be the set of nodes at distance at most βM_{i+1} from u whose transmission radius is at least M_{i+1}, where u is the node currently running the algorithm.

```
Algorithm local-disk-spanner (code for node u)
for i ← 0 to ⌊log_{1+α} M⌋
    v ← extract-min (N_i(u))
    if id(v) > id(u)
      E_SP^DIR ←
      obtain E^v(M_{i+1}, M_i) from every v ∈ N_i(u)
      let Ê ← (∪_{v∈N_i(u)} E^v(M_{i+1}, M_i)) ∪ E^u(M_{i+1}, M_i)
      for every (x, y) ∈ Ê do
          if ∄(x', y) ∈ E_SP^DIR s.t. x' ∈ N_i(u)
            E_SP^DIR ← E_SP^DIR ∪ {(x, y)}
            send (x, y) to x
```

Fig. 2. A localized spanner construction algorithm for *general* disk graphs

If the graph was undirected then this set could be obtained easily. However, in the directed case u may have neighbors whose transmission range it too small, and thus should not be in $N_i(u)$. By a simple procedure we can overcome this problem without adding any additional assumption to our model. Notice that every node in $N_i(u)$ can transmit to u, thus, u can broadcast a message within its transmission range and every neighbor that gets the message returns an acknowledgment to u if u is within its transmission range. By this procedure, u can find its neighbors that can transmit to it and this is the only information that is needed in order to form the set $N_i(u)$.

The pivot selection is done as follows. The node v with minimal id in $N_i(u)$ is extracted from $N_i(u)$ and if u's id is smaller than v's then u marks itself as a pivot in level i. If u is not a pivot then nothing further is done. However, if u is a pivot then it performs a centralized computation of the spanner edges emanating from nodes of $N_i(u)$, and informs these nodes. For a node v, let $E^v(M_{i+1}, M_i)$ denotes the set of edges of $E(M_{i+1}, M_i)$ emanating from v. The edges that emanate from u and from the nodes of $N_i(u)$ are scanned in a non-decreasing order of length and an edge (x, y) is added to the spanner if and only if it is the first edge from a vertex of $\{u\} \cup N_i(u)$ to y. The algorithm is formally given in Figure 2. Next, we show that the message complexity is linear in the number of edges of the disk graph.

Lemma 4. *The message complexity of the localized algorithm is* $O(m + n/\epsilon^d)$.

Proof. In the i-th iteration every node v sends its edge set $E^v(M_{i+1}, M_i)$ to every pivot u where $v \in N_i(u)$. From packing arguments it follows that there is a constant number of pivots that have v in their close neighbor set. Thus, every edge of $E^v(M_{i+1}, M_i)$ is passed to a constant number of pivots and in total v generates $O(deg(v))$ messages. When a pivot computes the spanner edges it sends messages only to points that have to maintain a link that corresponds to a spanner edge. The total number of such messages is simply the number of spanner edges which is $O(n/\epsilon^d)$.

Table 1. Stretch 2

Region	Max Radius	Points	Edges	Removed Edges	Required Stretch	Savings
10×10	16	50	1565	343	2	0.22
15×15	20	100	5769	1878	2	0.33
25×25	25	200	18145	6999	2	0.39
30×30	35	500	133752	81916	2	0.61

2.5 Topology Updates

A fundamental question in topology control is what will happen when the underlined communication graph is being changed. For example, points are removed from the network or new points are added.

In our case it is easy to see that a deletion or an insertion of one point may remove or add many links which are essential to the connectivity of the network and thus must be in the spanner without considering the distances. As a result of that at the worse-case it may take $\Omega(n)$ time to update the spanner. Deleting and inserting the point u causes to update cost which is proportional to the number of points with small transmission range that are within the transmission range of u.

2.6 Simulations

We have implemented our spanner construction algorithm and tested it on randomly generated disk graphs. The graphs are generated by picking random points in a region of predefined size. Each point is also assigned a random transmission range from a predefined interval. A disk graph is then created by adding an edge from a point p to q if q is within the transmission radius of p. We have constructed spanners with required stretch factors of 2 and 3. Given a region size and a maximal radius, 100 different graphs were generated and the results

Table 2. Stretch 3

Region	Max Radius	Points	Edges	Removed Edges	Required Stretch	Savings
10×10	16	50	1553	697	3	0.45
15×15	20	100	5813	3336	3	0.57
25×25	25	200	18304	11725	3	0.64
30×30	35	500	134203	108331	3	0.81

were averaged over all these graphs. The results are summarized in Table 1 and Table 2. A careful look at the spanner construction reveals that the average degree of a node in the spanner is at most $25 \log M$. As the results indicate (and as one may expect), when the random disk graph becomes denser, then the spanner obtains a better compression rate. The average degree of a node in the random disk graph is reduced by half or more in most of the cases and the resulted spanner has an average degree which is less than $25 \log M$. It implies that there are many natural instances on which better bounds then the worse case bound can be obtained by our spanner construction algorithm.

3 A $(1+\ )$-Spanner for Unit Disk Graphs

In this section we show how to compute a $(1+\epsilon)$-spanner for a unit disk graph. More specifically, we show that given a set of points S any $(1+\epsilon)$ geometric spanner of S can be turned into $(1+\epsilon')$-spanner of the unit disk graph of S.

Let $H(S, E_{\text{SP}})$ be a geometric $(1+\epsilon)$-spanner of S and let $I(S, E)$ be the unit disk graph of S. The following lemma shows that the distances induced by the graph $I(S, E)$ are approximated with a stretch factor of $1+\epsilon$ in $H(S, E_{\text{SP}})$.

Lemma 5. *Let S be a set of points and let $H(S, E_{SP})$ be any $(1+\epsilon)$-spanner of S. If $I(S, E)$ is the unit disk graph of S then $\delta_H(p, q) \leq (1+\epsilon)\delta_I(p, q)$ for every pair of points $p, q \in S$.*

Proof. Let $p, q \in S$ and let $p = x_1, x_2, \ldots, x_\ell = q$ be the vertices on a shortest path between p and q in $I(S, E)$. By the definition of H, $\delta_H(x_i, x_{i+1}) \leq (1+\epsilon)|x_i x_{i+1}|$. Thus, $\delta_H(p, q) \leq (1+\epsilon)\sum_{i=1}^{\ell-1} |x_i x_{i+1}| = (1+\epsilon)\delta_I(p, q)$.

The above lemma states that for any pair of points there exists a path in H that approximates the shortest path between them in the unit disk graph I. However, H is not necessarily a *spanner* of I, as it might have edges that are not included in I, while a spanner must be a subgraph of the original graph. At first glance it might seem that a possible solution to this problem is to remove every edge whose length is strictly greater than 1 from H. Indeed, by doing so we ensure that the resulting graph is a subgraph of I. However, it might no longer be a $(1+\epsilon)$-spanner for I. In particular, consider two points p and q such that $|pq| = 1$. It might so happen that the path σ that approximates this distance in G is composed of two edges, (p, r) and (r, q), where $|pr| = 1 + \epsilon/2$ and $|rq| = \epsilon/2$. In such a situation, if all edges whose weight is greater than 1 are removed from H, then the path σ is disconnected.

Our solution to this problem is as follows. Starting from a $(1+\epsilon)$ geometric spanner $H(S, E_{\text{SP}})$ of S, every edge whose length is in the range $(1, 1+\epsilon]$ is removed from E_{SP}. In compensation, for any removed edge we add, if possible, at most three replacement edges. Each of these three new edges belongs to the unit disk graph and their total length is at most $1 + 2\epsilon$.

Specifically, let (x, y) be an edge whose weight is in the range $(1, 1+\epsilon]$. We look for a pair of points u and v such that $|xu| \leq \epsilon$, $|vy| \leq \epsilon$ and $(u, v) \in E$. If such

```
Algorithm unit-disk-spanner (I(S,E), )
H(S,E_SP) ← geom-spanner(S, )
E_SP^UDG ← E_SP
for every (x,y) ∈ E_SP^UDG do
    if |xy| > 1 then
       E_SP^UDG ← E_SP^UDG \ {(x,y)}
    if |xy| ∈ (1,1+ ] then
       if ∃(u,v) ∈ E s.t. |xu| ≤  ∧ |vy| ≤
          E_SP^UDG ← E_SP^UDG ∪ {(x,u),(u,v),(v,y)}
return H^UDG(S,E_SP^UDG)
```

Fig. 3. A high level implementation of the spanner construction algorithm for *unit* disk graphs

a pair of points exists, we add the edges (x, u), (u, v) and (v, y) to the spanner instead of the edge (x, y). Such a situation is depicted in Figure 4. Notice that it might be that $u = x$ or $v = y$. If no such pair of points u and v is found, then nothing is done. Denote the resulting spanner by $H^{\text{UDG}}(S, E_{\text{SP}}^{\text{UDG}})$. The algorithm is given in Figure 3.

3.1 The Properties of the Spanner

In this section we show that the unit disk graph spanner constructed by our algorithm has the same properties as a regular geometric spanner.

Lemma 6. *The graph $H^{UDG}(S, E_{SP}^{UDG})$ constructed by* **unit-disk-spanner** *Algorithm is a $(1+\epsilon)$-spanner of $I(S, E)$ with $O(n/\epsilon^d)$ edges.*

Proof. It is easy to see that $E_{\text{SP}}^{\text{UDG}} \subseteq E$, as every edge of $E_{\text{SP}}^{\text{UDG}}$ is of weight at most 1. From Lemma 5 it follows that every edge of $I(S, E)$ is approximated by the geometric spanner. The removal of an edge whose weight is strictly greater than $1+\epsilon$ has no effect on the approximation of edges of the unit disk graph, since these edges are of weight 1 or less, so edges of weight greater than $1 + \epsilon$ do not participate in approximating them. When an edge whose weight is in the range $(1, 1+\epsilon]$ is replaced with a path of length at most $1+2\epsilon$, only the approximation factor is affected, increasing from $1 + \epsilon$ to at most $(1 + \epsilon)(1 + 2\epsilon) \leq 1 + 5\epsilon$. It remains to show that if a removed edge whose weight is from the range $(1, 1+\epsilon]$ has no replacement path, then its removal is harmless, i.e., there is no edge in the unit disk graph whose approximation is affected. Consider such a removed edge (x, y), and assume that there is no edge $(u, v) \in E$ such that $|xu| \leq \epsilon$ and $|vy| \leq \epsilon$. It follows that for every edge in the unit disk graph, at least one of its endpoints is at distance strictly greater than ϵ from both x and y. Thus, the edge (x, y) cannot be used in the approximation of any edge of the unit disk graph and it can be removed without effecting the approximation factor.

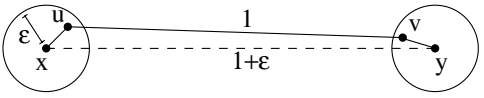


Fig. 4. A geometric spanner edge and its possible replacement path

The size of $E_{\mathrm{SP}}^{\mathrm{UDG}}$ remains $O(n/\epsilon^d)$, as at most three edges are added for any removed edge.

3.2 The Construction Time

In this section we explain how to efficiently implement the algorithm when the set of points is in the plane. For every edge of the geometric spanner whose weight is in the range $(1, 1+\epsilon]$, we need to check whether a replacement path exists. For every $p \in S$, we create a nearest neighbor data structure for the points within a radius of ϵ around p (including the point itself). The cost for that is at most $O(\deg(p) \log \deg(p))$, where $\deg(p)$ is the degree of p in $I(S, E)$ (See, [8,5]). Queries can be answered in $O(\log \deg(p))$ time. Given an edge $(x, y) \in E_{\mathrm{SP}}^{\mathrm{UDG}}$ whose weight is in the range $(1, 1+\epsilon]$, we scan all the edges of length at most ϵ that touch x in $I(S, E)$. For each such edge (x, u), the algorithm queries the data structure of y to find the closest point to u among the points within distance ϵ from y. If the closest point is at distance 1 or less, then we have found the replacement path. If not, then we proceed to the next edge of x. The cost of this search is $O(\deg(x) \log \deg(y))$. This is done for every geometric spanner edge whose weight is in the range $(1, 1+\epsilon]$. A problem may arise if a point with large degree in $I(S, E)$ also has many spanner edges. To avoid that, we use a geometric spanner of bounded degree [4,7], that is, one where every point has $O(\epsilon^{-d})$ spanner edges. Hence every point will take part in $O(\epsilon^{-d})$ tests, each of cost proportional to its degree. The total running time is thus $O(m \log n)$.

The next theorem summarizes the above arguments.

Theorem 1. *Let S be a set of n points in the plane. Let $I(S, E)$ be the unit disk graph that corresponds to S, where $|E| = m$. There exists a $(1+\epsilon)$ spanner of $I(S, E)$ with $O(n/\epsilon)$ edges that can be constructed in $O(m \log n)$ time.*

4 Concluding Remarks

We have presented in this paper two constructions. The first and most important is the first construction ever of spanners for disk graphs. This result raises many other questions, both practical and theoretical. From the perspective of routing it is interesting to use this construction as a topology for greedy based routing algorithms in ad-hoc networks. Our spanner construction allows routing in ad-hoc networks with variable transmission radii. It is also interesting to consider the question of whether efficient compact routing schemes exhibiting a tradeoff between the space usage of each node and the stretch of the paths exist for the model of disk graphs. From a theoretical perspective it is interesting to explore which other natural classes of directed graphs have good spanners.

References

1. Bose, P., Morin, P.: Online routing in triangulations. In: Aggarwal, A.K., Pandu Rangan, C. (eds.) ISAAC 1999. LNCS, vol. 1741, pp. 113–122. Springer, Heidelberg (1999)
2. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks 7(6), 609–616 (2001)
3. Callahan, P.B., Kosaraju, S.R.: A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. J. ACM 42, 67–90 (1995)
4. Chan, T.-H., Gupta, A., Maggs, B.M., Zhou, S.: On hierarchical routing in doubling metrics. In: Proc. 16th ACM-SIAM Symp. on Discrete Algorithms, pp. 762–771 (2005)
5. Chan, T., Patrascu, M.: Point location in sublogarithmic time and other transdichotomous results in computational geometry. In: Proc. 47th IEEE Symp. on Foundations of Computer Science, pp. 325–332 (2006)
6. Cole, R., Gottlieb, L.: Searching dynamic point sets in spaces with bounded doubling dimension. In: Proc. 38th ACM Symp. on Theory of Computing (2006)
7. Das, G., Naraimhan, G., Salowe, J.: A new way to weigh malnourished Euclidean graphs. In: Proc. 6th ACM-SIAM Symp. on Discrete Algorithms, pp. 215–222 (1995)
8. Dobkin, D.P., Lipton, R.J.: Multidimensional searching problems. SIAM J. Comput. 5(2), 181–186 (1976)
9. Gao, J., Guibas, L., Nguyen, A.: Deformable spanners and applications. In: Proc. 20th ACM Symp. on Computational Geometry, pp. 179–199 (2004)
10. Gao, J., Guibas, L.J., Hershberger, J., Zhang, L., Zhu, A.: Geometric spanners for routing in mobile networks. IEEE J. on Selected Areas in Communications 23(1), 174–185 (2005)
11. Gao, J., Guibas, L.J., Nguyen, A.: Distributed proximity maintenance in ad hoc mobile networks. In: Proc. IEEE Conf. on Distributed Computing in Sensor Systems, June 2005, pp. 4–19 (2005)
12. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proc. 6th Conf. on Mobile computing and networking, pp. 243–254 (2000)
13. Kuhn, F., Zollinger, A.: Ad-hoc networks beyond unit disk graphs. In: DIALM-POMC 2003: Proceedings of the 2003 joint workshop on Foundations of mobile computing, pp. 69–78. ACM Press, New York (2003)
14. Li, X.-Y., Calinescu, G., Wan, P.-J., Wang, Y.: Localized delaunay triangulation with application in ad hoc wireless networks. IEEE Trans. on Parallel and Distributed Systems 14(10), 1035–1047 (2003)
15. Li, X.-Y., Song, W.-Z., Wang, Y.: Localized topology control for heterogeneous wireless sensor networks. ACM Transactions on Sensor Networks 2(1), 129–153 (2006)
16. Narasimhan, G., Smid, M.: Geometric Spanner Networks. Cambridge University Press, Cambridge (2007)
17. Onus, M., Richa, A.: Efficient broadcasting and gathering in wireless ad-hoc networks. In: ISPAN 2005 (2005)
18. Peleg, D.: Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)
19. Peleg, D., Schäffer, A.A.: Graph spanners. J. Graph Theory 13, 99–116 (1989)

20. Rajaraman, R.: Topology control and routing in ad hoc networks: a survey. SIGACT News 33(2), 60–73 (2002)
21. Salowe, J.S.: Constructing multidimensional spanner graphs. Int. J. Comput. Geometry Appl. 1(2), 99–107 (1991)
22. Stojmenovic, I., Lin, X.: Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Trans. Parallel Distrib. Syst. 12(10), 1023–1032 (2001)
23. Vaidya, P.M.: A sparse graph almost as good as the complete graph on points in K dimensions. Discrete & Computational Geometry 6, 369–381 (1991)
24. Wang, Y., Li, X.-Y.: Efficient delaunay-based localized routing for wireless sensor networks. Int. J. of Communication Systems 20(7), 767–789 (2006)
25. Wang, Y., Li, X.-Y.: Localized construction of bounded degree and planar spanner for wireless ad hoc networks. MONET 11(2), 161–175 (2006)
26. Yao, A.C.-C.: On constructing minimum spanning trees in k-dimensional spaces and related problems. SIAM J. Comput. 11(4), 721–736 (1982)

Geographic Routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks

Luminita Moraru[1,⋆], Pierre Leone[1], Sotiris Nikoletseas[2], and Jose Rolim[1]

[1] Computer Science Department
University of Geneva
1211 Geneva 4, Switzerland
[2] University of Patras and CTI
26500 Patras, Greece

Abstract. Existing geographic routing algorithms for sensor networks are mainly concerned with finding a path toward a destination, without explicitly addressing the impact of obstacles on the routing performance. When the size of the communication voids is increased, they might not scale well with respect to the quality of paths, measured in terms of hop count and path length.

This paper introduces a routing algorithm with early obstacle detection and avoidance. The routing decisions are based on path optimality evaluation, made at the node level, gradually over time. We implement our algorithm and evaluate different aspects: message delivery performance, topology control overhead and algorithm convergence time. The simulation findings demonstrate that our algorithm manages to improve significantly and quite fast the path quality while keeping the computational complexity and message overhead low. The algorithm is fully distributed, and uses only limited local network knowledge.

1 Introduction

Geographic routing algorithms represent one of the most suitable solution for routing within sensor networks, mainly due to their stateless nature. The path is built only with information about the one hop neighbors and of the destination, thus they require negligible memory at sensor nodes - a direct consequence is network scalability - no additional topology control traffic is needed when the network changes.

The simplest *geographic routing* strategy, *greedy*, chooses for forwarding the neighbor closest to the destination [3],[13],[17]. But it has a main drawback, called the local maximum phenomenon: when the current node has no neighbor closer to the destination then itself, the delivery of the message fails. This is often the case if there is an obstacle or a void in the network, or in low density network areas.

⋆ Research partially funded by the Seventh Framework Project FRONTS (contract number 215270) of the Prevasive Adaptation Proactive Initiative of IST/FET.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 148–161, 2008.
© Springer-Verlag Berlin Heidelberg 2008

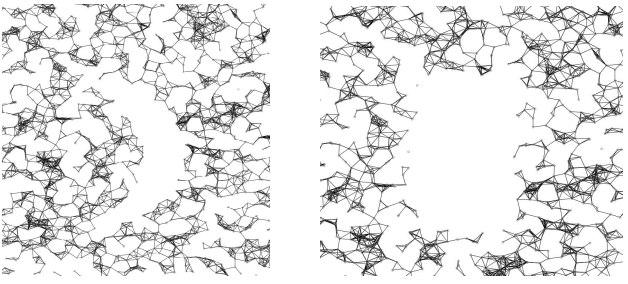


Fig. 1. Communication Voids

The solution to this problem is a recovery mode, an alternative routing method with guaranteed delivery, used when greedy fails. Several classes of algorithms have been proposed for this purpose. Further we will discuss the class of memoryless recovery mechanisms, *perimeter* routing, based on planar graph traversal techniques. The algorithms in this class work only on planar graphs, thus before entering this mode, a planar subgraph of the initial graph must be available. The basic idea behind this algorithms is as follows: a message is forwarded clockwise along a face of a planar graph. When it reaches a link that intersects the line between the source and the destination, it switches to the adjoining face. A message will leave the *perimeter* mode when it will find a node closer to the destination than the perimeter entry point.

Geographic routing algorithms scale well with respect to effectiveness of the path when the size of the communication voids is varied. But these paths are not optimal in terms of length, and in fact they might be quite long, thus inefficient. This is due mainly to the nature of the protocol used during the rescue mode: *perimeter* routing. It will choose sometimes relays that are further away from the destination than the current node. Additionally, it requires graph planarity, and the planarization process preserves the shortest links, thus increasing the hop count.

The complexity of obstacle avoidance problem is influenced as well by the shape of the obstacles. Difficulties appear mainly in avoiding concave obstacles (see Fig 1(a)). Even if we consider only the case of convex obstacles (see Fig 1(b)), an important constraint remains: nodes should exploit only local information.

In this paper we consider the behavior of *geographic routing* algorithms within network configurations with obstacles and local irregularities. Our contribution is to identify the presence of the object early on the routing path and redirect the messages on a shorter path as soon as possible. The strategy we are proposing is as follows: during message forwarding, each node evaluates the optimality of the paths that go through it. The node tags itself based on the outcome of the node optimality evaluation method - the evaluation is positive if a node has at least one neighbor tagged as optimal closer to the destination then itself. If a node is non-optimal, than we consider that any path toward the destination using it will be as well non-optimal.

Subsequent message forwarding decisions will analyse first the suitability of optimal nodes when choosing the relays. If no optimal node is suitable (e.g. no

neighbor closer to the destination than the current node is optimal), then a non optimal node is used.

When obstacles are present, the consequences of our method are the tagging of the nodes in the vicinity of the object as non optimal, and the early redirection of the message toward the edge of the object, resulting in a significant decrease of the path length. The cost is a small overhead, depending on obstacle size and shape, (independent of the network size) and paid only once.

2 State of the Art and Comparison

We address the problem of early detection and avoidance of obstacles in geographic routing algorithms. Although several geographic routing with obstacles avoidance techniques were proposed so far, most of them are concerned mainly in guaranteeing the delivery: finding some path when greedy forwarding is not possible. Moreover, there are situations where the constraints like the stateless nature (i.e. the low memory needed) of geographic routing, are in contrast to the quantity of data they need to make a decision. Further we will introduce the techniques with guaranteed data delivery, outlining their characteristics and drawbacks. The solutions are divided in the following categories, as described in [4]: *planar graph based, geometric obstacle detection, cost based, flood based, and hybrid*.

Planar graph based obstacle avoidance techniques, [1],[6],[9],[11], are used since they were proved to guarantee delivery if a path exists. In the initial stage, these strategies use greedy. When a node has no neighbor closer to the destination, greedy is replaced by one of existing planar graph traversal algorithms [13],[14],[15],[19],[22]. Since the representation of the network is not always a planar graph, this class of strategies uses a distributed planarization algorithm, like those proposed in[8],[12],[21]. The performances of these strategies depend on two factors: the graph traversal and the distributed planarization algorithms. Nevertheless, most of the algorithms are concerned with improving the planar graph traversal algorithms while ignoring the optimality of the path. Still, the gain in path length (compared with the optimal path) becomes significant when obstacles are present and it is proportional with their size.

An optimality evaluation method is described in [18]. It can be built on top of any method based on planar graph traversal. Each node keeps track of the ratio between greedy decisions and the total number of routing decisions. If the ratio is higher than a specific threshold, then the node is considered as being optimal. The main drawback of this method, is that the optimality of the path does not depend on the network topology only, this way failing to correctly evaluate some of the nodes.

Geometric obstacle detection is proposed in [7]. It uses the geometric properties of a node to determine if a message can be stuck at that node. An algorithm is developed to find holes in the network, defined as areas of the network bounded by the stuck nodes. The disadvantage of this technique is the high complexity

of the detection of the holes. Additionally, it does not guarantee delivery when the destination is inside the hole.

Cost based approach [5] consists in assigning a cost to each node, proportional to the distance to the destination. When greedy forwarding fails, a node will forward a packet to a neighbor with a lower cost than itself. Although the complexity and the overhead of the algorithm is rather medium, it does not choose optimal paths. *Flooding based* techniques [20],[10] are using broadcast to forward the message, once a packet is stuck. Although the complexity is low, the overhead is high. They guarantee delivery, but path optimality is not a concern. Multipath techniques, like [16], [2], explore several paths toward the destination, to trade-off efficiency with fault tolerance. Similar with the case of flooding techniques, the overhead may be high. *Hybrid* techniques use at least a combination of two obstacle avoidance methods. The motivation is the improved efficiency of the path and the guaranteed delivery of the message. They are used when only one of the two techniques is not enough to achieve these requirements. The disadvantage is the increased overall complexity.

The methods described above are mainly concerned with guaranteeing delivery. In contrast, we aim at providing high quality paths, by keeping track of previous evaluations in a distributed manner. Additionally, our technique preserves the properties of the network, like scalability and low complexity since it works only with local information about the direct neighbors of the node currently propagating data.

3 Non Optimality Evaluation Methods

The algorithm presented in this paper is part of a class of algorithms based on non optimal nodes detection. It will be presented in parallel with the previous work in the same area. In each case we propose a different method for the detection of non-optimal nodes. We define a node as *non optimal* if any message using the node as a relay will eventually use rescue mode to reach the destination. A *non optimal* path between a source and the destination is a path containing at least one non optimal node.

3.1 Behavior Based Tagging (BBT)

In [18], the optimality of a node is evaluated as follows: if a node uses greedy forwarding, then a positive counter is incremented, if perimeter mode is used, then a negative counter is incremented. A node is considered on an optimal path if the ratio between the greedy decisions and the total number of decisions is higher than a specified threshold.

The routing algorithm will consider the result of the evaluation of the nodes while selecting the relays for a message. When a message is routed in the greedy mode, the node will first search for neighbors closer to the destination and marked as optimal. If no neighbor is found, it will switch to perimeter. When a message is routed in the perimeter mode, the current relay will switch back to

greedy if it finds a node closer to the destination than the perimeter entry point, otherwise it will continue in the perimeter mode.

The behavior of this method is shown in the examples in Fig 2(a), 3(a), 4(a), and it will be discussed in the next subsection. The drawback of this approach is the wrong evaluation of some nodes as non optimal, due to the influence of the position of the perimeter entry point on the routing mode used at each node (this behaviour will be explained in more details in the next subsection). Therefore, a more precise evaluation method is needed.

3.2 Neighborhood Based Tagging (NBT)

The evaluation method is as follows: a node will mark itself as non-optimal toward a certain direction if it does not have optimal neighbors (or does not have neighbors at all) toward that direction. The impact of this method on the network is the apparition of a marked convex region along some of the faces of the object. Further, we will give a formal definition of non optimal nodes.

Let $G = (N, E)$ be a graph representation of the network, where N represents the set of nodes and E the set of links. We select $n_k \in N$ a random node in the network and d the sink receiving all the messages. Let $S_k = \{n_i | (n_k, n_i) \in E\}$ be the set of one hop neighbors and $S'_k = \{n_i | n_i \in S_k \wedge dist(n_i, d) < dist(n_k, d)\}$. If $M \subset N$ is the set of non optimal nodes in the network, then $n_k \in M$ if $S'_k \cap M = S'_k$.

Algorithm 1. Optimality Evaluation Method

```
this.setProperty(optimality,'NON-OPTIMAL')
for all n_i in S do
  if this.closer(D, n_i) and n_i.getProperty(optimality) == 'OPTIMAL' then
    this.setProperty(optimality,'OPTIMAL')
    break
  end if
end for
```

The pseudocode of the algorithm is presented herein. Algorithm 1 describes the optimality evaluation method. *this* refers to the node making the evaluation. Algorithm 2 describes the routing strategy that includes non optimality of the nodes for path evaluation.

We define the marked area as the area in the vicinity of the object containing nodes tagged as non optimal. The unmarked area is represented by the rest of the network. The influence of optimality tag on routing decisions is as follows:

- Unmarked area: the behaviour of the routing protocol remains unchanged. Once a node in the marked area, it will use greedy to get to the destination. Once there are no closer neighbors, the node uses perimeter.
- Border: The routing protocol tries to avoid the entry into the marked area. Therefore, for a message in the greedy mode, a node will search first a neighbor, closer to the destination than itself, between the optimal nodes. If it

fails, it will start a new search considering the set of non optimal nodes, closer to the destination than itself.
- Marked area: similar with the unmarked area.

Algorithm 2. Optimality based Routing Strategy

```
if routing_mode is "perimeter" then
  next ← get_next_hop("perimeter", neighbors)
else
  selected_neighs ← filter_by_property(neighbors, optimality, 'OPTIMAL')
  next ← get_next_hop("greedy", selected_neighs)
  if ! ∃ next then
    next ← get_next_hop("greedy", neighbors \ selected_neighs)
    if ! ∃ next then
      next ← get_next_hop("perimeter", neighbors)
    end if
  end if
end if
evaluate_optimality
```

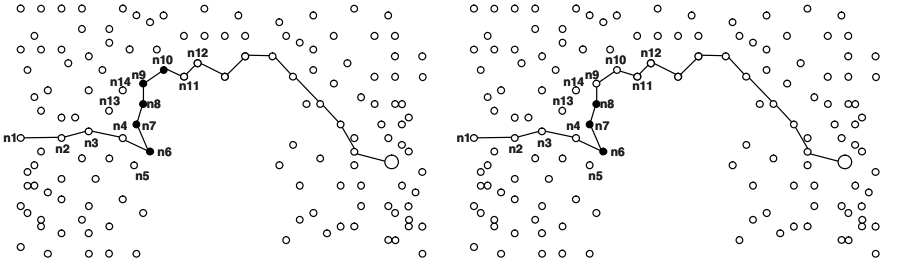
Our algorithmic design is aiming at the following improvements:

- *Smaller marked area* - there are nodes which have greedy neighbors toward the destination, but they are using perimeter routing since they are not closer to the destination than the perimeter entry point. The tagging method based on neighborhood will mark them as optimal, while the method based on behavior would have marked them as non optimal.
- *Shorter paths* - since greedy tries to route around the marked area, reducing this area will result in reducing the length of the path.
- *More accurate evaluation* of the optimality, since the dependence of the perimeter entry point and the position of the source is eliminated.

3.3 Example

An example of the behavior of the algorithm is presented in Fig 2, 3, 4. They show both the evaluation (tagging) and routing path chosen by the network during three transmitted messages. The evaluation is made progressively, during the routing tasks: each time a node has to make a routing decision, it checks the status of its neighbors.

Figure 2(a) shows the transmission of the first message. The message is originated at node $n1$. Each node from $n1$ to $n4$ has a greedy neighbour toward the destination. Node $n6$ has no greedy node toward the destination, therefore the algorithm switches to rescue mode, with $n6$ as the perimeter entry point. Since none of the nodes $n7 - n10$ is closer to the destination than $n6$, all these nodes will use perimeter mode. All the nodes $n6 - n10$ will increase their negative counter and will be evaluated as non-optimal. $n11$ is closer to the destination than $n6$, therefore the routing mode will be switched to greedy. Greedy mode



(a) Behavior based tagging (b) Neighborhood based tagging

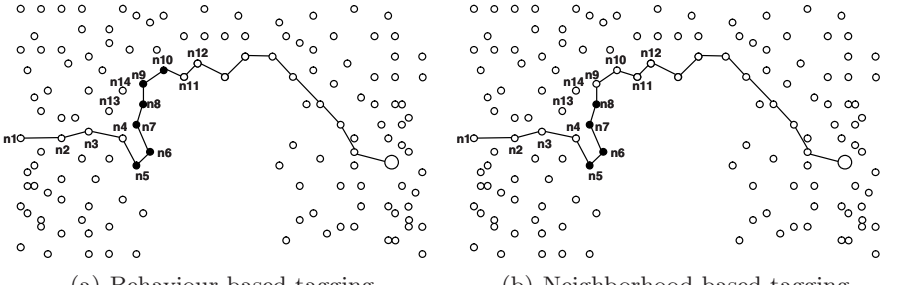
Fig. 2. The path of the first message

will be kept until the destination since all the remaining nodes on the path have neighbours closer to the destination than themselves.

Figure 2(b) shows the path of the same node when neighborhood based tagging is used. Nodes $n1 - n4$ have a neighbor closer to the destination than themselves. Therefore they are marked as optimal. Nodes $n6 - n8$ have no neighbor closer to the destination than themselves, therefore they are marked as non optimal. Starting from $n9$, the nodes are optimal again. Similar with Fig. 2(a), $n6$ is the perimeter entry point, and $n10$ is the perimeter exit point. At this step, neighborhood based tagging has no influence on the routing method.

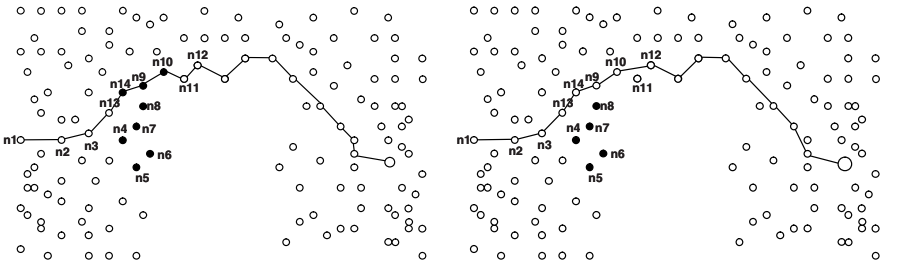
Figure 3 shows the path of the second message between the same source and destination. In both cases, $n4$ will choose the neighbor tagged as optimal and closer to the destination - $n5$. In Fig. 3(a), $n5$ will have no optimal neighbor closer to the destination, therefore it will start perimeter mode and increase the negative counter, becoming non optimal. In Fig. 3(b), $n5$ has no optimal neighbour closer to the destination and will tag itself as non optimal.

In Fig. 4 we will see the path of a message after a few other retransmissions. *NBT* finds an optimal path around the obstacle, while *BBT* will have some nodes marked as non-optimal on a path that could use only greedy forwarding towards the destination.



(a) Behaviour based tagging (b) Neighborhood based tagging

Fig. 3. The path of the second message



(a) Behavior based tagging (b) Neighborhood based tagging

Fig. 4. The path of the n-th message

4 Algorithm Analysis

Each node makes routing decisions based on the optimality of the neighbors. Therefore each node has to inform its neighbors about its current state. There are several options for transferring this information. First is by piggybacking it on the network control messages - periodic beacon messages, advertising their current status and position. This solution is suitable for the case of frequent state changes (i.e. behavior based routing).

The second option is to send an status update to the neighbors each time a node changes its state. This is suitable for a small number of node state changes, such is the case for neighborhood based routing. We will further show that for a static network, the state of node can switch at most once. Therefore, this option is more suitable for our case. We propose first a separation of nodes into layers, as follows:

- Layer 0: Nodes that have no greedy neighbors toward the destination: $L_0 = \{n_i | S'_i = \emptyset\}$
- Layer 1: Nodes that have greedy neighbors toward the destination only nodes of Layer 0: $L_1 = \{n_i | \forall n_k \in S'_i, n_k \in L_0\}$.
- Layer n: Nodes that have greedy neighbors toward the destination only nodes of Layers 0..n-1: $L_n = \{n_i | S'_i = \{n_k | n_k \in L_0 \cup L_1 \ldots \cup L_{n-1}\}\}$.

Proposition 1. *The Neighborhood Based Tagging Algorithm is stable: the tag of a node is switched only once.*

Proof. The status of a node $n_i \in L_0$ depends only on the network topology. If it is static, then the status of n_i once tagged as non-optimal, remains unchanged. The status of a node $n_i \in L_1$ depends only on its neighbors $n_k \in S'_i$, but $\forall n_k \in S'_i, n_k \in L_0$, therefore, once evaluated, their state will not change either. Similarly, the state of a node $n_i \in L_n$ depend only on $n_k \in L_0 \cup L_1 \ldots \cup L_{n-1}$, which are stable, therefore the nodes $n_i \in L_n$ are stable as well.

Another issue is the size of the tagged area. The total number of non optimal nodes depends only on the density and the topology of the network (the relative position of destination toward the object, and the size of the object). We

define the smallest density for which the the number of tagged nodes is both limited and proportional with the size of the marked area as the *critical density*. Experimentally, we found a critical density around 10.

For densities higher than critical density, the messages coming from sources for which exists a greedy path toward the destination, will generate the detection of a limited number of non optimal nodes before finding this greedy path that they will use afterwords, as shown in Fig. 4(b). Further we will prove that the algorithm preserves the greedy paths.

Theorem 1. *If there is a path* $P = n_0, n_1, ..., n_i$ *between a source* s *and a destination* d*, such that*

$$dist(n_i, d) > dist(n_{i-1}, d)...dist(n_2, d) > dist(n_1, d) > dist(n_0, d)$$

then no node $n_k \in P$ *is tagged as non-optimal.*

Proof. We proof the theorem by induction. The node n_0 is directly connected to the destination d, therefore it is *optimal*. The node n_1 has a neighbor closer to the destination, the node n_0, therefore it is *optimal*. We assume that the node n_{i-1} is optimal. Then n_i has an *optimal* neighbor toward the destination, therefore it is *optimal*.

Corollary 1. *If we can enclose the obstacle in a region such that for all the nodes outside this region it exists a greedy path toward the destination, then the marked region cannot exceed this region around the obstacle.*

In order to extend the suitability of the algorithm for any network configuration - nodes density smaller than the critical density, we redefine our algorithm by considering a new parameter during the optimality evaluation: the layer to which a node belongs, as defined at the begining of this section. We will shouw that the size of a layer is finite and if we limit the number of layers of marked nodes, then the algorithm is convergent to a stable state.

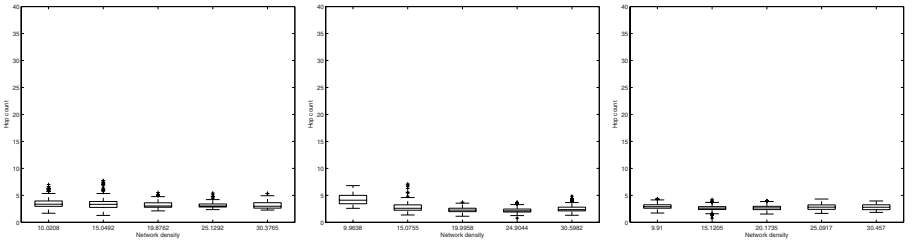
Proposition 2. *The size of a layer is finite.*

Proof. By induction on i.

Basis i=1 The size of Layer 0 is proportional with the object, therefore finite. A node in Layer 1 must have at least one greedy neighbor in Layer 0, it has to be in the transmission range of a node in Layer 0. Therefore the size of the Layer 1 is proportional with the size of Layer 0 and finite.

Inductive step. Suppose the size of Layers 0,1,2.. n-1 is finite. The nodes in Layer n have only greedy neighbors in one of the lower ranked layers. Therefore the size of the Layer n is finite.

The algorithm is convergent if the number of layers is finite. We can limit the number of layers by introducing a new parameter, a layer threshold. If a non optimal node is detected in a layer above this limit, then it will not switch its state. This will limit the evaluation to the nodes in the vicinity of the obstacle.



(a) GFG (b) Behavior based tagging (c) Neighbors based tagging

Fig. 5. Number of hops

5 Simulation Results

In this section we numerically validate the expected behavior and performance of our algorithms. The simulations we present compare our geographic routing algorithm and the well known greedy face greedy (GFG) algorithm which is considered a reference algorithm in the state of the art. Additionally we compare with a similar tagging based class of heuristic algorithms, described in [18].

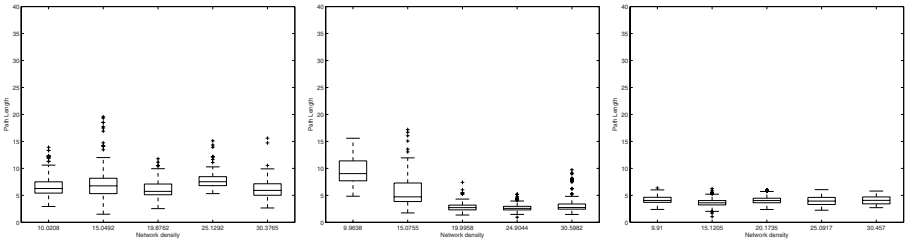
To make the comparison, the criteria we are interested in are (a) whether the tagging algorithm is convergent: whether the number of tagged nodes becomes constant after some time, (b) the total number of tagged nodes and (c) the performance in terms of path length and hop counts. The numerical experiments show that our algorithm competes well with the GFG and behavior based routing evaluation in terms of the total number of nodes on the routing paths, while reducing the number tagged nodes, thus the topology control traffic.

5.1 Details on the Experiments and the Representation of Results

The experiments are made with a network of nodes randomly distributed on a 200x200 units area. The size of the object (rectangulary shaped) is 30x50 units and the position of the upper left corner is 70x110. The transmission range of the nodes is constant, equal to 7 units, The total number of nodes varies between 2800 and 7900 such as to obtain different densities between 10 and 30.

For each step of the simulation, a new message is sent from a random source to a single destination (110,85), such that all the trajectories will intersect the object. The initial network setup is similar with Fig. 2. Within a step, a node that acts as a relay reads all the messages sent by its neighbors in the previous step and schedules them for retransmission within this step.

Each experiment is repeated 100 times with a different network topology, and the outcomes are presented in a box plot graphic. Box plots are composed of a box with the lower line being the lower quartile, the middle one the median and the upper one being the upper quartile of the sample. The dashed lines extending above and below the box show the span of the other samples. The plus sign represents outliers.



(a) GFG (b) Behavior based tagging (c) Neighbors based tagging

Fig. 6. Path length

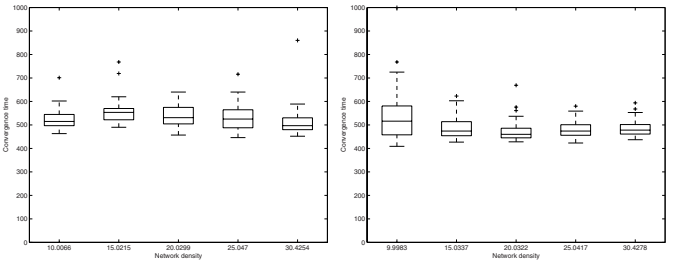
5.2 Performance Evaluation

The performances in terms of path length for the three algorithms are presented in Fig 6. We are evaluating the path stretch - defined as the ratio between the total path length of a message and the minimum euclidian distance between the source and the destination, while taking into account the presence of the obstacle.

For the smallest two densities considered, *BBT* has a major drawback: it performes worse than *GFG*. The reason is the influence of voids on the routing mode. Nodes are using perimeter routing due to the presence of the voids, therefore the size of the marked area will be increased by the lack of nodes, having as a consequence an increase of path lengths. For these densities, our protocol reduces with 50% the path stretch obtained by *BBT*. Therefore, we extend the suitablity of the early obstacle avoidance to a broader range of densities. It reduces for all densities the path stretch obtained by *GFG* with 30%. We extend the suitablity of the early obstacle avoidance to a broader range of densities. Still, *BBT* has slightly better performances for the highest densities: it has a decrease of 10% of the path stretch of *NBT* (but with 4 times more nodes marked).

Figure 5 shows the hops stretch of a message sent from a source to a destination. It is measured as the ratio between the number of hops of a message between the source and the destination, and the ideal number of hops (measured as the ratio between the euclidian path length described above and the transmission radius). The simulations show that for the lowest density *NBT* improves with 30% the performances of *BBT* and with 20% the performance of *GFG*.

We note that the overhead in our algorithm is independent of the network size. Thus our method scales well. Furthermore, additional messages are sent only once, i.e. the overhead is independent of the number of events generated in the network, while all messages routed around the obstacle benefit of smaller paths. Overall, the overhead impossed by tagging nodes is much less compared to the saving in routing messages. As an example, for routing 10 messages, we save 10 times the path gain (in this case 20 hops per message) i.e a total of 200 transmissions, while we spend only 50 messages for tagging. The convergence time for the two strategies is compared in Fig. 7. The variations are small, although the evaluation methods are different. Let the convergence time be the

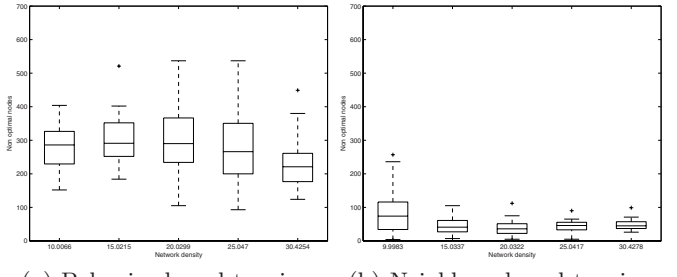


(a) Behavior based tagging (b) Neighbors based tagging

Fig. 7. Convergence time

time when the number of tagged nodes remained unchanged for the last 300 steps. Therefore we consider that the algorithm is fast convergent.

A significant difference can be noticed with respect to the number of tagged nodes (Fig. 8): *NBT* will mark only 1/4 of the nodes marked by *BBT*. Another important observation is that the number of tagged nodes does not increase for higher densities. The reason is that the geometrical surface covered by tagged nodes decreases as well with the increase in density. The probability that a node has greedy neighbors toward the destination is direct proportional to the density. Since only tagged nodes transmit overhead messages, and since this is done only once, reducing the number of tagged nodes leads to a smaller overhead.



(a) Behavior based tagging (b) Neighbors based tagging

Fig. 8. Number of tagged nodes

6 Conclusions

This paper presented an algorithm for early detection and avoidance of obstacles, by progressive evaluation of the nodes making routing decisions.We proved several properties of the algorithm: stability, convergence and we showed that it preserves previous properties of the geographic routing algorithms.

The simulations show the performances of the proposed algorithm, better then those of the state of the art algorithms. At the same time, the algorithm is lightweight, it needs only 1 bit of information piggybacked on the topology

maintenance messages, or sent reactively, and only one extra bit of storage for each neighbor.

The complexity is low - for a fixed destination the overhead introduced depends only on the obstacle size and shape, while it is independent of the network size. Furthermore, this overhead is paid only once, independently of the load of the network, while all messages benefit of reduced path length. Additionally, the algorithm is flexible, it can be used on top of a large class of routing and planarisation algorithms. At the same time it is independent on the physical layer model used.

Future work will consider different assumptions for network topology: multiple base stations and mobile base station.

References

1. Bose, P., Morin, P., Stojmenović, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wirel. Netw. 7(6), 609–616 (2001)
2. Chatzigiannakis, I., Dimitriou, T., Nikoletseas, S., Spirakis, P.: A probabilistic algorithm for efficient and robust data propagation in smart dust networks. Ad-Hoc Networks Journal 4(5), 621–635 (2006)
3. Chatzigiannakis, I., Nikoletseas, S., Spirakis, P.G.: Efficient and robust protocols for local detection and propagation in smart dust networks. Special Issue on Algorithmic Solutions for Wireless,Mobile, Ad Hoc and Sensor Networks, ACM/Baltzer Mobile Networks and Applications(MONET) Journal 10(1-2), 133–149 (2005)
4. Chen, D., Varshney, P.K.: A survey of void handling techniques for geographic routing in wireless networks. Communications Surveys and Tutorials, 50–67 (2007)
5. Chen, S., Fan, G., Cui, J.: Avoid void in geographic routing for data aggregation in sensor networks. International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Special Issue on Wireless Sensor Networks (2006)
6. Datta, S., Stojmenovic, I., Wu, J.: Internal node and shortcut based routing with guaranteed delivery in wireless networks. Cluster Computing 5(2), 169–178 (2002)
7. Fang, Q., Gao, J., Guibas, L.J.: Locating and bypassing holes in sensor networks. Mob. Netw. Appl. 11(2), 187–200 (2006)
8. Gabriel, K.R., Sokal, R.R.: A new statistical approach to geographic variation analysis (1969)
9. Heissenbüttel, M., Braun, T., Bernoulli, T., Wälchli, M.: BLR: Beacon-less routing algorithm for mobile ad-hoc networks (2003)
10. Jain, R., Puri, A., Sengupta, R.: Geographical routing using partial information for wireless ad hoc networks (1999)
11. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: Mobile Computing and Networking, pp. 243–254 (2000)
12. Kim, Y.-J., Govindan, R., Karp, B., Shenker, S.: Geographic routing made practical. In: NSDI 2005: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, pp. 217–230. USENIX Association, Berkeley (2005)
13. Kranakis, E., Singh, H., Urrutia, J.: Compass routing on geometric networks. In: Proc. 11 th Canadian Conference on Computational Geometry, Vancouver, August 1999, pp. 51–54 (1999)
14. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: Of theory and practice (2003)

15. Kuhn, F., Wattenhofer, R., Zollinger, A.: Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In: Proc. 4th ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc) (2003)
16. Lin, X., Lakshdisi, M., Stojmenovic, I.: Location based localized alternate, disjoint, multi-path and component routing schemes for wireless networks. In: MobiHoc 2001: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pp. 287–290. ACM, New York (2001)
17. Mathar, R., Mattfeldt, J.: Optimal transmission ranges for mobile communication in linear multihop packet radio networks. Wirel. Netw. 2(4), 329–342 (1996)
18. Moraru, L., Leone, P., Nikoletseas, S., Rolim, J.D.P.: Near optimal geographic routing with obstacle avoidance in wireless sensor networks by fast-converging trust-based algorithms. In: Q2SWinet 2007: Proceedings of the 3rd ACM Workshop on QoS and security for wireless and mobile networks, pp. 31–38. ACM Press, New York (2007)
19. Nikoletseas, S., Powell, O.: Simple and efficient geographic routing around obstacles for wireless sensor networks. In: Demetrescu, C. (ed.) WEA 2007. LNCS, vol. 4525, pp. 161–174. Springer, Heidelberg (2007)
20. Stojmenovic, I., Lin, X.: Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Trans. Parallel Distrib. Syst. 12(10), 1023–1032 (2001)
21. Toussaint, G.: Some unsolved problems on proximity graphs (1991)
22. Urrutia, J.: Routing with guaranteed delivery in geometric and wireless networks. In: Handbook of wireless networks and mobile computing, pp. 393–406. John Wiley & Sons, Inc., New York (2002)

DIN: An Ad-Hoc Algorithm to Estimate Distances in Wireless Sensor Networks

Freddy López Villafuerte and Jochen Schiller

Freie Universität Berlin, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
{lopez,schiller}@inf.fu-berlin.de

Abstract. A current challenge in wireless sensor networks is the positioning of sensor nodes for indoor environments without dedicated hardware. Especially in this domain, many applications rely on spatial information to relate collected data to the location of its origin. First of all, an estimation of the distance between two nodes is necessary to determine their positions. So far, the majority of approaches have explored physical properties of signals such as the strength of a received signal or its arrival time. However, this has been problematic since either the complexity on the software or on the hardware side is not adequate for embedded systems, or the approaches lack the required accuracy. In this paper we present the DIN algorithm (Distance by Intersection of Neighborhoods) to determine the distance between two nodes in an Ad-hoc manner, relying solely on the investigation of local node densities. To evaluate the accuracy of this algorithm, we conducted extensive simulations and experimented with different testbed setups using real sensor nodes. We were able to assure competitive values for the measured error.

Keywords: Localization, Neighborhood, Network Density.

1 Introduction

Wireless Sensor Networks (WSN) [1] store and partially process the sensed data either within the same sensor nodes which take the local samples or transmit the sensed data to a remote central computer where the data will receive a bigger and more complex handling process. To have a record of the place of study it is very important to correlate the collected measurements sensed by the nodes to a specific location. Furthermore, the position of the nodes opens up new ways to detect special events track an object of interest and improve the network coordination by executing geographic routing algorithms. The location problem is especially crucial in WSN, because it is necessary to find methods that work in ad-hoc fashion and without additional specialized hardware to save scarce resources since the positioning indoors is not possible with GPS.

The first step into this direction is to estimate the distance between nodes. To obtain this information there is a variety of techniques that exploit physical phenomena such as the time of arrival of sound signals [2], the time difference of arrival between radio and ultrasonic signals [3,4], the use of interferometry [5], radio

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 162–175, 2008.
© Springer-Verlag Berlin Heidelberg 2008

signal strength indicator (RSSI) [6], or the use of camera pictures with a previous scene analysis [7]. In this paper we focus on the problem of GPS-less, ad-hoc and low cost localization for WSN. We propose a method to estimate distances based only on the analysis of local node densities called Distance by Intersection of Neighborhoods (DIN). This algorithm estimates distances between nodes which share a communication link using the number of nodes that are positioned in the union and intersection area of their communication ranges. We evaluated our algorithm for indoor usage using simulations and real hardware experiments.

The structure of the paper is as follows: First we motivate the need for a new, flexible and ad hoc technique to estimate distances applicable for indoor usage. Through different network setups and a thorough calibration of real sensor nodes, we present the results of a RSSI-based distance estimation in section 2. We propose a new alternative to develop a similar range-free system using the DIN algorithm in section 3.

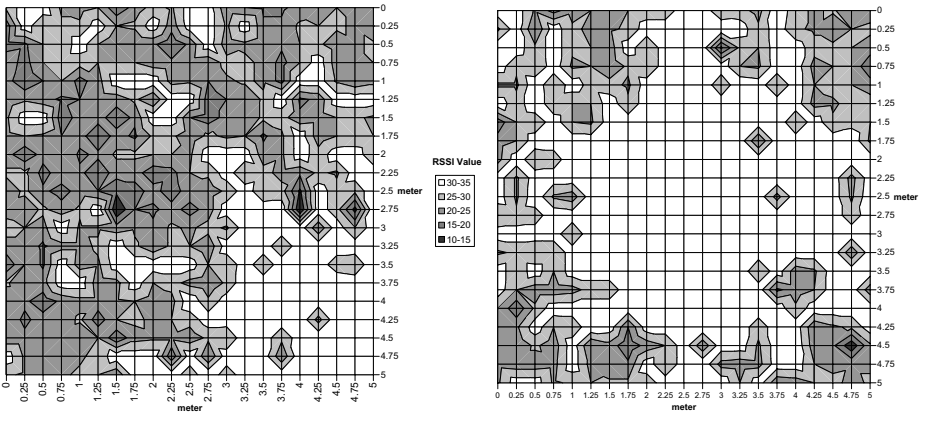
The mathematical model relating the number of nodes in the union and intersection area with the distances between them will be described in this section as a foundation of our proposed algorithm. Making use of the `ns-2` simulator, we look at the behavior of the DIN algorithm in uniform and near-uniform nodes distribution with different node densities. We verify the quality of our algorithm not only with the results of these simulations but also putting into practice the proposed technique on real sensor nodes with different network configurations.

The evaluation of the distance errors of the RSSI-based system, the `ns-2` simulations and the implementations of the DIN algorithm on real hardware in section 3 is presented. Section 4 discusses the related work and other approaches for distance estimations. Finally, we give an outlook on future work in section 5 and summarize our findings in the conclusions in section 6.

2 RSSI as Statement of the Problem

The Determination of the distance between sensor nodes that are close to one another (within the range of tens of centimeters up to a few meters) is usually carried out with the help of Time of Arrival (TOA) or Time Difference of Arrival (TDOA) systems. The accuracy that these systems are able to provide comes at the cost of a high synchronization overhead, thus high energy expenses at runtime and the need for dedicated hardware on the sensor nodes [3]. In contrast, range-free algorithms rely solely on conventional hardware of sensor nodes, with the preferred present technique to conclude the distance of the receiving node from the sender by means of mapping the measured RSSI value to a distance. This mapping has to be justified by previous measurements, but has the advantage that it imposes no additional cost on a node since it is provided by the transceiver practically for free.

To understand the distribution of RSSI values in an indoor setup, we measured these values with our MSB sensor nodes, (see section 2.1) at regular points and created maps, two of which are depicted in Figures 1 a and 1 b. These maps vizualize very well the problem that arises when utilizing a simple mapping:



(a) (b)

Fig. 1. (a) Received signal strength of a sending node placed at the lower right corner within an indoor testbed, (b) Received signal strength of a sending node placed in the middle of the network within an indoor testbed

As can be seen in map 1 a the transmission range is far from being regular, nodes may be far away from the sender and still receive a high RSSI value while others are closer and exposed to lower values, and thus will miscalculate their distance. Fluctuation of the received signal imposes a major challenge on current range-free algorithms. Also, Figure 1b indicates that the determination of a small distance in the range of tens of centimeters is not possible, since the resolution of RSSI does not allow for such an accuracy. Even worse, the distribution of RSSI values is influenced by spatial, temporal and environmental parameters, the orientation of the antenna and the choice of transceiver, making the calibration an almost unaccomplishable challenge. First at all, we implemented an RSSI-based distance estimation experiment to obtain the idea of its performance with real hardware using the Scatterweb nodes described in the next subsection. The main purpose is to know the quality of this range free technique for indoor environments using WSN.

2.1 ScatterWeb Sensor Network Platform

The hardware used to test the DIN algorithm was the ScatterWeb Modular Sensor Boards (MSB) [8]. These MSB feature the 16-bit microcontroller MSP430 from Texas Instruments equipped with 55 KB of flash memory and 5 KB RAM. In order to communicate to other nodes, each board with a Chipcon CC1020 transceiver uses the ISM band at 869 MHz. This transceiver allows monitoring the received signal strength (RSSI) at reception, the transmit power can be set manually by the developer. A number of additional sensors can be plugged to the core board, such as temperature, humidity or light sensors, in order to expand the standard functionalities of the node.

2.2 Communication Range Calibration

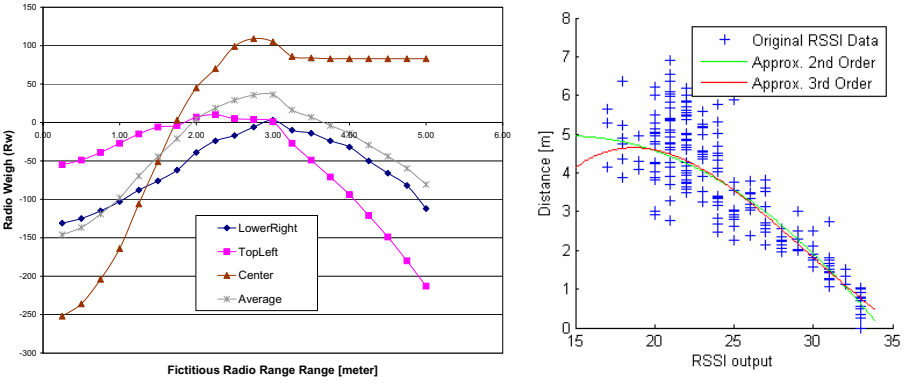
The primary goal has been to test the behaviour of distance estimation based on RSSI values with the best hardware configuration possible. Furthermore, it is also necessary for the implementation of our proposed algorithm (see section 3) to construct an experimental indoor set up where not all the nodes were within each others transmission range. The first problem that we encountered was that even with the smallest value for setting the transmit power, every node in the network could establish a radio communication link in the place where the nodes where deployed. To solve this problem, we use an RSSI value to artificially limit the transmission range by filtering signals below a certain value. Although at first glance this solution may seem to be a testbed workaround, the results obtained will still be valid in a larger multi-hop environment since the filter will be equally used here, thus no difference in the behaviour of our algorithm will be observed.

The next calibration for an approximation of a circular transmission range was obtained by mapping the radiation pattern of the MSB nodes on an indoor environment. For this purpose a sending node was located on three different positions of a 5x5 square area (upper-left corner, central position, and lower-right corner) in a seminar room of our institute. Two of the created maps are shown in the Figure 1a and 1b. The RSSI measurements were taken every 25 cm from an emitter node until a complete sweep of the setup area was finished. Both nodes where positioned over cleared desk height in order to provide a good transmission scenario. As we expected these figures have confirmed that the transmission is far from being circular but strongly irregular and without homogeneity. The nodes can be far away from the transmitter node and still receive high RSSI values while others are closer and exposed to lower values.

The process to determine a standard radio range for the DIN algorithm was done by analyzing every transmission pattern map previously produced and evaluating the quality of the transmission range in terms of fluctuations of the RSSI values of the area covered by the signal. From the measurements we reason that with an RSSI threshold of 33(-42.5 dBm) an artificially limit transmission range could be implemented. To determine a standard radio range for our system, it was necessary to analyze every transmission pattern map previously produced and to evaluate the quality of the transmission range in terms of fluctuations of the RSSI values of the area covered by signal. Taking as a reference the position of the sender node, we create different circular transmission range in steps of 0.25 cm until it covered the complete setup area. In order to find the best circular transmission range that fit better with the RSSI threshold, we evaluate every disc communication range with the help of the variable called radio weigh (R_W) defined as follow:

$$R_W = R_I - R_O - NR_I \tag{1}$$

In the matemathical expression of R_W from Equation 1 the number R_I is defined as the number of regular points inside the fictitious radio scope within the range of the artificially RSSI limit value (33). R_O is the variable that counts the RSSI



(a) (b)

Fig. 2. (a) Radio Weigh curves to determine a general approximation of the radio communication range, (b) Approximation of the distance between nodes based on RSSI for indoor scenario at a transmission power of 0x01

values in range but outside a given radio range and finally NR_I it is the number of points that are inside the disc communication range but that have an RSSI value lower than 33. Averaging all the R_W values of all the received signal strength measurements over the different scenarios we obtained the average curve of the Figure 2 a. We can observe the curve reaches its maximum value in 3 m. Thus, we decided to consider a radio transmission range equal to 3 m using an RSSI threshold value of 33 with a transmit power of 0x01 from the CC1020 radio transceiver.

2.3 RSSI-Based Distance Estimation

In order to determine distances from RSSI values, we interweave the corresponding RSSI data to each sender-receiver distances of different radiation pattern maps such as Figure 1 a and 1 b. Using Matlab, we construct a polynomial function as depicted in Figure 2 b. This approximation curve

$$f_x = -0.0127x^2 + 0.3697x + 2.2688 \tag{2}$$

is constrained to the measurement area since extrapolation instantly leads to intolerable errors, a fact that once again emphasises the need for careful calibration when relying solely on RSSI values. The protocol used to determine distances in the network was developed as follows: Every node in the network has the opportunity to broadcast its id. The receptor nodes register the signal strength of this packet and compute its distance to the sender node by substituting the value of the signal strength for x in Equation 2 and solving for f_x. Two different testbed layouts with 36 MSB nodes were used to test the RSSI-based distance estimation, see Figure 3 a and 3 b.

The results of our experiment are shown in Figure 3 c. We chose to use interquartile diagrams since it is possible to judge the value dispersions of the

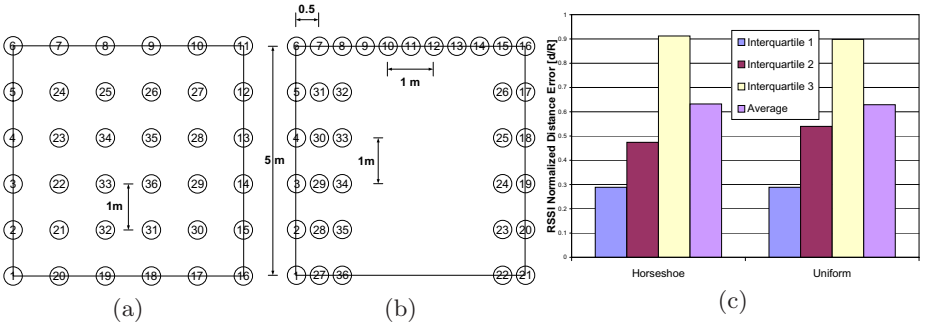


Fig. 3. (a) Uniform node distribution, (b) Horseshoe distribution, (c) Averaged normalized errors per interquartiles in horseshoe and uniform distribution derived from RSSI distance estimation

distance errors. It can be observed that the errors are artificially normalized after the collection of data, it means that every error value its divided by 3 m (the fictitious radius). This was applied to find an easier form to compare the results with the obtained on the next sections.

Although the normalization on the RSSI results was applied, the artificial radio transmission range was not used in this case. That means, every node in the network has been able to communicate and estimate its distances to each other. The data on RSSI-distance estimation shown in Figure 3 c, reveals an average misplacement of 1.88 and 1.89 m for an uniform and near-uniform distributed network respectively. As we expected, the RSSI measurements lack the required accuracy to determine distances between adjacent nodes.

3 Distance by Intersection of Neighborhoods

We introduce the Distance by Intersection of Neighborhoods algortihm as a proposal to increment the accuracy and flexibility of the range-free techniques such as the RSSI-based distance estimation. The essence of the approach is to determine distances between nodes through the analysis of the local density that they find to each other. To obtain a distance from the local density survey and put into practice our proposed algorithm, we began with a mathematical expression of the distance between two nodes in term of the union and intersection areas of their communication radii. We verify our mathematical model using the help of the `ns-2` simulator with different nodes distribution. Finally we put into practice the DIN algorithm with real sensor nodes in uniformly and near-uniformly distributed networks.

3.1 Relating Distance to the Union and Intersection Communication Areas

An important consideration in our mathematical foundations is that we based the DIN algorithm on an idealized radio model. Although we are aware that

assumption is not valid in reality, as we have shown in section 2, we use it because it was simple and easy to reason about mathematically. Three main assumptions were taken into consideration:

1. Unit disc graph radio transmission range.
2. Identical transmission ranges for all the nodes in the network.
3. Uniform distribution of nodes in the network.

Considering two neighbouring nodes which share a radio link communication, we can obtain the mathematical expression by geometrical analysis of their Circle-Circle Intersection. We defined the function H(d_n) as the relationship between the intersection area (A_i) and the union area (A_u) of the overlapping transmission ranges depicted as circles:

$$H(d_n) = \frac{A_i}{A_u} = \frac{4\cos^{-1}(\frac{d_n}{2}) - d_n\sqrt{4-d_n^2}}{4\pi - 4\cos^{-1}(\frac{d_n}{2}) + d_n\sqrt{4-d_n^2}} \tag{3}$$

Since we are interested in finding an expression for the distance between two neighbour nodes, we have to solve equation 3 for d_n, where d_n is the distance between the communicated nodes normalized by the circular radio range R. For the expression $H(d_n)$, we relate $\frac{A_i}{A_u} \approx \frac{k_i}{k_u}$. Where k_i is the number of nodes in the intersection area A_i and k_u denote the number of nodes that are in the union area A_u. Using MatLab, we obtained a polynomial approximation of degree 3 to determine the normalized distance between nodes.

$$d_n \approx \begin{cases} -2.73H_n^3 + 5.66H_n^2 - 4.88H_n + 1.88 & k_i \neq k_u \\ \frac{1}{k_i - 1} & k_i = k_u \end{cases} \tag{4}$$

Equation 4 is limited by $H(d_n)$ values between 1 and $\frac{4\cos^{-1}(\frac{1}{2})-\sqrt{3}}{4\pi-4\cos^{-1}(\frac{1}{2})+\sqrt{3}}$. Those values assure a shared link communication between two adjacent nodes.

3.2 Simulation Results of the DIN Algorithm with `ns-2`

An important issue to examine with the help of `ns-2` has been to determine the behaviour of DIN under variable network settings. The simulation results are obtained with a fixed number of 100 nodes. In order to examine the effect of node density variability, the network size was increased until the density of nodes become too sparse, thus the network disconnected. Another way to get the density variability is changing the transmission range of nodes accordingly. Using the DIN algorithm, every node in the network is able to compute its relative distances to those nodes that are in its transmission range. To measure the performance of DIN independent of variable radio communication ranges we use the normalized error as can been seen in Figure 4 a and Figure 4 b. The corresponding normalized error is simply the absolute value of the actual distance between nodes and the calculated distance divided by the radius of the node transmission range. To be able to compare the accuracy of the DIN algorithm with different network sizes, we define the Space-Range Ratio (SRR)

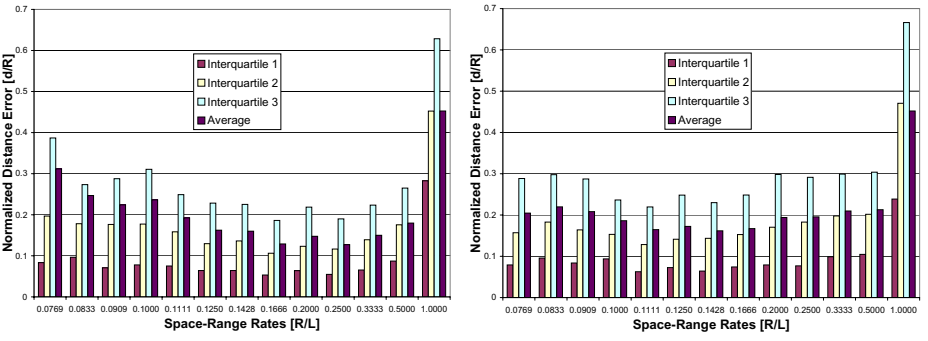


Fig. 4. (a) Absolute normalized errors in distance estimation versus covered radio range in a uniformly distributed network, (b) Absolute normalized errors in distance estimation versus Space-Range Rates in a Horseshoe setup

as the relationship between the radio communication scope of the idealized node over the Length (L) of the square side where the nodes are deployed. Although the nature of DIN was yielded for uniform distributions, a set of simulations with a near-uniform distribution using `ns-2` was implemented. This second step is in order to test the accuracy of the DIN algorithm on a different set up.

Figure 4 a shows that the best performance of the DIN algorithm is produced with an SRR value of 0.1666, where the 25% of the estimations have normalized error values less than $0.0532R$. Comparing to our previous work, we discover that the DIN algorithm yields better performance than WDNI [9] algorithm, where the smallest normalized error reported was with a value of $0.16R$. Although the absolute normalized distance error in Figure 4 a shows the trend to decrease with increasing node densities. Unlike WDNI, the DIN algorithm uses solely the number of local nodes without the help of a weighting function, thus it is not compensated for high node densities. We can denote that the duty zone of DIN is between SRR values of 0.0769 and 0.5. Those values represent deployed spaces with L values from $2R$ to $13R$ respectively. In this interval, we can see that the normalized distance error for the 75% of the estimations is less than $0.39R$, see Interquartile 3. For bigger deployed areas than SRR values of 0.0769, the average normalized distance error increase smoothly. This is due to the connections in the network star to break, so the interquartiles begin to reach the maximum normalized error. In Figure 4 b, we can observe that for SRR values between 0.0625 and 0.5 the normalized average error and the 75% of the error values in every case is lower than $0.37R$. For this configuration, DIN has better performance than in the uniform distribution, that means, for values of SRR smaller to 0.0769 it continues displaying smaller error under to $0.37R$. Unlike to the uniform distribution, the errors in horseshoe set up grow up in a smooth way. That is due to the deployed area in this configuration is smaller compared to the uniform distribution, in such a way that the network remains connected longer but producing higher errors for bigger deployed spaces.

Table 1. Minimal and maximal normalized error values in simulations of the Uniform and Horseshoe distributions

	Uniform Distribution							
	1_{st} Interquartile		2_{nd} Interquartile		3_{rd} Interquartile		Average	
	Value	SRR	Value	SRR	Value	SRR	Value	SRR
Min. Norm. Error	0.0532	0.1666	0.1061	0.1666	0.186	0.1666	0.127	0.25
Max. Norm. Error	0.0966	0.0833	0.1968	0.0769	0.387	0.0769	0.3118	0.0769
	Horseshoe Distribution							
	1_{st} Interquartile		2_{nd} Interquartile		3_{rd} Interquartile		Average	
	Value	SRR	Value	SRR	Value	SRR	Value	SRR
Min. Norm. Error	0.0627	0.1111	0.1283	0.1111	0.2195	0.1111	0.1618	0.1428
Max. Norm. Error	0.1046	0.5	0.2	0.5	0.36	0.0625	0.2832	0.0625

Looking at the normalized error of interquartile one, we realize that the best 25% of the errors is presented for a SRR value of 0.1111 showing error values less to 0.0627. Once again, we can observe an operation area in terms of SRR values, for high node densities (SRR=1) the distance estimations have less accuracy. They assume to be uniformly distributed causing high error rates in the distance estimations. On the other hand, DIN loses precision for networks with low node densities due to the lack of nodes producing less available information.

Table 1 show the minimum and maximum values over all the experiments obtained with the different network distribution using the `ns-2` simulator. In this section, we confirm that a node which uses the DIN can estimate distances between their neighbours as long as the transmission range is set to a value that enables most of the nodes to experience a neighbourhood close to a uniform distribution. The second step to test the accuracy of our algorithm is implementing DIN using real hardware; the results of this new test are presented in the next section.

In this section, we saw that DIN will work very well compared to the error value of the uniform distribution, see table 1 . We confirm once again that our algorithm works as long as the transmission range is set to a value that enables most of the nodes to experience a neighbourhood close to a uniform distribution.

3.3 Experimental Evaluation of DNI

Simplifying assumptions about radio propagation, network coverage and node distributions are common in network research. The core idea of DIN is to use a circular transmission range of nodes to find a relationship between distances and local nodes densities. Since the results of simulations were very promising and trying to validate a comparison with the RSSI-based distance estimation, we decided to implement DIN with the same testbed settings including the physical setup of the sensor nodes with the same transceiver settings. We measure the impact of a real environment on the performance of our algorithm using the approximation of the circular radio transmission scope limited by an RSSI threshold presented in section 2.2. First of all, we replace the computation of Equation 4

with a density-to-distance lookup table. Depending on the number of nodes in common K_i and the local node density K_u, a node can derive its distance from a neighbouring node. The protocol of DIN proceeds in three phases. In phase one, every node in the network broadcast a HELLO packet to discover neighbouring nodes within its communication range. It is important to take into consideration that signals received with an RSSI value below 33 will be dropped automatically to preserve the artificially constructed transmission range. To avoid collisions on the medium, we implemented a delay timer depending on the node ID. The information obtained in the first phase is a neighbor table with a single entry for each discovered neighbour.The second step on the DIN protocol is the exchange of neighbour tables. This process allows finding how many nodes are in the union and intersection transmission area of two neighbour nodes in the network. With this information, every node is able to compute the distance to an adjacent node looking to its density-to-distance table. The main problem to exchange neighbour tables in the network was the communication link asymmetries. Here, a sensor node can receive signals of another node perfectly but communications in the other direction fail. To prevent retransmission of neighbour table's request, the expiration of an internal timer limits the overall waiting time. When a node in the network experiments an asymmetric link, the DIN set the estimation distance to the maximum value.

The protocol of DIN can be naturally integrated into any routing overhead. The exchange of neighbourhood information and HELLO packets are subject to most routing schemes, thus may also be utilized by DIN when available. Additional information such as the local view on the network of each node can be piggybacked on regular data packets to minimize the overhead for the distance estimation. Therefore, DIN can be implemented on top of existing sensor network software at very low additional communication costs. As we mentioned before, we used the same two network configurations shown in section 2.3 to experiment with the DIN algorithm. To assure a best comparison between our algorithm and the results of the RSSI-based distance estimation, the same nodes were deployed on the same seminar room over cleared desk height. The room was big enough to save a distance at least 1 meter between border nodes and the walls.

The main results are depicted in Figure 5 a. Here, the average normalized distance error per interquartile with the DIN is plotted, as well as the dispersion of obtained error values with the help of the interquartile diagrams. Once again, DIN works best for uniformly distributed network. The average, normalized miscalculation of the nodes of $0.325R$ in this setting equals to 0.975 m, with the best 25% of the distance calculations having an error below $0.118R$ or 0.354 m, a value that provides a good accuracy for indoor usage. In 75% of all cases, the error remains at a value of $0.443R$ or a maximal offset of 1.3 m within acceptable bounds. On the other hand, the horseshoe distribution remains below a threshold of $0.15R$ which is equivalent to 0.45 m in interquartile 1, and features an average error of roughly 1 m at the most, an observation that shows the validity of applying the DIN to near-uniform network distributions despite its initial design for uniform distributions.

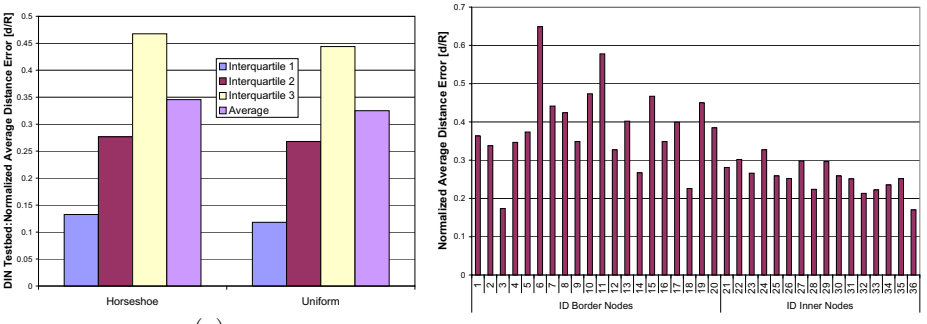


Fig. 5. (a) Normalized distance errors per interquartile in horseshoe and uniform distribution derived from the DIN testbed, (b) Average normalized errors in uniform node distribution for border and inner nodes in the experimental setup

An interesting question that we wanted to examine has been the influence of the node placement, more precisely the membership of nodes to the border or inner portion of the network on the distance estimation. The bars of nodes 1 to 20 in Figure 5 b represent the error of nodes placed at the border of the network, while nodes 21 to 36 denote the inner sensor nodes. In the portion of the inner nodes the best estimation of the network is presented with a value of 0.17R. The 94% of the estimations in this section is lower than 0.30R. The border nodes made distance estimations with a value lower than 0.4R for the 65% of the total cases. Making a closer analysis over the network, we found that the border nodes distance estimations are worse than the inner nodes due to a poor neighbour table quality. Asymmetric links and packet collisions led to a neighbour table with far less entries than usual.

Two interesting points in the graphical are the normalized average distance estimations of the nodes 6 and 11 which present distance estimations far from being acceptable. Both nodes were placed on the top corners of our experimental testbed. Analyzing its information obtained during run time, we realized that they could set a communication link with nodes that were outside of the artificial radio communication scope. Thus they underestimate the real distance to these nodes. However there are border nodes like the node number 3 that presents good distance estimations. This is because to each calculated distance estimation has been exceptionally good. Overall, we can conclude that the node placement does seem to have an influence on the distance calculation but more and larger scenarios have to be evaluated to add statistically significant evidence to such a proposition.

4 Evaluation of DIN

The good results observed in the interquartile diagrams by testing the DIN algorithm on the `ns-2` simulator, are confirmed by the test run conducted by the Scatterweb sensor nodes. Looking on the interquartile 1 and 2 for uniform and horseshoe node distributions, we realize that the average normalized

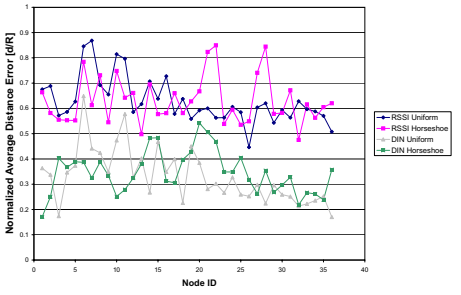


Fig. 6. Average normalized errors per node in horseshoe and uniform distribution with the RSSI and the DIN distance estimation

errors are slightly lower in a simulation environment than in an implementation on real hardware. Keep in mind that the SRR value for the experimental setup correspond to a simulative value between 0.5 and 1 which has to be considered when comparing the overall averages of testbed and simulations results.

Taking as a reference the interquaertiles with an SRR value of 0.5 for the cases of horseshoe and uniform nodes distribution on simulation diagrams, we can see that the interquartile 3 of these both distributions in our testbed add to higher values due to an increase number of miscalculation. We consider that this behaviour is due to external influences such as fading, interference or asymmetric links. However the discrepancies on the average normalized errors between real and simulation environments is not higher than $0.14R$ which it is an acceptable behaviour for the practical usage. The data on RSSI distance estimation as shown in Figure 3 c reveals the weaknesses of relying solely on RSSI readings. In average RSSI distance estimation errors are almost twice as high for all tested scenarios compared with the ones provide by the DIN algorithm using real hardware. The average misplacement using the RSSI distance estimations is of 1.88m in a uniformly distributed network. A view on the normalized average error per node, see Figure 6, nicely illustrates the superiority of the DIN in the different node distributions, a result that confirms our expectations.

As we see in Figure 6, the approximation of the distances based on RSSI values lacks simply the required flexibility to cope with the problem of asymmetries, interferences and fluctuations that are typical of the received strength maps

Table 2. Testbed Comparisons

	DIN Real		RSSI		DIN Simulation	
	Uniform	Horseshoe	Uniform	Horseshoe	Uniform	Horseshoe
Min.Norm.Error	0.00047	0.003833	0.003957	0.00766	4.52E-5	9.99E-6
Max.Norm.Error	1.3132	1.1574	2.1666	2.089	0.602477	0.7145

shown in section 2. With the help of the DIN algorithm it is at least partially solved with the knowledge of the local node densities. The best and worst error values for the distance estimations in the different scenarios and implementations are depicted in table 2.

5 Future Work

First of all, it is necessary to evaluate in a rough manner the impact of the fictitious variable radio transmission range in the DIN algorithm using different transmission power into different setup configurations. Another important point is to confirm the accuracy of DIN in larger testbeds for variable node densities. On one hand, we have to make an analysis of the behaviour of the distance estimations in multi-hops environment using our algorithm. On the other hand, it will be especially interesting to find out whether a lower bound for the number of neighbouring nodes and a given accuracy can be derived for multi-hop, low, medium and high density networks. Finally, the main goal will be the use of the DIN algorithm in the localization context. Therefore, we plan to simulate and develop with real hardware the position through our algorithm. Quality comparisons with other localization approaches such as DVHop or APIT [10] will be included in future work.

6 Conclusion

In this paper we presented DIN, an algorithm to estimate distances between two adjacent nodes based solely on local neighbourhood information. As a foundation, the area of intersection of two overlapping transmission ranges has been related to the number of local density of the nodes involved to determine their distances. In simulations, we can observe that the duty zone of DIN is between SRR values of 0.0769 and 0.5 for uniformly distribution networks and between SRR values between 0.0625 and 0.5 for near-uniform distribution networks with the majority of normalized error values below 0.39R. Here, the best average, normalized distance error has been $0.127R$ for a uniform distribution of sensor nodes. The good results obtained putting into practice the DIN algorithm with different testbed layouts, reflect the findings of the simulations, although we were only able to analyse a fraction of the simulation cases. Finally, we confirm the better accuracy on distance estimation of our approach comparing the real hardware tesbed results with the obtained using solely RSSI-values. With this work, we demonstrated that DIN yields competitive error values for distance estimation. The advantage of this approach is that neither the usage of specialized hardware, nor the measurements of physical properties that are inaccurate or unreliable are necessary for this estimation. The DIN is a completely Ad-hoc algorithm that it keeps the overhead in communication and calculation at minimum. We therefore believe that the knowledge about local node densities can be used as a new parameter to solve the localization problem.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. Journal Communications Magazine, 102–114. IEEE (2002)
2. Simon, G., Maróti, M., Lédeczi, Á., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., Frampton, K.: Sensor network-based countersniper system. In: SenSys 2004: Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, pp. 1–12. ACM Press, New York (2004)
3. Priyantha, N.B.: The Cricket Indoor Location System. PhD thesis, Massachusetts Institute of Technology (2005)
4. Broxton, M., Lifton, J., Paradiso, J.: Localizing a sensor network via collaborative processing of global stimuli. In: Proceedings of the Second European Workshop on Wireless Sensor Networks, pp. 321–332. IEEE, Los Alamitos (2005)
5. Maróti, M., Völgyesi, P., Dóra, S., Kusý, B., Nádas, A., Lédeczi, Á., Balogh, G., Molnár, K.: Radio interferometric geolocation. In: SenSys 2005: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, pp. 1–12. ACM Press, New York (2005)
6. Lenz, D.: Homepage of the ekahau project (2007), http://www.ekahau.com/
7. Microsoft-Research: Homepage of the easyliving project (2007), http://research.microsoft.com/easyliving
8. FU-Berlin: Homepage of the scatterweb project (2007), http://www.scatterweb.mi.fu-berlin.de
9. López-Villafuerte, F., Terfloth, K., Schiller, J.: Using network density as a new parameter to estimate distance. In: The Seventh International Conference on Networking, ICN 2008, Cancun, Mexico, p. 6. IEEE Press, Los Alamitos (2008)
10. He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free localization schemes in large scale sensor networks. In: MobiCom 2003: Proceedings of the 9th annual international conference on Mobile computing and networking, pp. 81–95. ACM Press, New York (2003)

Cheating on the CW and RTS/CTS Mechanisms in Single-Hop IEEE 802.11e Networks

Szymon Szott, Marek Natkaniec, and Andrzej R. Pach

AGH University of Science and Technology,
Department of Telecommunications,
Kraków, Poland
{szott,natkanie,pach}@kt.agh.edu.pl

Abstract. This paper presents a work in progress which deals with the problem of node misbehaviour in ad-hoc networks. A realistic approach is used to determine the impact of contention window manipulation and RTS/CTS cheating. It is explained why IEEE 802.11e ad-hoc networks are more prone to misbehaviour. The paper presents simulation results related to the mentioned types of misbehaviour. The analysis is performed for several distinct scenarios, which yields novel results. It is shown under which conditions a misbehaving node can gain a significant advantage over well-behaving nodes. The limitations of the IEEE 802.11e standard in providing QoS in the presence of misbehaving nodes is also presented.

Keywords: Ad-hoc networks, IEEE 802.11e, misbehaviour.

1 Introduction

With the increasing popularity of wireless connectivity in mobile devices (laptops, PDAs, cell phones, etc.) there is a need for interconnecting these devices in a spontaneous manner. Mobile ad-hoc networks (MANETs) are networks built without infrastructure in which every node acts as both terminal and router. Thus, they rely on the cooperation of nodes to ensure the proper functioning of the network. A problem arises if a node decides not to cooperate with others. We call such actions *misbehaviour*. A node may decide to misbehave in order to gain certain measurable profits (such as higher throughput, increased battery life). Misbehaviour is always done at the cost of the well-behaving nodes in the network. Therefore, it would be beneficial if such actions were, if not made impossible, then at least discouraged.

The problem of node misbehaviour is strengthened by the fact that the current WLAN standards (the IEEE 802.11 family) do not contain any incentives for nodes to behave accordingly. The 802.11 standards are all based on the notion that each node will strictly adhere to them. However, new wireless drivers [8] enable easy modification of MAC layer parameters. Section 2 describes the 802.11 standard (in particular the QoS extension – 802.11e) and shows to what forms of misbehaviour the standard is prone to.

The focus of this paper is put on two types of misbehaviour in ad-hoc networks. One of them is contention window (CW) cheating. This means modifying the

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 176–188, 2008.
© Springer-Verlag Berlin Heidelberg 2008

parameters introduced in the 802.11 standard (CW_{min} and CW_{max}), which are responsible for channel access. This, and other different aspects of misbehaviour in MANETs, has already been addressed in the literature (Section 3). However, the proposed solutions do not take many aspects into account. One particular aspect is the RTS/CTS mechanism (normally used to avoid the hidden node problem) and its influence on network performance in the presence of misbehaving nodes. This is related to the second type of misbehaviour discussed in this paper – cheating on the RTS/CTS mechanism. A node may decide on not using this mechanism, even though other nodes in the network do.

In this paper we show the results from several simulation scenarios (Sections 4 and 5). We try to answer the following questions: How does CW cheating impact network performance (throughput, delay, and fairness) when RTS/CTS is used? Is this affected by the network size? Is cheating on the RTS/CTS mechanism beneficial for the misbehaving user? Should it be used alone or together with CW cheating? How do these two types of misbehaviour impact the QoS provisioning mechanisms of 802.11e? The authors of the paper prove that a rational misbehaving node will choose the lowest possible CW parameters as they are the most beneficial. The most innovative contribution of this paper is the study of RTS/CTS cheating. To the authors' best knowledge, this has not been done before.

2 Misbehaviour in the 802.11 Standard

The IEEE 802.11 standard [3] defines a distributed access method for wireless networks – DCF (Distributed Coordination Function). This is the basic access method in ad-hoc mode. It is based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance).

In the context of DCF, the 802.11 MAC protocol distinguishes two important time periods: SIFS and DIFS (Short- and DCF- Inter Frame Space), the latter is longer. The lengths of both of these times are defined in the standard. When stations sense that the medium is free, they begin to measure these periods in order to estimate when they can begin their own transmission. The choice of the time period depends on the frame type.

The contention window algorithm works as follows. Each node, ready to transmit, senses the medium to determine whether it is idle. If so, it begins to transmit. Otherwise, since the channel is busy, the node waits for the current transmission to finish and then waits until the medium is free for one DIFS period. Afterwards, it randomly chooses a backoff value from the range [0, CW]. The chosen value denotes the time slot in which the node will begin its transmission. This decreases the probability that two nodes will transmit simultaneously and thus cause a collision. The countdown of the backoff value is paused when the channel is busy. When the backoff reaches zero, the node may transmit. At the beginning, the parameter CW is equal to a predefined value CW_{min}. After each collision, CW is doubled until it reaches another predefined value – CW_{max}. A successful transmission resets CW to the value of CW_{min}.

The IEEE 802.11e standard [4] introduces EDCA (Enhanced Distributed Channel Access) as the new distributed channel access mechanism. Traffic is divided into four access categories (AC) to provide appropriate QoS. These categories are, from the

highest priority: *Voice* (Vo), *Video* (Vi), *Best effort* (BE), and *Background* (BK). Each category has its own set of access parameters: AIFS (Arbitration InterFrame Space), TXOP (Transmission Opportunity), and, in particular, CW_{min} and CW_{max} (Table 1). These parameters are responsible for traffic differentiation.

Table 1. Values of CW parameters in 802.11e

AC	CWmin	CWmax
Voice	7	15
Video	15	31
Best effort	31	1023
Background	31	1023

The medium contention rules for EDCA are similar to 802.11 DCF. The difference in channel access prioritization is shown in Fig. 1 and Fig. 2. Each frame arriving at the MAC layer is mapped, according to its priority, to an appropriate AC. There are four transmission queues; one for each AC. AIFS[AC] is the parameter which replaces the DIFS of DCF. An internal collision resolution mechanism (virtual collision) is used to determine which frame can be sent. A physical collision can still occur, when two or more nodes start their transmissions simultaneously.

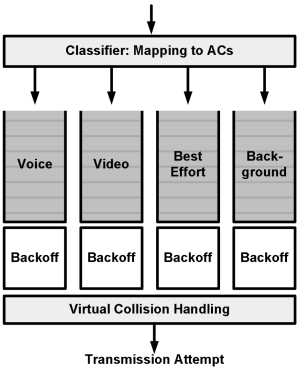


Fig. 1. Mapping to access categories [4]

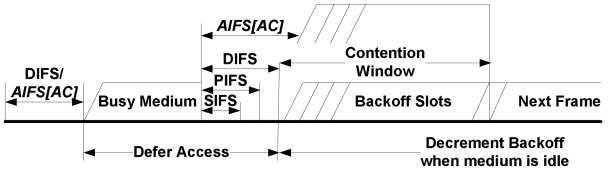


Fig. 2. Channel access prioritization [4]

In 802.11 the data exchange is made by the default simple DATA-ACK. This means that the sender sends a DATA frame and the receiver acknowledges it with an ACK frame. However, this leads to the hidden node problem. To counter this problem, the data exchange can be switched to RTS-CTS-DATA-ACK. The RTS/CTS mechanism uses two small frames (Request/Clear to Send) sent prior to the actual data exchange to inform neighbouring nodes about planned transmissions. This consumes bandwidth, but is necessary to avoid collisions caused by hidden nodes.

The IEEE 802.11 family of standards contain no incentive for nodes to adhere to the specified parameter values. Since new drivers allow manipulating these parameters it is possible that users will want to cheat to maximize their network performance. Based on the described characteristics of 802.11, several types of misbehaviour can be considered. In this paper we concentrate on two of those: cheating on the contention window parameters and the RTS/CTS mechanism. Both these mechanisms result in a decrease in channel access time. The former is done by choosing lower CW values and the latter by refusing to send the RTS/CTS frames.

3 State-of-the-Art

One of the first papers dealing with the problem of contention window misbehaviour was [6] (later extended in [7]). The authors take into account several misbehaviour strategies, such as selecting a smaller backoff (from the range [0, CW/4]), having a fixed backoff (1 slot) or not doubling the CW. It was the first paper to report degraded throughput in 802.11 infrastructure networks. The authors proposed an algorithm to solve this problem, under the assumption that the receiver (802.11 Access Point) is well-behaved. In their approach, it is the receiver, not the sender which chooses the random backoff value. This value is transferred to the sender in either a CTS or ACK frame. Misbehaviour occurs when the sender deviates from that backoff. The penalty assigned by the receiver is a higher backoff value in subsequent transmissions. The problem with this approach, other than requiring changes to the 802.11 standard, is that it is unsuitable for ad-hoc networks, where the receiver cannot be trusted. Hidden nodes also cause a problem in terms of determining the correct backoff.

Several works in the field were written by Baras et al.: [1], [2], and [9]. In [1], an algorithm (named ERA-802.11) for ensuring randomness in ad-hoc networks is proposed. It is based on the negotiation of CW parameters by sender and receiver (inspired by a protocol for flipping coins over the telephone). This assures a truly random backoff. The detection system developed in [6] is used to monitor nodes. In the case of misbehaviour, a report is sent to an external reputation management system. ERA-802.11 introduces extra messages so it is not compatible with the 802.11 standard.

The problem of trying to detect CW cheating is how to correctly observe the chosen backoff of another node. Observations are hindered by such factors as: interference from other transmissions, unsynchronized clocks, and non-deterministic medium access. It is also necessary to determine when to stop the observation and make a decision. This problem is discussed in [9]. The authors take into account an adaptive attacker and prove that a particular decision rule, the sequential probability ratio test

(SPRT), is the optimal approach to minimizing the number of needed observations. Similar work was done in [11].

Paper [10] presents DOMINO, an advanced software application designed to protect hotspots from greedy users. It monitors traffic, collects traces and analyzes them to find anomalies. DOMINO can detect many types of malicious and greedy behaviour, including backoff manipulation techniques. Anomaly detection is based on throughput (instead of observed backoff), which the authors acknowledge is not an optimal detection metric. The application can be seamlessly integrated with access points and it complies with standards. However, it cannot be directly used in ad-hoc networks.

To summarize, research efforts have so far been mostly focused on detecting nodes cheating on backoff in 802.11 infrastructure scenarios. Ad-hoc networks pose a challenge because they are distributed and have no centralized authority. Thus, there have not been that many papers discussing contention window cheating in MANETs. In papers [12] and [13] the authors show how modifying the CW values can degrade the performance of an 802.11e ad-hoc network. However, to the authors' knowledge, no papers have considered cheating on the RTS/CTS mechanism. Therefore, the subsequent sections address this issue.

4 Simulation Scenarios

The purpose of the simulation study was to determine how misbehaviour impacts ad-hoc network performance. The actions taken into consideration were manipulating CW parameters and cheating on the RTS/CTS mechanism.

The simulation analysis was performed with the use of the ns2 simulator with a modified version of the TKN EDCA model [14]. This model implements the 802.11e standard in ns2. The modification of the TKN EDCA model involved correcting the RTS/CTS implementation. The following scenario was considered. The number of homogenous nodes in the ad-hoc network was set to 5, 25, and 100 to represent small, average and large network sizes, respectively. All stations were within hearing range of each other (i.e., it was a single-hop network). The per-station offered load changed from 64 kb/s to 8 Mb/s.

Table 2. Simulation parameters

Parameter	Value
WLAN Standards	802.11b + 802.11e
Data rate	11 Mb/s
Routing protocol	None
Transport protocol	UDP
Node distribution	Random
Traffic generator	CBR
Packet size	1000 B
Packet exchange	DATA-ACK and RTS-CTS-DATA-ACK

Table 2 presents the various simulation parameters used. The node distribution was random and the traffic pattern – circular (with each node sending and receiving exactly one traffic stream). An example topology, for 5 nodes, can be seen in Fig. 3.

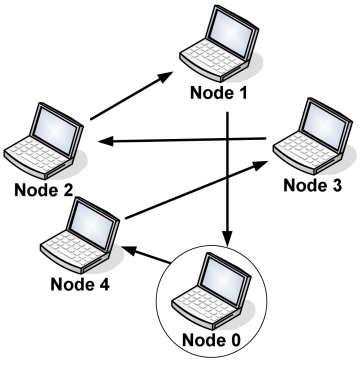


Fig. 3. Network topology

In each scenario, there was one misbehaving node (e.g., the encircled node in Fig. 3). All nodes used the *Best effort* priority to send their traffic. The well behaving (*good*) nodes had unaltered contention window parameters: CW_{min} = 31, CW_{max}=1023. The misbehaving (*bad*) node had these parameters significantly decreased: CW_{min} = 1, CW_{max} = 5. It seems realistic that the misbehaving node would choose such low (or even lower) parameters to maximize its gain. The effect of choosing other CW values and their impact on the use of the RTS/CTS mechanism is studied further on.

5 Results

The results of the uplink simulations are presented in the following figures. The plots present the curves, where the error of each simulation point for a 95% confidence interval does not exceed 2% (this is too small for graphical representation).

Fig. 4 presents the simulation results for the small network size (5 nodes). The figure shows the achieved uplink throughput as a function of offered load. The throughput is given for the well-behaving *good* nodes (on average) and for the *bad* node which cheats on the CW. The difference in the throughput of the *good* nodes was insignificant, that is why only the average is shown. In the first case RTS/CTS is off and in the second it is on. In the next case misbehaviour is turned off and RTS/CTS is either on or off. Finally, in the last case, the misbehaving node cheats both on CW and the RTS/CTS mechanism.

The black dashed lines are the reference values and represent the situation in which there is no misbehaviour. Turning on RTS/CTS lowers the saturation throughput. The solid lines represent the situation in which one node misbehaves (cheats on the CW) with RTS/CTS turned off. The misbehaving node dominates the

network (this has been shown in [12]). If RTS/CTS is turned on in such a network the throughput, of course, decreases: for the misbehaving node by 30 % and for the good nodes by 40 %.

Another case has been considered – when the misbehaving node decides not to use RTS/CTS despite the fact that the other nodes are using this mode of transmission. The gain is obvious – the misbehaving node's throughput almost reaches the throughput it had when RTS/CTS was not used in the network. This is obviously at the cost of the good nodes' throughput. Therefore, there is a strong incentive for the misbehaving node to turn off RTS/CTS whenever possible.

Similar results regarding obtained throughput occur for medium and large network sizes (Fig. 5 and Fig. 6). The difference is in the throughput achieved by the misbehaving node when the network is saturated because it decreases with network size.

There are two characteristic points in the figures which present throughput. The first occurs once the network reaches congestion. In other words, it is the point where if the network consisted only of well-behaving nodes it would become saturated. Until that point the *bad* node's presence is not harmful. After reaching the congestion point, the *bad* node increases its throughput at the cost of the *good* nodes. This occurs until the second characteristic point is reached. After this, the network is in saturation and the *bad* node has much more throughput than the average *good* node. These two characteristic points can be perhaps most clearly seen in Fig. 4. The first one appears for an offered load a bit higher than 1 Mbit/s, the second one – at approximately 7 Mbit/s. The conclusion is that analysis of misbehaviour should be limited to congestion scenarios. In non-congested networks the misbehaving node does not impact network performance.

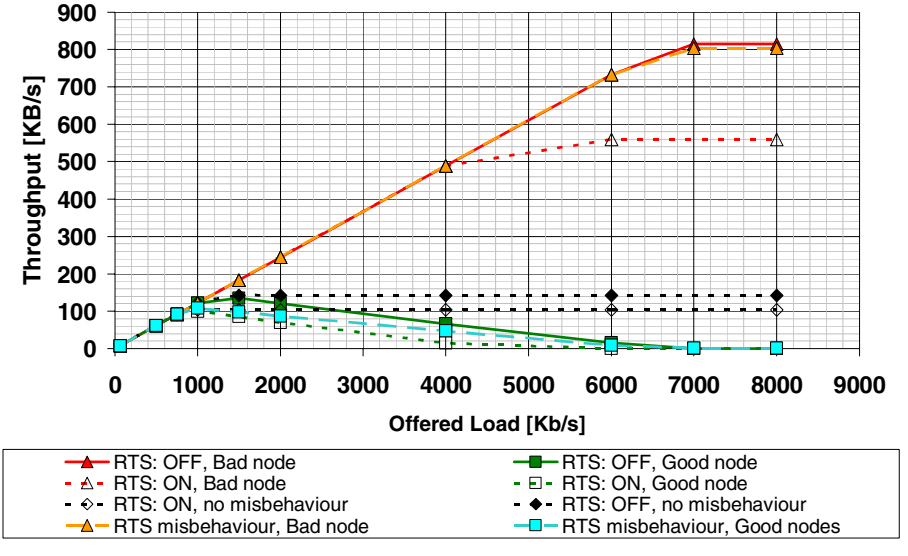


Fig. 4. Throughput vs. offered load (total no. of nodes: 5)

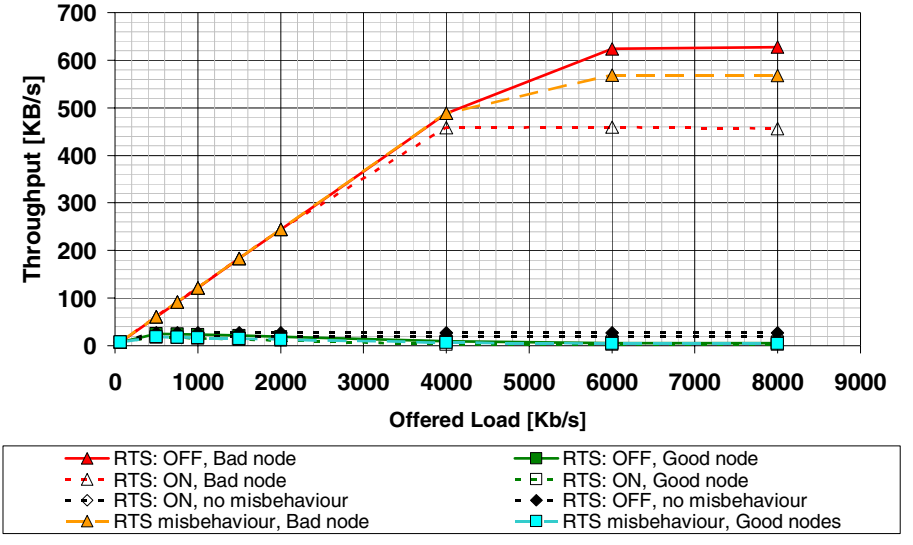


Fig. 5. Throughput vs. offered load (total no. of nodes: 25)

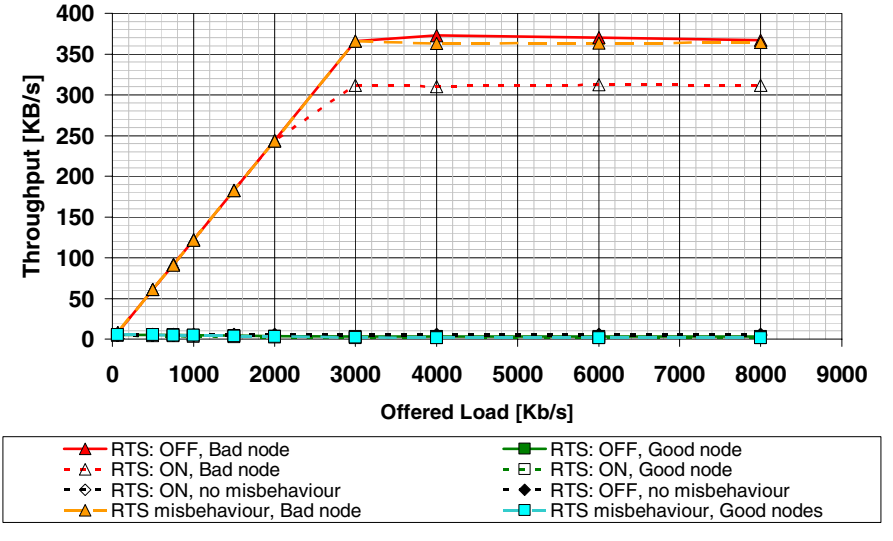


Fig. 6. Throughput vs. offered load (total no. of nodes: 100)

presents the average frame delay of the misbehaving and well-behaving nodes in the small network scenario. The delay of the *good* nodes suffers greatly in the presence of misbehaviour. It quickly rises very sharply in all cases. The delay of the *bad* node is at an acceptable level for much higher offered loads. With the RTS/CTS mechanism turned on, the delay is low until 4 Mbit/s. If it is turned off (intentionally

Fig. 7. Packet delay vs. offered load (total no. of nodes: 5)

or maliciously), it is at a low level until 6 Mbit/s. These observations confirm the conclusions presented above: cheating on the RTS/CTS mechanism "restores" the achieved delay to the value as when the network was not using RTS/CTS. Furthermore, it can be once again noted that in non-congested networks the misbehaving node does not impact network performance (in this case: delay). The measured delay was similar for larger simulated networks, therefore only this figure is being presented.

Two types of cheating have been discussed: manipulating the CW parameters and disabling RTS/CTS in a network which uses this mechanism. The following question arises: is the misbehaviour gain different when these actions are performed alone and together? The answer can be seen in Fig. 8, which shows the throughput gain of the misbehaving node in absolute values. In this case, simulations were performed for a network of 5 nodes (the rest of the simulation parameters remained unchanged) in which RTS/CTS was always enabled. Three cases were considered: the misbehaving node used either CW cheating, RTS/CTS cheating or a combination of both. The achieved throughput was compared with the average node throughput in a network with no misbehaviour. The result is that cheating only on the RTS/CTS mechanism does not give almost any benefits. This is obvious because when there are no hidden stations, the RTS/CTS mechanism only introduces a delay in the medium access. However, if this is combined with CW cheating the gain is much larger than when cheating only on the CW mechanism. There is a synergy between low contention window parameters and refusing to use RTS/CTS. When a node accesses the channel more often (through low CW parameters) the gain from not using RTS/CTS is greater.

In the previously mentioned simulations, the CW parameters of the misbehaving node were set to $CW_{min} = 1$ and $CW_{max} = 5$. In order to determine the exact impact of the CW values the following simulation study was performed. The network of 5 nodes (Fig. 3) was in saturation – all nodes were sending UDP traffic of an offered

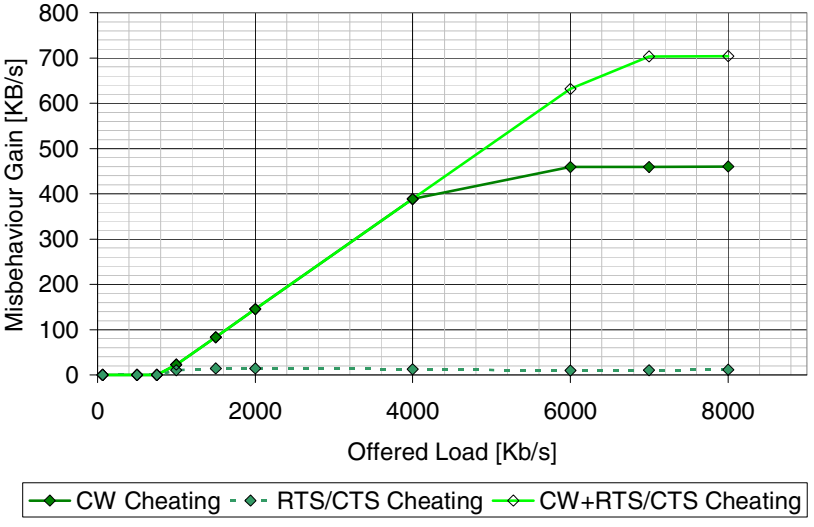


Fig. 8. Misbehaviour gain for different forms of cheating

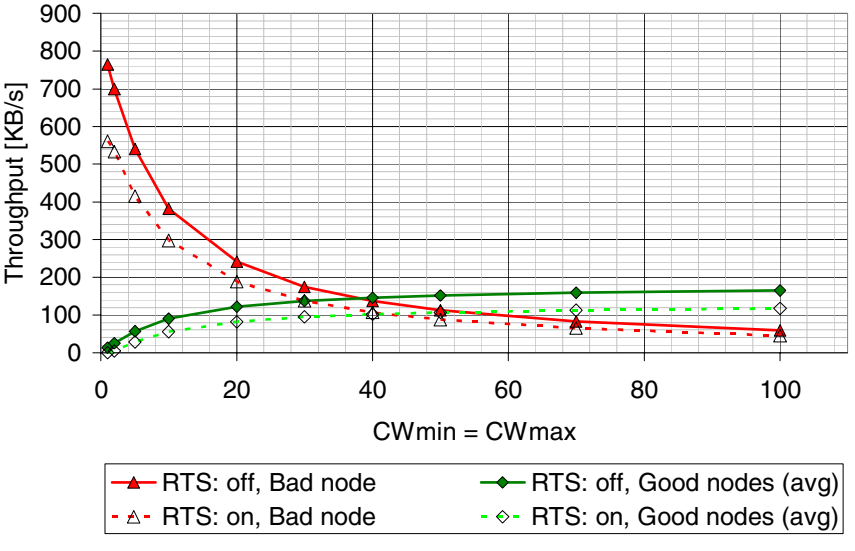


Fig. 9. Throughput comparison for different CW parameters

load of 7 Mbit/s. The RTS/CTS mechanism was either off or on. The misbehaving node varied it CW parameters ($CW_{min} = CW_{max}$) from 1 to 100 (Fig. 9). The highest throughput it achieved was for the smallest CW parameters and for RTS/CTS turned off. The *bad* node's throughput decreases in an exponential manner with the increase of the contention window size. The point where the *bad* node's throughput is

approximately equal to the average throughput of the *good* nodes occurs for CW_{min} = CW_{max} = 40. Since the 802.11 standard does not include any incentives for cooperation, a misbehaving user is free to chose the most profitable CW parameters (i.e., equal to 1).

When dealing with the 802.11e standard it is important to determine the impact of misbehaving in one AC on the performance of a higher priority AC. Simulations were performed, likewise, for a 5 node scenario. The RTS/CTS mechanism was turned on. The well-behaving nodes were using *Voice* priority to send their traffic (CW_{min} = 7, CW_{max} = 15). The misbehaving node continued to use *Best effort* traffic (with misbehaviour parameters CW_{min} = 1 and CW_{max} = 5). The results are presented in Fig. 10. In the first case, with no misbehaviour, the achieved throughput rates are in line with the 802.11e standard. When the *bad* node cheated on the CW, it was able to dramatically increase its throughput at the cost of the *good* nodes. Surprisingly, when the *bad* node cheated on both the CW and RTS/CTS mechanisms, an increase in throughput was observed for all nodes (even the *good* ones). This result can only be explained by the fact that the RTS/CTS mechanism introduces overhead which consumes a small portion of bandwidth. Since one node (the *bad* one) did not use RTS/CTS frames, the total available throughput in the network increased. Therefore, even the *good* nodes could use a small share of this newly available throughput to slightly increase their performance. Had the network consisted of more nodes, the increase in throughput of the well-behaving nodes would be even less significant. If the network was multihop and hidden nodes were present, the gain would depend on how the stations (especially the hidden ones) were placed. In particular it can be assumed, based on [5], that if the misbehaving node was a hidden one in a simple star topology, it would benefit neither from CW manipulation, nor from RTS/CTS cheating.

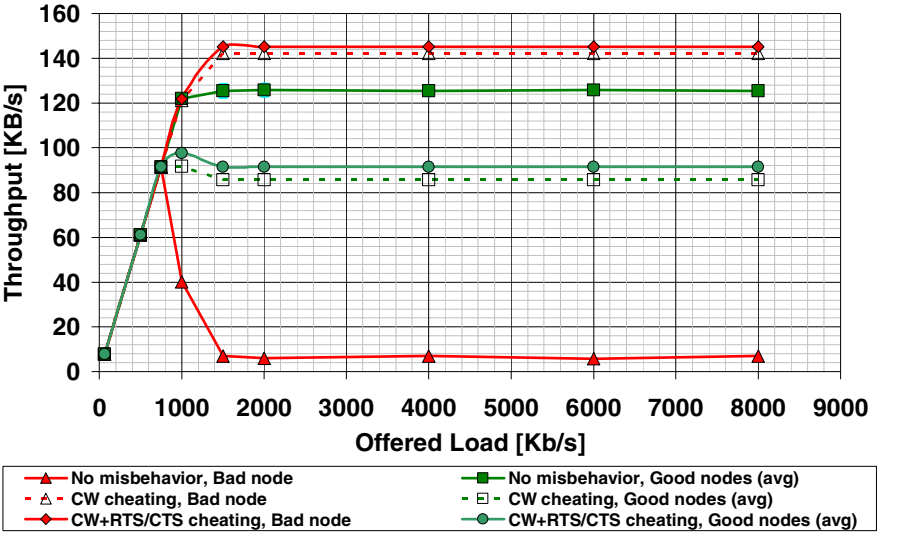


Fig. 10. Throughput vs. offered load for BE vs. Vo priority scenario

6 Conclusions

This paper presented the impact that cheating on the contention window and RTS/CTS mechanism has on single-hop ad-hoc networks. Several simulation scenarios were analyzed. Throughput, delay and fairness were considered for networks of different sizes. A rational misbehaviour model was assumed, i.e., the malicious user would perform simple actions to obtain significant gains.

The first conclusion is that the use of modified CW parameters allows a misbehaving node to jeopardize network performance. The throughput and delay of such a node is significantly better than well-behaving nodes. This occurs regardless of network size and whether the RTS/CTS mechanism is used.

Secondly, a node can cheat on the RTS/CTS mechanism, i.e., refuse to turn in on, even though the whole network is using it. It has been shown that while such behaviour does not provide gains, it is especially beneficial when joined with CW misbehaviour. When used together, these two types of misbehaviour can give greater advantages than when used alone.

Furthermore, a simulation analysis was performed for different CW values of the *bad* node. Assuming that the misbehaving user is rational, and taking into consideration the fact that 802.11 has no mechanisms to encourage proper behaviour, it is obvious that the lowest possible CW values should be chosen.

In non-congested networks, a node's misbehaviour, though theoretically observable, has no influences on its neighbours and is therefore harmless. Therefore, future studies should be focused on congested networks. In real-world ad-hoc networks saturation can be a common situation because of multimedia and peer-to-peer applications.

Finally, it was shown that 802.11e fails to provide QoS in the face of CW and RTS/CTS cheating. A misbehaving node can easily manipulate MAC layer parameters and thus gain an advantage over other nodes. Low priority traffic can be assigned such parameters, with which it can outperform high priority traffic.

Future work will take an even more realistic approach. Studies will focus on multihop ad-hoc networks, which suffer from the hidden node problem. Cheating on other EDCA parameters (AIFS, TXOP) will be taken into account. Furthermore, more complex traffic patterns and networks with more misbehaving nodes will be considered. It is important that misbehaviour is simple, straightforward and advantageous so that it can be performed by any casual user, not just an expert hacker. An analytical model will be derived to support the findings.

Acknowledgments. This work has been carried out under the Polish Ministry of Science and Higher Education grant no. N51739133.

References

1. BenAmmar, N., Baras, J.S.: Incentive compatible medium access control in wireless networks. In: Proceedings From the 2006 Workshop on Game theory For Communications and Networks (GameNets 2006), Pisa, Italy, October 14 (2006)
2. Cardenas, A.A., Radosavac, S., Baras, J.S.: Detection and Prevention of MAC Layer Misbehavior for Ad Hoc Networks. Technical Report (2004)

3. IEEE 802.11 Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specification. IEEE Inc., New York (1999)
4. IEEE 802.11e-2005, IEEE Standard for 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: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements (2005)
5. Kosek, K., Natkaniec, M., Vollero, L., Pach, A.R.: An Analysis of Star Topology IEEE 802.11e Networks in the Presence of Hidden Nodes. In: Proc. The International Conference on Information Networking 2008, ICOIN 2008, Korea (January 2008)
6. Kyasanur, P., Vaidya, N.H.: Detection and Handling of MAC Layer Misbehavior in Wireless Networks. In: International Conference on Dependable Systems and Networks (DSN 2003), p. 173 (2003)
7. Kyasanur, P., Vaidya, N.H.: Selfish MAC Layer Misbehavior in Wireless networks. IEEE Transactions on Mobile Computing 4(5) (September/October 2005)
8. MADWiFi – Multiband Atheros Driver for WiFi, `http://madwifi.org`
9. Radosavac, S., Baras, J.S., Koutsopoulos, I.: A Framework for MAC Protocol Misbehavior Detection in Wireless Networks. In: Proc. 4th ACM workshop on Wireless security (WiSe), Cologne, Germany (September 2005)
10. Raya, M., Hubaux, J., Aad, I.: DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In: Proceedings of the 2nd international Conference on Mobile Systems, Applications and Services (MobiSys 2004), Boston, MA, USA, June 06 - 09 (2004)
11. Rong, Y., Lee, S.-K., Choi, H.-A.: Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis. In: Proceedings of INFOCOM 2006, 25th IEEE International Conference on Computer Communications (April 2006)
12. Szott, S., Natkaniec, M., Canonico, R., Pach, A.R.: Impact of Contention Window Cheating on Single-hop IEEE 802.11e MANETs. In: IEEE Wireless Communications and Networking Conference (WCNC 2008), Las Vegas, NV, USA, March 31 – April 4 (2008)
13. Szott, S., Natkaniec, M., Canonico, R., Pach, A.R.: Misbehaviour Analysis of 802.11 Mobile Ad-Hoc Networks – Contention Window Cheating. In: Med Hoc Net 2007, Corfu, Greece, June 12–15 (2007)
14. Wiethölter, S., Emmelmann, M., Hoene, C., Wolisz, A.: TKN EDCA Model for ns-2. In: Technical Report TKN-06-003, Telecommunication Networks Group, Technische Universität Berlin (June 2006)

Adapting BitTorrent to Wireless Ad Hoc Networks

Mohamed Karim Sbai, Chadi Barakat, Jaeyoung Choi,
Anwar Al Hamra, and Thierry Turletti

Project-Team Planète, INRIA Sophia Antipolis, France
{mksbai,cbarakat,jchoi,aalhamra,turletti}@sophia.inria.fr

Abstract. BitTorrent is one of the Internet's most efficient content distribution protocols. It is known to perform very well over the wired Internet where end-to-end performance is almost guaranteed. However, in wireless ad hoc networks, many constraints appear as the scarcity of resources and their shared nature, which make running BitTorrent with its default configuration not lead to best performances. To these constraints it adds the fact that peers are both routers and end-users and that TCP-performance drops seriously with the number of hops. We show in this work that the neighbor selection mechanism in BitTorrent plays an important role in determining the performance of the protocol when deployed over a wireless ad hoc network. It is no longer efficient to choose and treat with peers independently of their location. A first solution is to limit the scope of the neighborhood. In this case, TCP connections are fast but there is no more diversity of pieces in the network: pieces propagate in a unique direction from the seed to distant peers. This prohibits peers from reciprocating data and leads to low sharing ratios and suboptimal utilization of network resources. To recover from these impairments, we propose an enhancement to BitTorrent which aims to minimize the time to download the content and at the same time to enforce cooperation among peers. Our solution considers a restricted neighborhood to reduce routing overhead and to improve throughput, while establishing few connections to remote peers to improve diversity of pieces. With the help of extensive NS-2 simulations, we show that these enhancements to BitTorrent significantly improve the file completion time while fully profiting from the incentives implemented in BitTorrent to enforce fair sharing.

Keywords: BitTorrent, wireless ad hoc networks, neighbor selection, piece selection, completion time, fair sharing.

1 Introduction

Wireless ad hoc networks and P2P file sharing applications are two emerging technologies based on the same paradigm: the P2P paradigm. This paradigm

* This work was supported by the Expeshare (Experience Sharing in Mobile Peer Communities) Project of the Eureka ITEA programme.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 189–203, 2008.
© Springer-Verlag Berlin Heidelberg 2008

aims to establish large scale distributed services without the need for any infrastructure. Within this paradigm, users have symmetric roles. The global service is ensured thanks to their collaboration. In the case of a wireless ad hoc network, the network is a set of wireless nodes with no central administration or base station. Nodes in such a network operate both as routers and hosts. Multi-hop routing approaches are used to ensure connection between distant nodes. For P2P file sharing applications, peers collaborate in downloading data and multimedia content. Each peer shares some of its upload capacity by serving other peers. The global capacity of the system grows then exponentially with the number of peers. Gnutella [5] and BitTorrent [1] are two examples of P2P content sharing applications in the Internet.

Both P2P file sharing applications and wireless ad hoc networks are mature fields of research. They have been studied heavily but separately in the literature. Only few works try to study how they perform together (e.g., [10] [11] [12]). These works focus on the content lookup problem in wireless ad hoc networks without studying the efficiency of the content sharing itself. Studying the performance of file sharing applications over wireless ad hoc networks is challenging because of the diverse constraints imposed by the use of wireless channels. Indeed, as nodes are both routers and end-users, the routing overhead must be taken into consideration. Furthermore, the performance of transport protocols such as TCP drops seriously when multi-hop paths are used. That is why current topology-unaware P2P file sharing applications are not expected to perform well when deployed over wireless ad hoc networks. Designing efficient file sharing solutions for such networks is an important area of research. Indeed, a P2P solution for file sharing has diverse advantages over other data dissemination techniques like multicast in general and this applies to wireless ad hoc networks in particular. For instance, in case of multicast, the construction and update of the virtual topology (tree or mesh) is costly in terms of bandwidth consumption namely in dynamic scenarios. Moreover, the data replication in multicast follows the virtual topology and so nodes like leaves of a tree only receive data and do not spend resources to provide it to other nodes. Thus, no fair cooperation is ensured when using multicast unless constructing a different virtual topology (or tree) per piece of data, which is technically unfeasible.

In this work, we investigate how well a P2P file sharing solution developed for the wired Internet performs over a wireless ad hoc network. Our aim is to come up with a solution that minimizes the content download time while at the same time improving collaboration by enforcing fair sharing among peers. As efficient and fair content sharing is targeted, we choose to adapt BitTorrent [1] as a file sharing protocol given its large usage and its known close to optimal performances in the wired Internet [13]. When data is distributed using BitTorrent, interested peers supply pieces of the data to other peers, reducing the burden on any individual peer, providing redundancy in the network, and reducing dependency on the original seed. In addition, BitTorrent implements incentives that encourage peers to collaborate in downloading the content, which is not the case of multicast-tree based solutions.

In a first effort to understand this problem, we consider the particular case when every ad hoc node is interested in downloading the content. In this case, the underlying topology has a big impact on the performance of BitTorrent. Indeed, any piece sent over a suboptimal route will cause resource consumption in all intermediate nodes. When all nodes are peers, this will affect all peers located on these nodes by stealing bandwidth from them without being able to profit from this transmission since it happens at the routing layer. However, if intermediate nodes are not peers interested in the same content, this suboptimal piece transmission will have less impact on the torrent itself since it does not directly steal bandwidth from peers (it will steal bandwidth from other applications however). Add to this the fact that when all nodes are peers, the traffic generated by the torrent is maximal and an optimization is further required. We aim at well understanding this case and proposing an efficient solution for it before moving into less loaded scenarios in future work namely the scenario where only a part of the nodes are peers. The performance evaluation is done through extensive NS-2 simulations using regular modules for the ad hoc routing and wireless medium and our implementation of BitTorrent in NS-2[1].Our main contributions can be summarized as follows. Ordinary BitTorrent establishes TCP connections with neighbors independently of their location. This choice of neighbors can lead to slow TCP connections due to long multi-hop paths and routing overhead. Sharing can also be bad when using large pieces since complete pieces cannot be sent too far to be reused later by other peers. A first solution is to limit the scope of the neighborhood. In this case, we noticed shorter download times but poor sharing since there is no diversity of pieces in the network. To recover from these impairments, we propose an enhanced variant of BitTorrent, tuned to ad hoc networks, which considers a restricted neighborhood to diminish routing overhead and to improve throughput, while establishing few connections to remote peers to improve diversity of pieces. To implement this, we modify the choking algorithm and add a new piece selection strategy. The simulations show that these enhancements to BitTorrent considerably improve the file completion time while fully benefiting from the incentives implemented in BitTorrent to enforce fair sharing.

Section 2 of this paper presents an overview of the state of the art in deploying P2P solutions over wireless ad hoc networks. Section 3 describes briefly the BitTorrent protocol. The framework of the study is discussed in Section 4. Section 5 shows the importance of the piece size in determining the performance of BitTorrent. Section 6 studies the impact of the scope of the neighborhood. Section 7 presents our enhanced variant of BitTorrent. Section 8 summarizes the work and gives some ideas on our future work.

2 State of the Art

In this section, we present an overview of the state of the art of P2P file sharing applications and their different implementations in wireless ad hoc networks.

[1] NS-2 code and scripts at: http://planete.inria.fr/personnel/Mohamed_Karim.Sbai/BitTorrent/AdaptedBitTorrent.htm

P2P applications in the Internet: There are several design approaches for the construction of P2P overlays over the Internet. One can distinguish between structured and non-structured overlays. This classification is done from the standpoint of resources lookup. In non-structured overlays like Gnutella [5], there is no control on the structure of the overlay. Peers discover each other by flooding the network and by learning from previous sessions. The P2P application in this case is not conscious of the topological location of the other peers. In case of structured overlays, an overlay routing algorithm is introduced to locate the content in the network. Several structured overlay networks have been proposed like CAN [6], Chord [7], Pastry [9] and Tapestry [8]. All of them use Distributed Hash Tables (DHT) in their routing of lookup requests. Such tables allow the lookup to scale logarithmically with the number of nodes in the overlay. Again most of these structured overlays are topology independent. On the other hand, there is BitTorrent [1] that does not concentrate on the information lookup since it uses a centralized tracker to discover neighbors. However, it concentrates on optimal utilization of the network capacity when sharing the file between the different interested peers. Since we are mainly concerned in this work by the data transfer plane, we adopt BitTorrent and we extend it to wireless ad hoc networks. More details on BitTorrent are presented in Section 3.

P2P applications in Mobile Ad hoc NETworks: Both structured and non-structured overlays have been implemented in MANET. Since nodes are both end-users and routers, some cross-layer design approaches have been introduced. These approaches suppose that P2P applications operate both at the network layer and at the application layer. One can divide the design space into four subspaces:

- Non-structured and layered design: Oliviera et al. study in [10] the performance of Gnutella deployed over three ad hoc routing protocols DSR, AODV and DSDV. Their results show that the ratio of delivered packets is lower than those of unicast applications deployed over MANET. This is due to the fact that Gnutella chooses neighbors independently of their locations. The overlay construction is topology independent.
- Non-structured and cross-layer design: The work done by Klemm et al. in [11] proposes to integrate the peer lookup mechanism of a P2P application like Gnutella in the network layer and compares this design to the layered design proposed by Oliviera et Al. They propose ORION that establishes connections on demand through the routing mechanism. The cross-layer lookup implemented by ORION is shown to provide higher successful transfers ratio than in the layered scenario.
- Structured and layered design: A proximity-conscious DHT (Pastry) has been deployed over the DSR routing protocol in [12]. As it is a layered design, there is no interaction between the DHT and the routing protocol. This leads to an overhead in maintaining routes for both the application layer and the network routing layer.
- Structured and cross-layer design: This design is named Ekta by Das et al. in [12]. The functionalities of the Pastery DHT are integrated within the

routing protocol. The main idea is the mapping of the peer identifiers in the same namespace than the IP addresses. Their results show that Ekta is better than the layered design in terms of number of successfully delivered packets.

Former studies on BitTorrent over wireless ad hoc networks: Several works tried to adapt BitTorrent to wireless ad hoc networks (e.g. [14] and [15]). They only focus on the tuning of the peer discovery phase without addressing the efficiency of the content sharing itself. Michiardi et al. study in [4] the performance of a cooperative mechanism to distribute content from one source to a potentially large number of destinations. They propose to deploy BitTorrent with a minor change allowing neighbor discovery and traffic locality. This is done by selecting only near neighbors as effective neighbors. The result is a decrease in the total download time and energy consumption. Their work is relevant to ours; however we go beyond by focusing not only on the download time but also on the sharing among peers which will show to suffer if pieces are only exchanged with close neighbors. The solution we propose in this work is able to improve the sharing ratio and the completion time simultaneously.

3 BitTorrent: A Content Distribution Protocol

BitTorrent (see e.g., [1], [13]) is a scalable P2P content distribution protocol. Each client shares some of its upload bandwidth with other peers interested in the same content in order to increase the global system capacity. Peers cooperating to download the same content form a *torrent*. A peer discovers other peers by contacting a central rendezvous node called *tracker*. The latter stores IP addresses of all peers in the torrent and maintains statistics on uploads and downloads per peer. To facilitate the replication of the content in the network and to ensure multi-sourcing, a file is subdivided into a set of pieces. Each piece is also subdivided into blocks. A peer which has all pieces of the file is called *seed*. When the peer is still downloading pieces, it is called *leecher*. Each peer maintains a peer list. Neighbors are those of this list with whom the peer can open a TCP-connection to exchange data and information. Only four simultaneous outgoing *active* TCP connections are allowed by the protocol. The corresponding neighbors are called *effective neighbors*. They are selected according to the *choking algorithm* of BitTorrent. This algorithm is executed periodically. Once the *choking period* expires, a peer chooses to unchoke the 3 peers uploading to him at the highest rate. It is a best slot unchoking. This strategy, called *tit-for-tat*, ensures reciprocity and enforces collaboration among peers. Now to discover new upload capacities, a peer chooses randomly a fourth peer to unchoke. This unchoking slot is called optimistic slot. All other neighbors are left choked. When unchoked, a peer selects a piece to download using a specific piece selection strategy. This strategy is called *local rarest first*. Indeed, each peer maintains an update-to-date list of pieces owned by all its neighbors. When selecting a piece, a peer chooses the piece with the least redundancy in its neighborhood. In case

of equality, one of the rarest pieces is chosen randomly. Rarest first is supposed to increase the entropy of pieces in the network which enforces collaboration and hence improves global performance.

Here are the performance metrics relevant to BitTorrent that we will use in our study and that are calculated at the end of the experimentation:

- U_{ij}: Total bytes uploaded by peer i to peer j.
- D_{ij}: Total bytes downloaded by peer i from peer j. ($U_{ij} = D_{ji}$)
- R_{ij}: Ratio of sharing between peer i and peer j.

$$R_{ij} = \frac{min(U_{ij}, D_{ij})}{max(U_{ij}, D_{ij})} \tag{1}$$

- N_i: Number of neighbors j of peer i such that $U_{ij} \neq 0$ or $D_{ij} \neq 0$.
- R_i : Sharing ratio for node i.

$$R_i = \frac{1}{N_i}. \sum_{j|\ U_{ij} \neq 0\ or\ D_{ij} \neq 0} R_{ij} \tag{2}$$

- F_i: The finish time of peer i. It is the time by which it receives all pieces of the file.

As we are studying BitTorrent over wireless ad hoc networks where topology matters, we consider some additional performance metrics related to topological positions of peers. The file is supposed to exist at one seed S at the beginning of the session. Our metrics quantify the quality of service perceived by peers as a function of their relative positions with respect to the seed.

- F_h: Average finish time of peers (or nodes) located at h hops from seed S.

$$F_h = \frac{1}{n_h}. \sum_{i|\ H(i)=h} F_i \tag{3}$$

  where n_h is the number of peers located at h hops from seed S and *H(i)* a function that gives the number of hops between any node i and the seed S.
- R_h: Average sharing ratio of peers (or nodes) located at h hops from seed S.

$$R_h = \frac{1}{n_h}. \sum_{i|\ H(i)=h} R_i \tag{4}$$

4 Framework of the Study

We proceed with an experimental approach using the NS-2 simulator. In this section, we describe preliminary changes we made to BitTorrent to allow peer discovery and the exchange of signaling over wireless ad hoc networks. Then, we discuss the stack of protocols we use in our deployment of BitTorrent in NS-2. Finally, we introduce the scenario used in our evaluation.

4.1 Trackerless BitTorrent

Wireless ad hoc networks are infrastructureless. It is convenient that one does not rely on a centralized tracker when applying BitTorrent to such networks. So, we opt in our study for a trackerless approach. Since the most important role of a tracker in the Internet is to provide peers with the identifiers of other peers, we need to introduce a peer discovery mechanism. In our evaluation framework, to discover new peers, a peer floods periodically the network with a HELLO message and waits for HELLO REPLY messages. HELLO messages are transmitted to wireless neighbors with some initial TTL (Time-To-Live) to control the scope of the flood and hence the visibility of a peer. This TTL is a parameter of our study. Receiving a HELLO message, a peer decrements the TTL and forwards it to its wireless neighbors, and so on. The message is not forwarded when its TTL reaches zero.

4.2 Stack of Protocols and Packets Exchanged between Peers

In BitTorrent, peers exchange two types of packets: Data packets and control packets. We choose in our NS-2 implementation to send data packets via TCP connections because reliability and congestion control are needed when transporting blocks of file. However, control packets as for peer discovery and piece updates contain small and urgent information that is better to transport using UDP. Here are the different control packets exchanged between peers:

- **HELLO**: see Section 4.1.
- **HELLO REPLY**: see Section 4.1.
- **UPDATE PIECE LIST**: when a peer receives a new piece, it sends an UPDATE PIECE LIST to all peers with whom it can exchange data.
- **PIECE OFFER REQUEST**: when a peer i unchokes a peer j, it sends a PIECE OFFER REQUEST packet to j. This packet contains the list of pieces that i has already downloaded.
- **PIECE OFFER REPLY**: receiving a PIECE OFFER REQUEST, a peer answers with a PIECE OFFER REPLY packet. After applying the piece selection strategy, it decides whether to accept or to reject the offer. A flag included in the PIECE OFFER REPLY packet indicates this decision (ACCEPT or REJECT). In the case the offer is accepted, the peer indicates the number of the requested piece. During the choking period, many PIECE OFFER REPLY packets can be sent to the offering peer in order to allow the transmission of several pieces.

4.3 The Main Scenario

We consider a network of N nodes (N=40 when not specified) distributed in a plane following a grid topology (10 nodes per row). The distance between two physical neighbors is set to 40 m for a range of wireless transmissions equal to 50m. This ensures connectivity while minimizing interference. At the beginning of each simulation, node 0 located at the top left is the seed and the other nodes

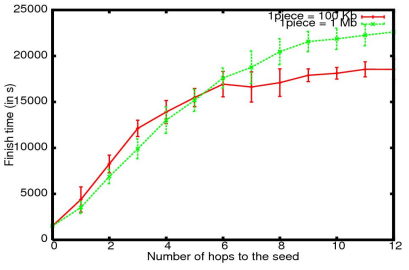


Fig. 1. Average finish time as a function of number of hops to seed

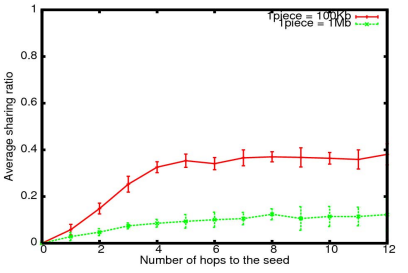


Fig. 2. Average sharing ratio as a function of number of hops to seed

are leechers. The file size is set equal to 10 Mbytes, which is large enough to ensure the convergence of the protocol to equilibrium. All peers start downloading the file at the same time t=1500s by first looking for each other then sharing the pieces of the file according to the BitTorrent algorithms. This time interval skipped at the beginning gives the network enough time to stabilize and calculate its routing tables. The bitTorrent choking algorithm period is taken in our simulations equal to 40s. A piece is subdivided into blocks of size 1KB. Concerning the underlying layers, the nodes connect to each other using the 802.11 MAC Layer with the RTS/CTS-Data/ACK mechanism enabled. The data rate is set to 1 Mb/s. For ad hoc routing, we use the DSDV proactive protocol.

5 Impact of Piece Size

We start by evaluating regular BitTorrent where the overlay is constructed without considering the underlying wireless topology. We give a particular attention to the piece size and to its impact on both the finish time of peers and their sharing ratios. The reason to consider the piece size is that it decides how far pieces can be sent over the network. The TTL of HELLO messages is set to its maximum value so that all peers are neighbors of each other. Two sizes of pieces are used while keeping constant the size of the file. We consider respectively the values 100 blocks and 1000 blocks for small and big size of pieces. Figure 1 plots the average finish time F_h as a function of the number of hops h to the seed for both small and large size of pieces. Each point in this figure is an average over multiple simulations and over all nodes located at the same number of hops to the seed. As expected, the finish time increases as far as we move away from the source. One can notice in the figure that for small pieces, remote peers have better finish time than for large pieces. This is because the range of transmission of small pieces is longer. A remote peer can then receive more pieces in the choking period and share them with others, which improves the reusability of pieces and network resources. This is confirmed in Figure 2 where we plot the average sharing ratio R_h as a function of the number of hops to the seed. It is clear that the sharing ratio in case of small pieces is more important because distant nodes (or peers)

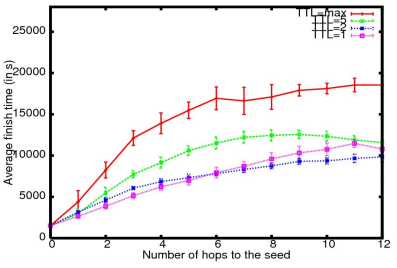


Fig. 3. Average finish time as a function of number of hops to seed for different flooding scope

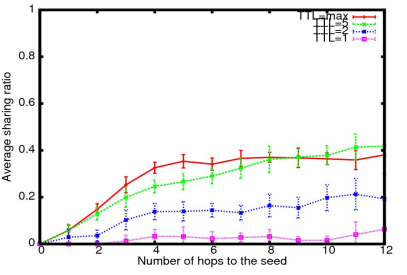


Fig. 4. Average sharing ratio as a function of number of hops to seed for different flooding scope

can now get quickly complete pieces and replicate them in their neighborhood. Unfortunately, this is not the case with large pieces. Large pieces cannot be sent far in the choking period so they propagate in the network as a wave resulting in an under-utilization of network capacity. One can see the case of large pieces as being the absence of sharing between distant nodes and the fact that nodes wait for pieces to arrive to their upstream nodes before obtaining them. The use of small pieces however make the pieces spread over the network, which reduces the finish time and makes the sharing incentives implemented by BitTorrent work better in wireless ad-hoc networks. Our solution supports this modification.

6 Impact of the Scope of the Neighborhood

Another important factor in BitTorrent over wireless ad hoc networks is the scope of the neighborhood. In this section, we study the impact of reducing this scope on both the finish time and the sharing ratio. We run several simulations on the topology described in 4.3 changing each time the flooding scope (TTL) of HELLO messages destined to peer discovery. Figure 3 compares the finish time for TTL=max, 5, 2 and 1. Interestingly, the finish time improves when the neighborhood scope is decreased. This is mainly due to better TCP performance over short paths and to smaller routing overhead. Control packets, namely PIECE UPDATE and HELLO packets, are sent only in the restricted neighborhood. The case TTL=2 is slightly better than the case TTL=1 because of the interference between physical neighbors. Figure 4 plots the average sharing ratio R_h as a function of the number of hops to the seed for the different values of TTL. Unfortunately, we can see that the improvement in finish time when reducing the neighborhood comes at the expense of a lower sharing ratio. The diversity of pieces in the network decreases and the file propagates more or less as a wave in a unique direction from the seed to the farthest nodes. Hence, distant peers can not participate in the replication of pieces, they only wait for pieces to arrive to their physical neighbors to obtain them. Clearly, this is bad for cooperation among peers. An optimal solution should improve the finish time while preserving large values for the sharing ratio.

7 BitTorrent Adapted to Wireless Ad Hoc Networks

The main objective of our variant of BitTorrent is to profit from the advantages of the limited neighborhood, namely the good performance of TCP on short paths, the reduced routing overhead, and the reduced load of flooding control packets. At the same time, we aim at improving the sharing ratio and the reusability of network resources by creating diversity of pieces in the network. Our main idea is to create few TCP connections to distant peers in addition to those with close peers. Pieces can then spread over the network and propagate in different directions, which improves the sharing and the download completion time. With this modification, several zones of the network can be active simultaneously, which is not the case of the wave generated by regular BitTorrent with limited neighborhood. To implement this idea, we tune BitTorrent to support the distinction between remote and close peers. The new choking algorithm is aware of the location of peers by using routing information. It distributes optimistic unchokes between remote and close peers and adds a specific neighbor selection mechanism to select a distant peer. It also applies a new piece selection strategy when the peer offering the piece is distant. Unlike BitTorrent with limited neighborhood, this modification requires a global knowledge about the identifiers of peers in the network. We propose that each peer maintains two neighbor tables: NEARBY NEIGHBORS TABLE (NNT) and FAR NEIGHBORS TABLE (FNT). When discovering new peers, neighbors whose number of hops is less than or equal to 2 are added to NNT. Other peers belong to FNT. The PIECE UPDATE packets are sent only to neighbors in NNT. Peers do not need to know about all pieces in the network as their piece selection strategy operates only on their NNTs. Indeed, in wireless networks, the replication of pieces is more efficient when it is based on statistics in the close neighborhood since this guarantees a faster local replication compared to when statistics are based on a large neighborhood. As in BitTorrent, when the choking algorithm is executed, three best uploaders are selected as effective neighbors. These three neighbors are chosen from both nearby and far neighbor tables. The peer then serves these three neighbors during the next choking period. But in addition to these effective neighbors, the peer selects a fourth random neighbor from one of the two tables (optimistic slot). The table from which it selects the neighbor is decided by a round robin policy that guarantees an optimal balance between the random unchokes locally and the transmission of pieces to distant neighbors in order to improve diversity. For a succession of optimistic unchokes, the peer selects a peer one time from FNT, q times from NNT and so on. In our protocol, the quantum q represents the ratio of the number of time slots spent on serving nearby neighbors and those for serving far neighbors. It is also the number of slots that a peer should wait before unchoking a distant neighbor again. Our simulations indicate that the choice of this quantum is fundamental in deciding the performance of our solution. Furthermore, the strategies of selecting pieces proposed by distant neighbors and selecting effective neighbors from FNT should differ from the ordinary strategies applied by BitTorrent because the objective of our version of BitTorrent in unchoking far peers is mainly to improve diversity. The

next paragraphs explain the different selection strategies we implement in our solution. The following ones study the performance of the enhanced BitTorrent and discuss the choice of the quantum q.

7.1 Selecting a Far Neighbor at Random

When a regular BitTorrent client decides to optimistically unchoke a peer, it selects it at random with a uniform probability. In wireless networks however, the gain we get from optimistic unchoking in terms of diversity increases with the number of hops. So a peer has more interest in unchoking a farther peer than another one closer to it. Thus, in our adapted version of BitTorrent, to select a far peer to unchoke from FNT, the peer starts by selecting the number of hops to that peer with a probability that increases linearly with the number of hops. Let h_m be the maximum number of hops seen by the peer. We suppose that FNT contains only peers at h_m and $h_m - 1$ hops. These are the farthest peers that if we send pieces to them, we are sure of having the largest gain in terms of diversity and reutilization of network resources.[2] It follows that the number of hops is first selected using a probability function p given by this formula:

$$p(h) = \begin{cases} \frac{h}{h_m+(h_m-1)} & \text{if } h \geq h_m - 1 \\ 0 & \text{else} \end{cases}.$$

When the number of hops h is chosen, the peer then selects, in a uniform random way, a peer among those located at h hops from it as the peer to optimistically unchoke.

7.2 Selecting a Nearby Neighbor at Random

When the peer needs to select a nearby neighbor, it chooses a node from NNT in a uniform random way. A nearby neighbor is supposed to replicate the pieces it receives in its two-hop limited neighborhood[3]. This replication is fast since the TCP protocol has a good throughput over short paths.

7.3 Piece Selection Strategy When the Offering Neighbor Is Far

When receiving a piece offer from a P2P neighbor, the peer checks the number of hops to the offering neighbor. If it is greater than 2, it considers that it is an offer from a far node. In this case, a specific piece selection strategy is applied in order to select the best piece to download from this node. This strategy will be called the *absent piece strategy*. The peer first computes the redundancy of the offered pieces in its close neighbors table and in its piece pool. At the opposite

[2] We add peers at h_{m-1} hops to FNT in order to reduce the load on the one or few peers located at h_m hops.

[3] We form NNT using two-hop neighborhood because according to results in Section 6, this leads to slightly better finish time and sharing than if limiting the neighborhood to only one hop.

of BitTorrent, the candidate pieces will be those with zero redundancy (no need to download a piece from a distant node if it exists at less than two hops). So a piece can be accepted only if neither the peer nor one of its near neighbors has downloaded it before. In case of multiple absent pieces, one piece among them is chosen in a uniform random way. The absent piece can then be replicated quickly in the near neighborhood. If no absent piece is noticed, the peer sends a REJECT in the piece offer reply packet. In summary, our solution supposes that it is better to download a piece existing in the nearby neighborhood from a nearby neighbor. Only absent pieces are taken from far neighbors so as to reduce the routing overhead. This strategy is fundamental for getting good performances with our variant of BitTorrent.

7.4 Piece Selection Strategy When the Offering Neighbor Is Near

Local rarest first is used when the peer receives a piece offer from one of its nearby neighbors. Pieces with the least number of copies in the close neighborhood are selected. This is the normal behavior of the standard version of BitTorrent but only applied in the two-hop neighborhood. Here the throughput of TCP is good and the routing overhead is almost inexistent so we can allow ourselves to apply the rarest first policy that guarantees the fast replication of pieces.

7.5 Simulation Results

To study the performance of our solution, we run several NS-2 simulations over the previously described topology. We vary the values of the quantum q and observe the behavior of the download finish times of peers and their sharing ratios. Figure 5 compares finish time of ordinary BitTorrent with limited neighborhood (TTL = 2) with our version of BitTorrent using different values of the quantum q (q=3, 2 and 1). Each curve presents the average finish time F_h as a function of the number of hops to the seed. Recall that the role of q is to balance optimistic unchokes between close and remote peers. The larger the q, the smaller the number of unchokes to remote peers. The finish time for our solution is better and more equally distributed since far nodes can receive pieces from the beginning of the session and can replicate them in their close neighborhoods. Our solution limits the number of pieces sent to far nodes in order to reduce the routing overhead. This creates parallel areas of activity in the network. Far nodes do not need to wait for pieces to arrive to their neighborhoods to download them. Hence, pieces propagate in the network in all directions. This observation is illustrated in Figure 6 which compares sharing ratios of ordinary BitTorrent with limited neighborhood (TTL=2) with our variant of BitTorrent using different values of the quantum. Each curve presents the average sharing ratio R_h as a function of number of hops to the seed. Figure 6 shows that the strategies used in our solution increase considerably the sharing ratios of all peers. This is due to the diversity created by sending original pieces to distant nodes. So, sharing incentives work well in this context and the distribution is less vulnerable to the selfishness of some nodes. Our results also show that a quantum equal to 1

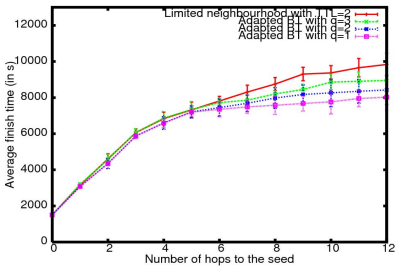


Fig. 5. Average finish time for our enhanced BitTorrent compared to ordinary BitTorrent with limited neighborhood

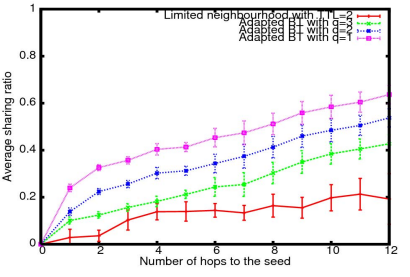


Fig. 6. Average sharing for our enhanced BitTorrent compared to BitTorrent with limited neighborhood

gives a better finish time and a better sharing ratio in our setting. Clearly, the performance of our solution depends on the choice of the quantum q. This choice is treated in the next section.

7.6 Optimal Choice of the Quantum q

In this paragraph, we establish an empirical formula for q and then validate it through simulations. Let h_m be the maximum length of a path between two nodes in the network. Let α_i be the number of pieces that can be sent during a choking slot to a node located at i hops. The objective of our balanced optimistic unchoking strategy is to send a copy of each piece to the end of the network and wait for it to return to the middle of the network. Forward and backward pieces meet then in the middle of the network, which guarantees the best gain. If there were only one piece in the file, only one seed and the content is sent to the farthest node, the piece will take approximately $\frac{h_m}{2}$ slots to return to the middle of the network. Now when the file contains several pieces, the node should wait $\frac{\alpha_{h_m}}{\alpha_1}.\frac{h_m}{2}$ before unchoking the farthest node again. It is the number of slots needed for the α_{h_m} pieces to return to the middle of the network hop by hop. Now, if all peers in the network are interested in the content and if we assume nodes to be uniformly distributed in the plane, $\frac{N}{2}$ nodes at maximum can participate in sending pieces to the farthest node. So one needs to increase the waiting time by a factor of $\frac{N}{2}$. So, the formula approximating q will be:

$$q = \frac{\alpha_{h_m}}{\alpha_1}.\frac{h_m}{2}.\frac{N}{2} \tag{5}$$

To validate this formula, we vary the number of nodes and observe how this impacts the optimal choice of q. We plot optimal q as a function of the number of nodes N. Simulations are done on grid topologies with N=20 to 100. Figure 7 plots the average finish time over all nodes as a function of the chosen quantum q for 40 nodes and 80 nodes (curves for all values of N are not included for clarity of presentation). Figure 8 plots both the computed and simulation results for

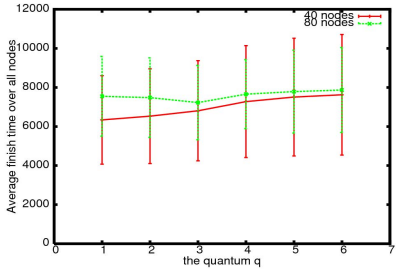


Fig. 7. Average finish time as a function of the chosen quantum

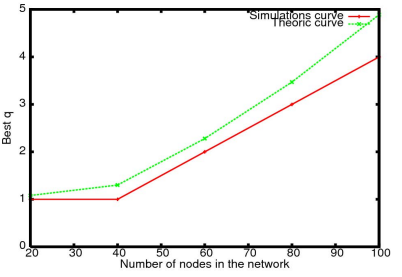


Fig. 8. Best quantum as a function of the number of nodes

best q. The values of α_{h_m} and α_1 are taken from simulations in both curves. Even though our expression for q is simple and approximate; we can see a good match between the two curves. In the figure, simulation values of the best q are rounded integer values of theoretical ones. Thus, the above formula describes well the behavior of the optimal q when number of nodes varies. One can notice that this quantum increases with N, which means less pieces sent by each peer to remote peers for larger networks.

8 Conclusions and Perspectives

P2P data sharing applications in wireless ad hoc networks should provide good quality of service to their users in terms of finish time and sharing. There is a high potential for these applications but unfortunately, the wireless nature of the network imposes many constraints to be taken into consideration before using regular applications tuned for the wired Internet. Solutions that reduce neighborhood scope allow better finish time than those with random graphs of communications. Nevertheless, limiting the neighborhood is shown, in this paper, to be dangerous in terms of reducing sharing ratios between peers. The solution we propose in this paper finds a good management of neighbor and piece selection that reduces finish time and encourages sharing. A peer concentrates on its nearby peers with few connections to far ones. When far neighbors are selected, a special piece selection strategy named absent piece strategy comes into effect. Simulation results show a decrease in service time and a great improve in sharing ratios. Our future work will be on adapting our solution to mobile scenarios. High dynamicity of such networks will open the way to new interesting problems.

References

1. BitTorrent protocol, http://wiki.theory.org/BitTorrentSpecification
2. NS: The Network Simulator, http://www.isi.edu/nsnam/ns/
3. Ding, G., Bhargava, B.: Peer-to-Peer File-Sharing over Mobile Ad hoc Networks. In: IEEE PERCOM-W, Orlando, USA (2004)

4. Michiardi, P., Urvoy-Keller, G.: Performance analysis of cooperative content distribution for wireless ad hoc networks. In: WONS 2007, Obergurgl (2007)
5. The Gnutella specification (2000), http://dss.clip2.com/GnutellaProtocol04.pdf
6. Ratnasamy, S., Francis, P., Handley, M., Karp, R., shenker, S.: A scalable content-addressable networks. In: ACM SIGCOMM (2001)
7. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications. In: ACM SIGCOMM (2001)
8. Zhao, B.Y., Kubiatowicz, J.D., Joseph, A.D.: Tapestry: an infrastructure for fault-resilient wide-area location and routing. T.R. UCB//CSD-01-1141, U.C.Berkeley (2001)
9. Rowstron, A., Druschel, P.: Pastry: scalable, diributed object location and routing for large -scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218. Springer, Heidelberg (2001)
10. Oliviera, L.B., Siqueira, I.G., Loureiro, A.A.: Evaluation of ad hoc routing protocols under a peer-to-peer application. In: WCNC (2003)
11. Klemm, A., Lindermann, C., Waldhorst, O.: A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In: VTC (2003)
12. Das, S.M., Pucha, H., Hu, Y.C.: Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks. TR-ECE-04-04, Purdue University (2004)
13. Legout, A., Urvoy-Keller, G., Michiardi, P.: Rarest First and Choke Algorithms Are Enough. In: IMC 2006, Rio de Janeiro (2006)
14. Nandan, A., Das, S., Pau, G., Gerla, M.: Cooperative downloading in vehicular ad hoc networks. In: WONS, Washington, USA (2005)
15. Rajagopalan, S., Shen, C.-C.: A cross-Layer Decentralized BitTorrent for Mobile Ad hoc Networks. In: MOBIQUITOUS, San Jose, USA (2006)

Optimal Gathering Algorithms in Multi-hop Radio Tree-Networks ith Interferences

Jean-Claude Bermond[1,⋆] and Min-Li Yu[2,⋆⋆]

[1] MASCOTTE, joint project CNRS-INRIA-UNSA,
2004 Route des Lucioles, BP 93, F-06902 Sophia-Antipolis, France
bermond@sophia.inria.fr
[2] University College of the Fraser Valley, Department of Mathematics and Statistics,
Abbotsford, BC, Canada V2S 4N2
joseph.yu@ucfv.ca

Abstract. We study the problem of gathering information from the nodes of a multi-hop radio network into a pre-defined destination node under the interference constraints. In such a network, a message can only be properly received if there is no interference from another message being simultaneously transmitted. The network is modeled as a graph, where the vertices represent the nodes and the edges, the possible communications. The interference constraint is modeled by a fixed integer $\mathsf{d}_I \geq 1$, which implies that nodes within distance d_I in the graph from one sender cannot receive messages from another node. In this paper, we suppose that it takes one unit of time (slot) to transmit a unit-length message. A step (or round) consists of a set of non interfering (compatible) calls and uses one slot. We present optimal algorithms that give minimum number of steps (delay) for the gathering problem with buffering possibility, when the network is a tree, the root is the destination and $\mathsf{d}_I = 1$. In fact we study the equivalent personalized broadcasting problem instead.

1 Introduction

1.1 Problem Statement

The problem we consider in this paper was motivated by a question asked by France Telecom about "how to provide Internet connection to a village" (see [6]) and is related to the following scenario. Suppose we are given a set of communication devices placed in houses in a village (for instance, network interfaces that connect computers to the Internet). They require access to a

⋆ Partially supported by the CRC CORSO with France Telecom , by the European FET project AEOLUS, and by the INRIA associated team RESEAUXCOM with S.F.U.

⋆⋆ Partially supported by the Natural Sciences and Engineering Research Council of Canada and by the INRIA associated team RESEAUXCOM with S.F.U.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 204–217, 2008.
© Springer-Verlag Berlin Heidelberg 2008

gateway (for instance, a satellite antenna) to send and receive data through a multi-hop wireless network. In this network, the devices communicate exclusively by means of radio transmissions, referred to as *calls*. A call involves a message and two devices, the *sender* and the *receiver*. The communication is subject to the following technological constraints:

Reachability constraint: In order to be reached by a call, the receiver of this call must be within reachability distance of the sender.

Interference constraint: A call may interfere with calls that are in the neighborhood of the receiver, or a message can be properly received only if no other senders are in the neighborhood of the receiver.

***t*-gathering problem:** Suppose each device of the network has a piece of information. The t-gathering consists of collecting (gathering) all these pieces of information into a special device t, called the *gathering node*, by the means of calls subject to the two constraints described before. The t-gathering problem is to realize such a constrained gathering without concatenating messages and with the minimum delay.

An equivalent formulation is the so-called.

***s*-personali ed broadcast:** Here a single device (the gateway in the problem of France Telecom) called source s has a different piece of information to broadcast to every other device in the network by the means of calls subject to the two constraints described before. The s-personalized broadcast is to realize such a constrained gathering without concatenating messages and with the minimum delay.

A slight variation of this problem has received much attention in the context of sensor networks. In such networks, each device contains a sensor and the gathering problem corresponds to the situation where information collected at each sensor has to be gathered to a single central device (base station). However, most of the articles are concerned with minimizing the energy consumption and allow aggregation of data. The work which is most related to ours is [11], in which reachability and interference constraints are also assumed, but most of its results apply for the case of directional antennas.

1.2 Model and Assumptions

According to the model adopted in [2], the network described above is represented by an undirected graph $G = (V, E)$, where V is the set of nodes, each of which representing a communication device, and E is the set of edges, representing the pairs of nodes involved in possible calls. There is a special pre-defined node s called the source (sink in the gathering case). Let $d_G(u, v)$ indicate the distance in G, defined as the length of a shortest path between u and v. We model the reachability and the interference constraints by two positive integers, respectively $d_T \geq 1$ and $d_I \geq d_T$. An important case is $d_T = 1$, which means that a node is able to communicate only with its neighbors in the graph (or equivalently G is the communication graph). The second parameter d_I models the interference constraint as

follows: if a receiver is within distance d_I from a sender, then this node cannot receive any other message. If u sends a message m to v, then the call (u, v) interferes with every node $w \in V$ such that $d_G(u, w) \leq d_I$. Two calls are said to be *compatible* if they do not interfere with each other (otherwise, they are *incompatible*). More precisely, two calls (s_1, r_1) and (s_2, r_2), for $r_1, r_2, s_1, s_2 \in V$, are compatible if $d_G(s_1, r_2) > d_I$ and $d_G(s_2, r_1) > d_I$. Observe that one of the consequences of the interference constraint is that $s_1 \neq r_2$ and $s_2 \neq r_1$, which implies that a node is not able to send and receive messages simultaneously. A *step* (*round*) is a set of compatible calls. We assume that every occurrence of a call takes one unit of time (or one slot) and involves a one unit-length message.We also assume that buffering is possible in intermediate nodes.

In this paper, our aim is to find efficient algorithms that give optimal solutions for the s-personalized broadcast problem when $d_T = d_I = 1$ and G is a tree.

1.3 Related Work

The broadcasting and gossiping problems have been widely studied for wired networks (see [15]), including models that assume no concatenation of messages (see [4]). For radio networks, the case when $d_I = 1$ is studied only for broadcasting in [10,12] and gossiping in [8,9,14]. Note that broadcasting is different from our problem which is personalized broadcasting, as in the process of broadcast, the same information has to be transmitted to all the other nodes and so flooding techniques can be used. Recently the gathering problem has gained much attention. In [2], assuming an arbitrary size of information in each node, a protocol for general graphs with an approximation factor of at most 4 is presented. It is also shown that the problem of finding an optimal gathering protocol does not admit a Fully Polynomial Time Approximation Scheme if $d_I > d_T$, unless P=NP , and is NP-hard if $d_I = d_T$. In the case where each node has exactly one unit of information to transmit (or to receive which is the case we consider), the problem is NP-hard if $d_I > d_T$ but the complexity is unknown for $d_I = d_T$. An extension of the problem where messages can be released over time is considered in [7] and a 4-approximation algorithm is presented. In [5], optimal solutions are provided for the two-dimensional square grid with $d_T = 1$. In [1] the case of a path is considered for $d_T = 1$ and any d_I. The problem is solved when the sink (source) is at one end of the path and only partly solved when the sink is in the middle of the path.

As mentioned before, sensor networks have been the subject of many papers. But, most of them deal with minimizing the energy consumption or maximizing the life time of the sensor network. In [11] they minimize the delay but their model is slightly different from ours as each node is equipped with directional antennas and no buffering capacity is available in the nodes. Furthermore they only suppose that a node cannot receive and send simultaneously, and more precisely, this corresponds to the case in our model when $d_T = 1$, interference distance is zero and each node is not allowed to receive more than one message at a time. Under their assumptions, they give optimal (polynomial) gathering protocols for path and tree networks. Their work has been extended

to general graphs in [13] for unitary messages. In [3], a companion paper to that one, the same problem as ours is considered, but no buffering is allowed. Finally, another related model can be found in [16], where the authors study the case in which steady-state flow demands between each pair of nodes have to be satisfied.

1.4 Main Result

In this paper, we deal with the situation when G is a tree T with N vertices and with a source (or root) s and $d_T = d_I = 1$ which can be viewed as a generalization of the results of [11] and [13]. In their case the only constraint is that a node cannot receive and transmit at the same time (which can be viewed as $d_I = 0$). They proved that the minimum number of steps is either $N-1$ or $2n_1 - 1$ where n_1 is the size of the biggest subtree.

Here we need to consider not only subtrees, but also subsubtrees. Indeed, when $d_I = 1$, two calls in two different branches are incompatible only if they have the same sender. If two calls (s_1, r_1) and (s_2, r_2) in the same path are incompatible and the arcs are in the order: $s, \ldots, s_1, r_1, \ldots, s_2, r_2, \ldots$, then $d(r_1, s_2) \leq 1$. Otherwise two calls in the same path are compatible if they are separated by at least two arcs.

Here we will have roughly three different forms of trees. Either the tree looks like a path with a big sub-sub-tree formed by the vertices at distance ≥ 2 from s, in which case we will need roughly 3 times the size of this big sub-component. Or the tree has only a big component but inside this component the sub-components are somewhat balanced in which case we need roughly 2 times the size of this big component.In the remaining case (balanced tree an example being a spider (generalized star) we need $N-1$ steps.

To state more precisely our main result, let assume that $deg(s) = m$. Let r_1, r_2, ..., r_m be the neighbors of s, and T_i be the subtree of T with root r_i, where $1 \leq i \leq m$. The size of T_i is simply $|T_i| = n_i$. Similarly let $r_{i,j}$ be the neighbors of r_i and $T_{i,j}$ be the subtree with root $r_{i,j}$. The size of $T_{i,j}$ will be denoted by $|T_{i,j}| = n_{i,j}$. Furthermore, we will assume that the $T_{i,j}$'s are ordered according to their sizes. So $n_{i,1} = max\ n_{i,j}$

Let $M_i = max\{2n_i - 1, n_i + 2n_{i,1} - 1\}$. For the rest of the paper, subtrees are ordered according to the values of M_i: $M_1 \geq M_2 \geq M_3 \geq \ldots \geq M_m$. In case of equality the order is determined by the sizes.

Theorem 1. *When $d_T = d_I = 1$ and T is a tree, the minimum number of steps to complete a personalized broadcasting (or gathering) is equal to* $\max\{N - 1, M_1 + \epsilon\}$, *where $\epsilon = 1$ if $M_1 = M_2$ and 0 otherwise.*

Although the lower bound is easy to prove and the minimum time can be expressed in a simple formula, in order to obtain optimal algorithms many different situations are needed to be considered and a lot of experiments were performed before the arrival to the final optimal algorithms.

2 Lower Bounds and Basic Algorithms

For the rest of the paper we will simply denote by $g(T)$ (instead of $g(T, s, d_T, d_I)$ used in [2]) the minimum number of steps required to complete the personalized broadcast from s (gathering to s) of one unitary message to each node of T under the interference constraint defined by $d_I = 1$.

2.1 Lower Bounds

Proposition 1. $g(T) \geq max\{N-1, M_1 + \epsilon\}$.

Proof. We exhibit different sets of incompatible calls which must be scheduled in different steps (or rounds).

Consider the calls on the arcs (s, r_i) and they are all incompatible and there are $N-1$ of them, as this is the number of messages needed to be sent by the source. So $N-1$ is a lower bound for $g(T)$.

Similarly, for each i, the n_i calls on the arc (s, r_i) and the $n_i - 1$ arcs leaving r_i, are all incompatible. Their number is $2n_i - 1$. So $2n_i - 1$ is a lower bound for $g(T)$.

Consider also the following incompatible calls : those on the arc (s, r_i) and there are n_i of them, the $n_{i,1}$ calls on the arc $(r_i, r_{i,1})$, and the $n_{i,1} - 1$ on the arcs leaving $r_{i,1}$. Altogether we have $n_i + 2n_{i,1} - 1$ incompatible calls and this is also a lower bound for $g(T)$.

Hence, M_i and therefore M_1 is a lower bound. If $M_1 = M_2$, then any algorithm starts calling one of r_1 or r_2 only at step 2 or after, and so it needs at least $M_1 + 1$ steps.

In the next subsections, we present algorithms that perform personalized broadcasting, which will give optimal solutions when there is only one subtree and will also be used for the general case, in particular when there are two subtrees, by applying them to each subtree. We describe the algorithms for one subtree T_i rooted in r_i. We call T_i a type 1 subtree if $M_i = 2n_i - 1$. Otherwise, it is called a type 2 subtree.

2.2 CASE 1: T_i Is a Subtree of Type 1

We first present an algorithm for a type 1 subtree T_i. In this case recall that $M_i = 2n_i - 1$.

Let X^t denote the set of vertices to which the source has sent a message before step t (that is at the end of step $t-1$) and let T_i^t be the subtree obtained from T_i by deleting X^t. Similarly denote by $T_{i,j}^t$ the component obtained from $T_{i,j}$ by deleting the vertices of X^t. Let $n_i^t = |T_i^t|$ and $n_{i,j}^t = |T_{i,j}^t|$.

The idea of the algorithm is the following: the source sends every odd step to r_i a message destinated to a leaf of a big component of T_i, in order to guarantee that at any step there is no component having more than half of the vertices (or $n_{i,j}^t \leq n_i^t/2$). Also in two consecutive odd steps, the source will send to different components of T_i in order to be able to do compatible calls efficiently

in even steps in different components. We first describe the algorithm, then use an example to illustrate it and finally we prove that it is valid and takes M_i steps (which is the lower bound as $M_i \geq N - 1 = n_i - 1$).

Algorithm A: Personalized broadcasting for a subtree of type 1
At the beginning $X^1 = \emptyset$ and $T_i^1 = T_i$.
- During an odd step $t = 2k - 1$, $k = 1, 2, \ldots, n_i$

Let T_{i,j_k}^t be the largest component of T_i^t not chosen at the preceding odd step (that is $j_k \neq j_{k-1}$) and let x_k be a leaf in this component. The source s sends the message m_k for x_k on the arc (s, r_i). Then we update $X^{t+1} = X^t \cup x_k$ and $T_i^{t+1} = T_i^t - x_k$.

During the odd steps, both r_i and the $r_{i,j}$ are inactive.

Finally any vertex at distance ≥ 3 from the source forwards immediately the message received at the preceding step except when it is the destination, in which case the message is stored (if it is m_l with destination x_l, then the message is forwarded to its neighbor on the path to x_l).
- During an even step $t = 2k$, $k = 1, 2, \ldots, n_i - 1$

- r_i sends to r_{i,j_k} the message m_k received at step $2k-1$ with the destination x_k in T_{i,j_k}^t.

- $r_{i,j_{k-1}}$ sends the message m_{k-1} (received at step $2k - 2$) to its neighbor on the path to x_{k-1}, except when it is the destination, the message is just stored.

- Any vertex at distance ≥ 3 from the source forwards immediately the message received at the preceding step except when it is the destination, in which case the message is stored.

Example: Table 2.2 illustrates how algorithm A works when it is applied to the type 1 tree given in Fig. 1.

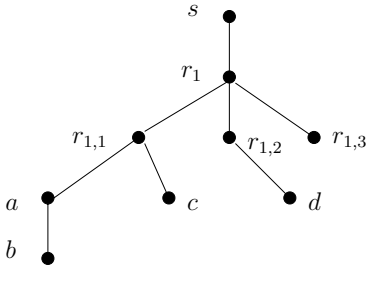


Fig. 1.

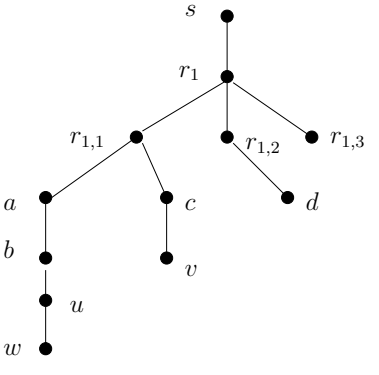


Fig. 2.

Here, $N = 9$, $n_1 = 8$ and $n_{1,1} = 4$. As $M_1 = 15 = 2n_1 - 1 = n_1 + 2n_{1,1} - 1$, it is a type 1 tree. At step 1, s sends a message destinated to a leaf in $T_{1,1}$ (the largest component), for example $x_1 = b$ (we could have chosen c). So $m_1 = m(b)$, the message destinated to b. At step 2, r_1 sends m_1 to $r_{1,1}$. At step 3, s sends a message destinated to a leaf in the largest component different from $T_{1,1}^3$, namely

Table 1. Personalized broadcasting on the tree in Fig.1 with source s using Algorithm A

step	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
1	$s \to r_1$							
2	$r_1 \to r_{1,1}$							
3		$s \to r_1$						
4	$r_{1,1} \to a$	$r_1 \to r_{1,2}$						
5	$a \to b$		$s \to r_1$					
6		$r_{1,2} \to d$	$r_1 \to r_{1,1}$					
7				$s \to r_1$				
8			$r_{1,1} \to a$	$r_1 \to r_{1,2}$				
9					$s \to r_1$			
10					$r_1 \to r_{1,1}$			
11						$s \to r_1$		
12					$r_{1,1} \to c$	$r_1 \to r_{1,3}$		
13							$s \to r_1$	
14							$r_1 \to r_{1,1}$	
15								$s \to r_1$

$T^3_{1,2}$ and the only choice is $x_2 = d$. At step 4, r_1 sends to $r_{1,2}$ $m_2 = m(d)$ and $r_{1,1}$ sends m_1 to a (its neighbor on the path to b). At step 5, s sends a message destinated to a leaf in $T^5_{1,1}$ (the largest component), for example $x_3 = a$ (we could have chosen c). Also a, which is at distance 3 from s, forwards m_1 to b where it is stored. The other steps are described in table 2.2: we have $x_4 = r_{1,2}$ (we could have chosen $r_{1,3}$), $x_5 = c$, $x_6 = r_{1,3}$, $x_7 = r_{1,1}$ and $x_8 = r_1$. Therefore, $m_1 = m(b)$, $m_2 = m(d)$, $m_3 = m(a)$, $m_4 = m(r_{1,2})$, $m_5 = m(c)$, $m_6 = m(r_{1,3})$, $m_7 = m(r_{1,1})$ and $m_8 = m(r_1)$.

Proposition 2. *Algorithm A is valid, i.e. all the calls are compatible.*

Proof. Consider a call with a sender s and it happens in an odd step. As r_i and $r_{i,j}$ are inactive, only the source is sending among the vertices at distance at most 2 from s and so this call is compatible with the others calls whose senders are at distance ≥ 3.

Now consider a call with a sender r_i and it must happen in an even step. Suppose it is a call done at step $2k$ from r_i to r_{i,j_k}. This call is compatible with the other calls in the component T_{i,j_k}, as they involve senders at distance at least 4 from s. Indeed the preceding messages in T_{i,j_k} have been sent at step at most $2k-4$ from r_i to r_{i,j_k} and at step at most $2k-2$ from r_{i,j_k} to a neighbor and then forwarded. Therefore they either arrived at the destinations or at a vertex with distance at least 4 from s. They are also compatible with the calls in other components as none of them involve r_i.

If two calls are in different components $T_{i,j}$, then they are compatible as the distance from a sender to a receiver of the other call is at least 3. Finally two calls with senders in the same component $T_{i,j}$ are compatible and this follows from the fact that they are sent by $r_{i,j}$ within two steps differing by at least 4, as the same component cannot be chosen in two consecutive even steps. Because the distance between two such senders is at least 4, the distance between a sender and the other receiver is at least $3 > 1 = d_I$.

Proposition 3. *At the end of the $M_i = 2n_i - 1$ steps of the algorithm A all the vertices of T_i have received their own messages and so the gathering time is $M_i = 2n_i - 1$.*

Proof. We first prove that at any step there is no component $T^t_{i,j}$ such that $n^t_{i,j} > \frac{n^t_i}{2}$. Indeed, it is true at step $t = 1$ as indeed T_i is type 1, $2n^1_{i,1} \leq n_i$. Suppose that the property is not true and let $t_0 = 2k_0 - 1$ be the first step at which it happens. Then there exists such a component of size strictly bigger than $\frac{n^{t_0}_i}{2}$. Hence, in the two preceding odd steps, this component was the biggest one and it should have been chosen in one of these two steps, and therefore, this component was already of size bigger than half at step $t_0 - 2 = 2k_0 - 3$ or $t_0 - 4 = 2k_0 - 5$ contradicting the choice of k_0.

Therefore at any step $t = 2k - 1$ there is a new vertex x_k to which a message can be sent. Hence, all the messages have been sent by the source at end of step $M_i = 2n_i - 1$.

Consider a message m_k which is sent by s at step $2k - 1$. If $k = n_i$ this is the last message with destination r_i and it arrives at step $2n_i - 1 = M_i$.

Otherwise r_i sends m_k at step $2k$ to r_{i,j_k}. If r_{i,j_k} is its destination, then it arrives at step $2k \leq 2n_i - 2 < M_i$, as $k < n_i$. Otherwise, m_k is sent by r_{i,j_k} on the path to x_k at step $2k + 2$ and then forwarded immediately till it reaches x_k. Let $d(s, x_k)$ be the distance between s and x_k. Note that $d(s, x_k) \geq 3$. The messages with destination on the path from s to x_k are all sent after x_k (otherwise we would have not chosen a leaf contradicting the algorithm). Therefore $k \leq n_i - d(s, x_k) + 1$. Finally m_k is received by x_k at step $2k + d(s, x_k) - 1 \leq 2n_i - d(s, x_k) + 1 \leq 2n_i - 2 = M_i - 1$ as $d(s, x_k) \geq 3$.

2.3 CASE 2: T_i Is a Subtree of Type 2

Here $M_i = n_i + 2n_{i,1} - 1$. So there is a component $T_{i,1}$ such that $2n_{i,1} > n_i$. The idea consists in considering a set of vertices S_i in this component such that the subtree T^*_i obtained by deleting them is of type 1 and then to apply algorithm A to $T^*_i = T_i - S_i$. For the vertices of S_i note that, in the formula for M_i, they are counted for 3. So we will send the messages destinated to them each 3 steps.

A natural way will be to send to the vertices of S_i during the first $3|S_i|$ steps of the algorithm: the source sends first a message to them at steps $3h$, where $0 \leq h \leq n_i - n^*_i - 1$ and then the message is forwarded immediately till it reaches the destination. This algorithm can be also viewed in an inductive fashion: take a leaf u in $T_{i,1}$; at step 1, the source sends to r_i the message to u and then the message is immediately forwarded; at step 2, (r_i sends it to $r_{i,1}$ and so on); at step 4 we apply the algorithm to the tree $T - u$ using either induction or the algorithm A if $T - u$ is of type 1.

This idea works perfectly for one subtree and will be in fact used later for 3 or more subtrees in Section 4.3. But unfortunately it does not lead to a solution in all the cases. For example suppose we have two subtrees. If T_1 is of type 1, then the source will send every odd step. Assume that T_2 is of type 2 with $M_2 = M_1 - 1$; so the source should first send to it at step 2. But then after 3

steps, the source has to send again at step 5. however, s is in fact busy sending to T_1 in this step.

So we will proceed in a different manner by first sending to vertices in T_i^* using Algorithm A, and then use what we call a 3-step extension to send to the rest of vertices by pushing the messages along some paths. So, messages arrive in the leaves only at the last steps of the algorithm. In fact if one thinks in terms of gathering (where the algorithm is the reverse of that for personalized broadcasting) it is more natural to send first the messages from vertices far away that are those from $S_i = T_i - T_i^*$.

We develop an algorithm that proceeds in 2 phases. In the first phase, each vertex receives an integer label which indicates the step in which this message will be sent by the source in the second phase. Therefore, in the second phase, the source will use the information from the labels given in the previous phase to send the proper message at each step. The algorithm is described below and will then be illustrated by an example. We will prove that it is valid and takes M_i steps (which is the lower bound as $M_i \geq N - 1 = n_i - 1$).

Algorithm B: Personalized broadcasting for a subtree of type 2

More precisely, let S_i be a set of σ_i vertices of $T_{i,1}$ such that, after deletion, we obtain a tree $T_i^* = T_i - S_i$ with $n_i^* = n_i - \sigma_i = |T_i^*|$ vertices. Now $M_i^* = 2n_i^* - 1 = n_i^* + 2n_{i,1}^* - 1$ where $n_{i,1}^* = n_{i,1} - \sigma_i$. Therefore, T_i^* is a type 1 subtree.

Phase 1: Run the algorithm A on T_i^*, except that the source sends at step $t = 2k-1$ just a label of value k ($1 \leq k \leq n_i^*$) (not the message). Then the source sends successively to each node of S_i an unique label (in the range $[n_i^*+1, \ldots, n_i]$) by using σ_i times the following "3-step extension" ($3\sigma_i$ more steps). Order the vertices of $S_i = \{s_{n_i^*+1+h}, 0 \leq h \leq n_i - n_i^* - 1\}$ such that the following property is satisfied: for each h, $s_{n_i^*+1+h}$ is connected to $T^* \cup \{s_{n_i^*+1}, \ldots, s_{n_i^*+h}\}$. Hence there exists a path from s to $s_{n_i^*+1+h}$, where all the nodes except the last one ($s_{n_i^*+1+h}$) have already received a label. Let the vertices of this path be $u_0 = s, u_1 = r_i, u_2 = r_{i,1}, u_3, \ldots, u_{d_h} = s_{n_i^*+1+h}$, where $d_h = d(s, s_{n_i^*+1+h})$.

Do the following 3 steps in any order: in one step, do the compatible calls (u_{3p}, u_{3p+1}), in the next step, do the compatible calls (u_{3p+1}, u_{3p+2}) and in the last one, do the compatible calls (u_{3p+2}, u_{3p+3}).

During each call, each sender (if it is not the source) sends the label it has stored. Therefore at the end of the "3-step extension" each node has the label of its predecessor on the path. The source sends to r_i a new label $n_i^* + 1 + h$. Note that the calls in an extension are compatible with the calls of any other extension as they are done at different steps.

Note also that the order in which we organize the 3 steps has no importance. However for the purpose of clarity and using in theorem 4, we do the steps in an order such that the source is always sending at an odd step as soon as it becomes possible. So we do the calls (u_{3p}, u_{3p+1}) (including the call with the source as a sender) at step $2n_i^* + 3h + \epsilon$, where $\epsilon = 1$ if h is even and 0 if h is odd. Here h ranges from 0 to $\sigma_i - 1 = n_i - n_i^* - 1$. We do the calls (u_{3p+1}, u_{3p+2}) at step

Table 2. 9 steps of 3-step extension to label u, v and w

step		
16	$r_1 \to r_{1,1}$	$b \to u$
17	$s \to r_1$	$a \to b$
18	$r_{1,1} \to a$	
19	$s \to r_1$	$c \to v$
20	$r_1 \to r_{1,1}$	
21	$r_{1,1} \to c$	
22	$r_1 \to r_{1,1}$	$b \to u$
23	$s \to r_1$	$a \to b$
24	$r_{1,1} \to a$	$u \to w$

$2n_i^* + 3h + (1-\epsilon)$ and the calls (u_{3p+2}, u_{3p+3}) at step $2n_i^* + 3h + 2$. So the source sends at steps $2n_i^* + 1, 2n_i^* + 3, 2n_i^* + 7, \ldots, 2n_i^* + 6q + 1, 2n_i^* + 6q + 3, \ldots$ and is inactive at steps $2n_i^* + 6q + 5$.

At the end of the phase 1 of the algorithm, each node has received exactly one unique integer label ranging from 1 to n_i. Let x_k be the node which has received the value k.

Phase 2: Run the same algorithm again, but in the first part the source sends at step $t = 2k - 1$, $1 \le k \le n_i^*$, the message m_k destinated to x_k, and in the extensions at step $2n_i^* + 3h + \epsilon$, where $\epsilon = 1$ if h is even and 0 if h is odd, the message $m_{n_i^*+1+h}$to $x_{n_i^*+1+h}$, where $0 \le h \le n_i - n_i^* - 1$. (Another way to describe this is that in the steps when the source s sends a message, it is $m(v)$ where v contains the smallest label and $m(v)$ has not been sent.).

Example: Consider the type 2 tree given in Fig. 2 obtained by adding three vertices u, v and w and edges (b, u), (u, w) and (c, v) to the tree in Fig.1. Here, $n_1 = 11, n_{1,1} = 7$ and $n_1^* = 8$. Hence, $M_1 = 24 = n_1 + 2n_{1,1} - 1 (> 21 = 2n_1 - 1)$. Remember that by deleting the vertices u, v and w, the resulting tree is type 1. Now we illustrate algorithm B by applying it to this tree.

In phase 1, first we apply Algorithm A to the subtree obtained by deleting vertices u, v and w from the given tree (the resulting tree is exactly that of Fig.1), and send a label to each vertex in this subtree, and this takes 15 steps. The resulting labels which are those obtained in the previous example are given in the first row of Table 3. Then 3-step extension is used to extend the labels to the vertices u, v and w. Note that in this process, the labels given in the first part of 15 rounds will be changed. The 3-step extension is illustrated in Table 2. For example, steps 16, 17 and 18 are used to extend the labeling to the vertex u by moving the labels from s to u along the path $(s, r_1, r_{1,1}, a, b, u)$. We need 9

Table 3. Labels of vertices after the phase 1 of Algorithm B

	r_1	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	a	b	c	d	u	v	w
labels after 15 steps	8	7	4	6	3	1	5	2	-	-	-
labels after 18 steps	9	8	4	6	7	3	5	2	1	-	-
labels after 21 steps	10	9	4	6	7	3	8	2	1	5	-
labels after 24 steps	11	10	4	6	9	7	8	2	3	5	1
names of the vertices	x_{11}	x_{10}	x_4	x_6	x_9	x_7	x_8	x_2	x_3	x_5	x_1

Table 4. Last 9 steps of phase 2 of Algorithm B

step	m_1	m_3	m_5	m_7	m_8	m_9	m_{10}	m_{11}
16	$b \to u$				$r_1 \to r_{1,1}$			
17		$a \to b$				$s \to r_1$		
18				$r_{1,1} \to a$				
19			$c \to v$				$s \to r_1$	
20						$r_1 \to r_{1,1}$		
21					$r_{1,1} \to c$			
22		$b \to u$					$r_1 \to r_{1,1}$	
23				$a \to b$				$s \to r_1$
24	$u \to w$					$r_{1,1} \to a$		

steps to complete the labeling of u, v and w, Table 3 gives the labels of vertices at the end of each 3-step extension in the phase 1 of Algorithm B. The source is not sending at step 21.

Once we have the labels for the vertices, we are able to determine which messages the source should send at different steps. Now we are ready for the second phase of the algorithm. In phase 2, we run again the same algorithm, except this time, instead of labels, at step $t = 2k - 1$, for $1 \leq k \leq 8$, the source sends the message $m(v)$, where the label of the vertex v from the first phase of the algorithm is k. For example, s sends $m_1 = m(w)$ at the first step as $x_1 = w$ or the label of w is 1, and sends $m_2 = m(d)$ at the third step, as $x_2 = d$ or the label of d is 2 and so on. Then s sends at step 17 $m(a)$ as $x_9 = a$, at step 19, $m(r_{1,1})$ as $x_{10} = r_{1,1}$, and at step 23, $m(r_1)$ as $x_{11} = r_1$. Note that the protocol is exactly the same as that of the previous example for the first 15 steps and so they are omitted in the table 4. In fact, the vertices not in $T_{1,1}$ have received their messages at the end of the first 15 steps (they are the messages $m_2 = m(d)$, $m_4 = m(r_{1,2})$ and $m_6 = m(r_{1,3})$ that have arrived at their destinations). We indicate in the table 4 the steps of transmission of the other messages.

Proposition 4. *Algorithm B is valid and uses M_i steps (so $g(T_i) = M_i$).*

Proof. The algorithm B is valid as during each step we have only compatible calls (that is the case for algorithm A applied to T_i^* and then the calls of each step of the extension have been designed to be compatible). At the end of the algorithm each vertex has received its message. In fact, a vertex will receive its message in the first part of the algorithm (before the 3-steps extension) if it is in $T_{i,j}$, where $j \neq 1$, and otherwise, in one of the 3-steps of the last extension. The algorithm uses $2n_i^* - 1$ steps in the first part and then $3\sigma_i$ steps for the extensions. Therefore we have altogether $2n_i^* + 3\sigma_i - 1 = (n_i^* + \sigma_i) + (n_i^* + 2\sigma_i) - 1$ steps. But $n_i^* + \sigma_i = n_i$. By definition of T_i^*, $n_i^* = 2n_{i,1}^* = 2(n_{i,1} - \sigma_i)$ so $n_i^* + 2\sigma_i = 2n_{i,1}$ and so the number of steps is $n_i + 2n_{i,1} - 1 = M_i$.

3 General Algorithms

We will apply basic algorithms (A or B according to the type of subtrees) first in the case of a single subtree and then of two subtrees. For $m \geq 3$, we will use some

other techniques and induction; however we will deal first with some special cases. Recall that subtrees are ordered according to the values of M_i: $M_1 \geq M_2 \geq M_3 \geq \ldots \geq M_m$. In case of equality the order is determined by the sizes.

3.1 Case of One Subtree

In that case we apply directly the basic algorithm to the tree and we get.

Theorem 2. *In the case where T consists of one subtree T_1, $g(T) = M_1 > N-1$.*

3.2 Case of Two Subtrees

We apply the basic algorithm to the subtree T_1. All the vertices are informed in M_1 steps. We also apply simultaneously the basic algorithm to the subtree T_2, but starting at step 2 ; all the steps are translated by one and therefore all vertices of T_2 are informed in $M_2 + 1$ steps.

Theorem 3. *In the case where T consists of two subtrees T_1 and T_2, $g(T) = max\{M_1, M_2 + 1\}$ (this value is equal to $max\{N-1, M_1 + \epsilon\}$ where $\epsilon = 1$ if $M_1 = M_2$ and 0 otherwise).*

Proof. Let us first prove that all the calls are compatible. The validity of Algorithm A or B covers the case when two calls belong to the same subtree. That is the case also for the calls having the source as sender; indeed both in algorithm A or B the source is sending only during some odd steps. So here the source sends to r_1 at some odd steps and to r_2 at some even steps. Finally if two calls belong to different subtrees and are not both sent by the source, then the distance between one sender and the other receiver is at least 2.

Altogether the algorithm uses $max\{M_1, M_2 + 1\} = M_1 + \epsilon$ steps. We claim that $M_1 + \epsilon \geq N - 1$, which will prove that the lower bound is attained in that case. The claim is true if $M_1 \geq N-1$. If $M_1 \leq N-2$, then $2n_1 - 1 \leq M_1 \leq N-2$ and $2n_2 - 1 \leq M_2 \leq N-2$. (1) That implies $n_1 + n_2 \leq N-1$. But $N-1 = n_1 + n_2$ and therefore there are equalities everywhere in (1); that is $n_1 = n_2 = \frac{N-1}{2}$ and $M_1 = M_2 = N - 2$ and therefore $M_1 + \epsilon = N - 1$

3.3 General Case: m > 2

Due to lack of space the proofs are omitted in this section. Complete proofs can be acessible via the webpage of the first author[1]. We first deal with a special case

Theorem 4. *Suppose T consists of at least 3 subtrees such that T_1 and T_2 are of different types and $M_1 \geq N - 1$ and $M_2 = M_1 - 1$. Then $g(T) = M_1$.*

Then, for the case $m > 2$, when we are not in the special case of the preceding theorem 4, we apply induction on N and present algorithms which complete the personalized broadcasting in the number of steps that meet the lower bound.

[1] http://www-sop.inria.fr/mascotte/personnel/Jean-Claude.Bermond/

Therefore, the exact number of $g(T)$ is determined. We will suppose that the source sends at steps 1 and 2 to two different subtrees. Furthermore, if $M_1 \geq N-1$ and T_1 is of type 1, the algorithm used to send messages to T_1 is the basic algorithm A (in particular the source will send to r_1 in all odd steps). We assume that $N > 4$ otherwise it is trivial and we will distinguish 3 cases getting the following theorems.

Theorem 5. *Suppose T consists of at least 3 subtrees and $N-1 > M_1$. Then $g(T) = N-1$.*

Theorem 6. *Suppose T consists of at least 3 subtrees and $M_1 \geq N-1$ and T_1 is of type 2. Then $g(T) = M_1 + \epsilon$.*

Theorem 7. *Suppose T consists of at least 3 subtrees and $M_1 \geq N-1$ and T_1 is of type 1. Then $g(T) = M_1$.*

4 Conclusion

In this paper, we present efficient algorithms that give optimal solution for the gathering problem with buffering possibility, when the network is a tree with $d_I = 1$. It should be noted that in our algorithms, the size of our buffers never exceeds 1. However with such a small buffer, we can in some cases decrease considerably the gathering time comparing to the non buffering assumption considered in [3]. An extension would be to consider a non uniform distribution of messages. Our algorithm can be easily extended to the case where a node receives or sends $w(u) > 0$ messages ; indeed it suffices to replace a vertex with $w(u)$ messages by $w(u)$ vertices with one message. However if $w(u)$ is allowed to be 0, then the problem will become much more complicated.

It would also be interesting to investigate this problem for different value of d_I or some other structures of networks. In particular it is still an open question to decide if the problem is polynomial for trees in general.

Acknowledgments

We would like to thank all the persons who help us with fruitful discussions in particular L. Gargano, A. Liestman, J. Peters and S. Perennes.

References

1. Bermond, J.-C., Corrêa, R., Yu, M.: Gathering algorithms on paths under interference constraints. In: Calamoneri, T., Finocchi, I., Italiano, G.F. (eds.) CIAC 2006. LNCS, vol. 3998, pp. 115–126. Springer, Heidelberg (2006)
2. Bermond, J.-C., Galtier, J., Klasing, R., Morales, N., Pérennes, S.: Hardness and approximation of gathering in static radio networks. Parallel Processing Letters 16(2), 165–183 (2006)

3. Bermond, J.-C., Gargano, L., Rescigno, A.A.: Gathering with minimum delay in tree sensor networks. In: Shvartsman, M.M.A.A., Felber, P. (eds.) SIROCCO 2008. LNCS, vol. 5058. Springer, Heidelberg (2008)
4. Bermond, J.-C., Gargano, L., Rescigno, A.A., Vaccaro, U.: Fast gossiping by short messages. SIAM Journal on Computing 27(4), 917–941 (1998)
5. Bermond, J.-C., Peters, J.: Efficient Gathering in Radio Grids with Interference. In: AlgoTel 2005, Presqu'île de Giens, pp. 103–106 (May 2005)
6. Bertin, P., Bresse, J.-F., Le Sage, B.: Accès haut débit en zone rurale: une solution "ad hoc". France Telecom R&D 22, 16–18 (2005)
7. Bonifaci, V., Korteweg, P., Marchetti-Spaccamela, A., Stougie, L.: An approximation algorithm for the wireless gathering problem. In: Arge, L., Freivalds, R. (eds.) SWAT 2006. LNCS, vol. 4059, pp. 328–338. Springer, Heidelberg (2006)
8. Christersson, M., Gasieniec, L., Lingas, A.: Gossiping with bounded size messages in ad-hoc radio networks. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 377–389. Springer, Heidelberg (2002)
9. Chrobak, M., Gasieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. Journal of Algorithms 43(2), 177–189 (2002)
10. Elkin, M.L., Kortsarz, G.: Logarithmic Inapproximability of the Radio Broadcast Problem. Journal of Algorithms 52(1), 8–25 (2004)
11. Florens, C., Franceschetti, M., McEliece, R.: Lower bounds on data collection time in sensory networks. IEEE Journal on Selected Areas in Communications 22(6), 1110–1120 (2004)
12. Gaber, I., Mansour, Y.: Centralized broadcast in multihop radio networks. Journal of Algorithms 46(1), 1–20 (2003)
13. Gargano, L., Rescigno, A.: Optimally fast data gathering in sensor networks. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 399–411. Springer, Heidelberg (2006)
14. Gasieniec, L., Potapov, I.: Gossiping with Unit Messages in Known Radio Networks. In: Proceedings of the IFIP 17th World Computer Congress, pp. 193–205. Kluwer, B.V, Dordrecht (2002)
15. Hromkovic, J., Klasing, R., Pelc, A., Ruzicka, P., Unger, W.: Dissemination of Information in Commmunication Networks: Part I. Broadcasting, Gossiping, Leader Election, and Fault-Tolerance. Springer Monograph. Springer, Heidelberg (2004)
16. Klasing, R., Morales, N., Pérennes, S.: Complexity of bandwidth allocation in radio networks: the static case (to appear in Theoretical Computer Science)

Distributed Qualitative Localization for Wireless Sensor Networks

Karel Heurtefeux and Fabrice Valois

ARES INRIA /CITI, INSA-Lyon, F-69621, France
{karel.heurtefeux,fabrice.valois}@insa-lyon.fr

Abstract. The use of localization mechanism is essential in wireless sensor networks either for communication protocols (geographic routing protocol) or for application (vehicle tracking). The goal of localization mechanism is to determine either precisely or coarsely the node location using either a global reference (GPS) or a locale one. In this work, we introduce a new localized algorithm which classified the proximity of the neighborhood for a node. This qualitative localization does not use any anchor or dedicated hardware like a GPS. Each node builds a Qualitative Distance Table according to the 2-hop neighborhood informations. Thus, the algorithm allows to determine coarsely the location of the neighbors which are classified as *very close*, *close* or *far*. The algorithm is analyzed on a regular particular topology and then we evaluate this accuracy on a random topologies. We apply this algorithm for a localized topology control and we show that these topology control algorithms remain effective even without GPS information.

Keywords: Localization, location, gps-free, wireless sensor networks.

1 Introduction

Many applications for wireless sensor networks, as vehicle tracking or environment monitoring, need location awareness to work successfully. Geographic or location-based routing protocols can be used without mechanism of route request packets flooded in the whole network and so, the energy is saved and the performances are improved. Moreover, in topology control protocols, where each sensor node needs to adjust its power transmission to minimize the energy consumption the algorithms must be location-aware.

GPS [HWLC01] solves the localization issue in outdoor environments. However, for large sensor networks where nodes must be very small, low power and cheap, putting a GPS chip in every device is too costly.

In this paper, we propose a localized algorithm that allows to each node of the network to localize their neighbors using only local informations. Our objective is to show that in a wireless sensor networks where special hardware or GPS cannot be used for cost reasons, there is a way to obtain coarse positions of

* This work is partially funded by the french ANR RNRT ARESA project and the CARMA INRIA project.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 218–229, 2008.
© Springer-Verlag Berlin Heidelberg 2008

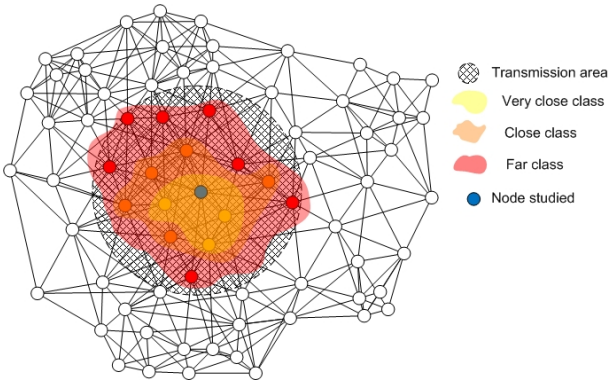


Fig. 1. Qualitative node proximity classification

the nodes. The algorithm uses only local informations obtained by exchanging neighborhood tables with classical hello packets to compute a **proximity index** for each 1-hop neighbor. We show that, despite the measurement errors, the algorithm is enough reliable and almost perfect on particular topologies (grid). The figure 1 illustrates the result of the algorithm: the neighbors of the studied node are classified in *very close* nodes, *close* one and *far* one.

The paper is organized as follows. In Section 2 some prior works about localization techniques are reviewed. The qualitative localization algorithm is presented in section 3. Next, the assumptions we made and the results we obtained are discussed in section 4. We conclude this work with some future work directions in section 6.

2 Related Works

Many localization techniques are proposed to allow nodes to estimate their location. We can distinguish two types of strategies of localization: fine and coarse localizations. The fine localization strategies determine precisely the coordinates of a node in the whole network whereas the coarse localization strategies specify a non precise area or introduce virtual coordinates, etc...

2.1 Fine Localization Strategies

The use of GPS system allows to localize a node precisely. However, it is expensive to install GPS receiver on each sensors. Some papers circumvent the problem and propose to use several anchors which are precisely located: each node can find its own position using triangulation or multi-lateration. For that, several solutions are proposed:

- The measuring from signal strength which is unrealistic because the radio signals can be disturbed by the environment,

- ToA (Time of Arrival) [CHH01] allows to compute the distance between two nodes by observing the time of propagation but this mechanism needs a nodes synchronization.
- TDoA (Time difference of Arrival) [WAH97], [NJ07]: two signals of different natures are used (ultrasound and radio for example) to improve the results of ToA.
- AoA (Angle of Arrival) [NN03], [AKBD06]: allows to determine the direction of a radio wave propagation.
- A combination of the TDoA and AoA [ML07] is also proposed to improve the accuracy and to adapt [CHH01] to 3D environments.

All those protocols don't take into account the energy consumption and assume that each node is able to compute the time or angle of arrival easily. Anyway, the anchor systems do not avoid the localization problem but reduce it to a subset of nodes of the network. Moreover other problems appear like the anchors placement in the network to allow a better localization of the nodes [BOCB07], [DT07].

2.2 Coarse Localization Strategies

Another strategy consists of finding approximate coordinates. If a non precise location of the sensor nodes is acceptable -depending on the application- several approaches are possible:

- The Active Badge system [HHB93]: each node is tagged and transmits a periodic hello packet every 10 seconds with a unique infra red signal which is received by dedicated sensors placed at fixed positions within a building, and relayed to the location manager.
- Location Estimation Algorithm [HE04] provides a probabilistic distribution of the possible node locations. According to both the prior location information and new observations from anchor nodes, impossible locations are filtered.
- The virtual coordinates [CA06]: each node determines its distance in number of hop to anchors and thus builds a virtual coordinates system. [WABDB07] shows that a routing protocol can be based only on virtual coordinates.

These protocols are not adapted to the sensor networks because either they require anchors connected to a fixed architecture or they require a centralized computation.

3 Algorithm Overview

Remember that the goal of our algorithm is to determine coarsely the location of the neighbors of a given node using only local informations. These local informations come from the hello packets which are exchanged between 1-hop neighbors. The qualitative location of a neighbor can be *very close*, *close* or *far*.

Such coarsely location can be used to construct a reliable unicast routing protocol in degraded wireless environment with a high level of interferences: to choose the *very close* nodes allows to choose the nodes with a high C/I ratio as relays. Applications in topology control or virtual coordinates for routing protocol are also possible.

A node A calculate proximity index with his neighbor B in the following way:

$$PI_A(B) = (|V(A)| \cap |V(B)|) - \frac{max(|V(A)|, |V(B)|)}{2}$$

where $V(A)$ is the neighborhood of A and $|V(A)|$ is the cardinality of $V(A)$.

The main idea is to give a high proximity index (PI) to the neighbor nodes having many common neighbors with the origin node (A) and few distinct neighbors. Indeed, we take into account the ratio between the number of common neighbors and the number of distinct neighbors. Effectively, close neighbors has a strong similar vicinity whereas distant neighbors will have much distinct neighbors. Thus, the proximity index is useful to represent the nodes which are *qualitatively* close. This logical proximity index is related to the geographical proximity in the case of dense and uniform networks. This mechanism allows to establish three distinguish classes among the neighbors: the *very close* class (or 1), the *close* class (or 2) and the *far* class (or 3) (see figure 1). We calculate the class node in the following way:

Let $PI(x)$ the proximity index of neighbor x:

$$inter = \frac{|max(PI(x_i)) - min(PI(x_i))|}{3}$$

$$class_x = \begin{cases} 1\ if PI(x) \geq max(PI(x_i)) - inter \\ 2\ if max(PI(x_i)) - inter > PI(x) \\ \quad \geq max(PI(x_i)) - 2.inter \\ 3\ if PI(x) < max(PI(x_i)) - 2.inter \end{cases}$$

Each node of the network computes a proximity index for each of its neighbors according to the local information received from its 1-hop neighbors. Each node maintains a table of his 1-hop and 2-hop neighborhood but diffuses only the table of its direct neighbors with periodic hello packets. Figure 2 and table 3 show an algorithm application on a particular node for a given topology. Node 27 classifies its neighbors in 3 proximity classes. We can see in details values found by the qualitative localization algorithm in Table 3. Table 3 proposes also a comparison between the qualitative classification of neighbors of the node 27 according to the algorithm and the real classification based on the Euclidean distance. Note that, on this example, the network is parse.

The protocol is inexpensive in energy because it only uses informations necessary to many other protocols: self-organization (CDS-rule-k [WL99], CDS-MIS [WAF02],...) and pro-active routing protocols (OLSR [CJ03]) deployed in wireless sensor networks. Moreover, if the network is not very dynamic (low mobility, not many birth or death of nodes in the network [HV07]) this exchange of packets can be reduced and limited to the deployment phase of the network.

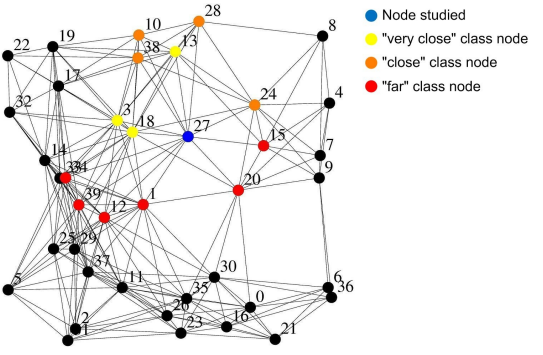


Fig. 2. Example of qualitative localization computed by the node 27

Neighbors nodes	proximity index	euclidean distance	proximity class	real class
18	2.0	50.067	very close	very close
3	1.0	65, 18	very close	very close
13	0.5	77, 01	very close	close
38	−0.5	83, 66	close	far
28	−0.5	103, 76	close	far
24	−0.5	66, 20	close	very close
10	−1.5	101, 18	close	far
1	−2.0	73, 09	far	close
20	−2.5	65, 96	far	very close
39	−3.0	115, 62	far	far
34	−3.0	115, 98	far	far
15	−3.5	68, 28	far	very close
12	−4.0	104, 40	far	far

Fig. 3. Comparison of the qualitative localization applied on the node 27. The classification obtained (*very close*, *close*, *far*) is compared to the classification obtained using a GPS with the Euclidean distance.

4 Simulation Results

All the results we provided here are computed using the simulator Java in Simulation Time (JiST) and Scalable Wireless Ad hoc Network Simulator (SWANS) [BHvR05]. The WSN topology is modeled as a Unit Disk Graph (UDG) and a CSMA/CA-like MAC layer is also used. Each node is motionless. The network cardinality varies between 50 and 700 nodes which are randomly and uniformly distributed in the simulation area except when we study the grid topology. The transmission power is used to control the average degree of network nodes. The objective is to investigate our protocol and observe its reliability to well classify the nodes.

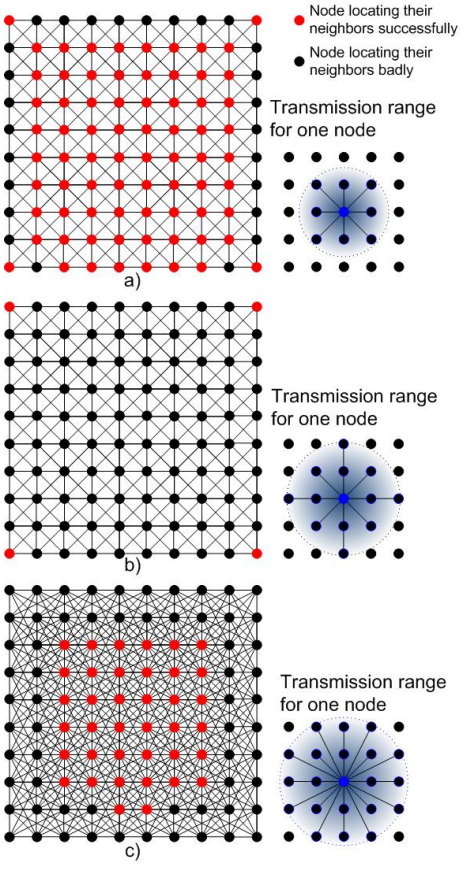


Fig. 4. Algorithm deployed on a grid

4.1 Qualitative Localization Protocol Behavior on a Regular Topology

We simulated a network of 100 sensors distributed uniformly to form a grid of 10x10 (see figure 4). Then, we increased the transmission power of each sensor and observe how our qualitative localization protocol reacts. Sometimes the vicinity of a node is not representative of the regularity of the whole network. In this case (Fig. 4, scenario b) or when the nodes are in the border area, the algorithm does not achieve to distinguish correctly the first two neighbors classes because of some incoherencies in the neighborhood. For other topologies, the neighbors classes can be determined without errors and the proximity index leads to the same classification that the euclidean distance. We can conclude that, when the topology and the neighborhood is almost uniform and regular, the qualitative localization is very effective and relevant.

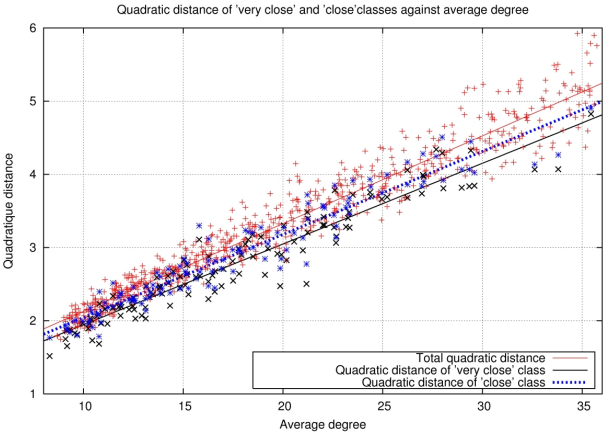


Fig. 5. Quadratic Distance in function of the average degree for each classes

4.2 Qualitative Localization Protocol Behavior on Random Topologies

But the sensor networks are seldom deployed with a regular topology. In order to measure the algorithm accuracy in more realistic environment we deployed, with a uniform and random distribution, 100 nodes and we varied the transmission power to increase the average degree. Then we calculated the quadratic distance between the neighbor nodes list classified using a GPS location and the same list classified using our algorithm.

Let two lists v and w in R^n be as follow: $v = (v_1, v_2, ..., v_n)$, $w = (w_1, w_2, ..., w_n)$. The quadratic distance dq is:

$$dq = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - w_i)^2}$$

In this study we investigate the quadratic distance of the algorithm for the classes *close* and *very close* and all the classes (Fig 5). We observe that the quadratic distance increases but in a much slower way than the average degree. When the average degree increases, the number of neighbors to be located for each node increases. If the quadratic distance remains low that means that the precision increases. This phenomenon is explained by a higher number of informations and thus a high reliability. The various classes evolve in the same way. Nevertheless, we can observe a lower increase for the classes *very close* and *close*.

In the case of dense topology (700 nodes, average degree: 40), the localization is very effective. We can see the localization into three classes on the figure 6. The yellow nodes are in the *very close* class, the orange ones in the *close* class and the red ones in *far* the class.

Each node allocates a class to its neighbors according to its proximity index. How evolve those classes when average degree increases? Will the *very close*

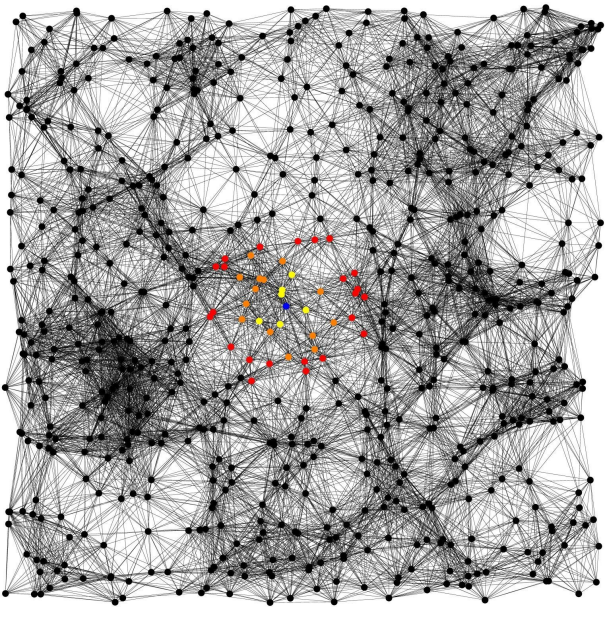


Fig. 6. Application of the algorithm in a random topology

class increases proportionally with the number of neighbors? We saw that the quadratic distance increased slightly when the average degree increased. However this metric is very sensitive to the length of the lists evaluated. Thus we investigate the average percentage of nodes selected in the *very close* class and in the *close* class (Fig. 7). We can note that, when the average degree increases, the percentage of nodes of the *very close* class decreases, whereas that of the *close* class increases. The *far* class remainder constant. This indicates that more

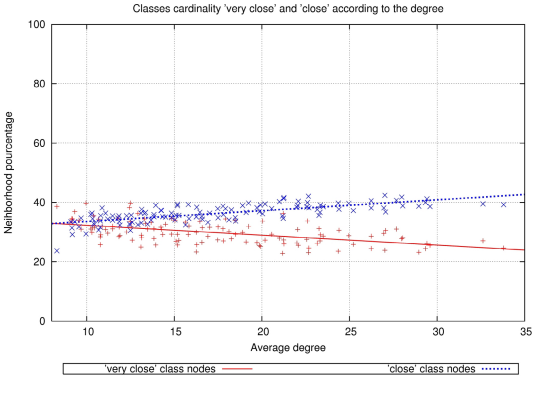


Fig. 7. Classes cardinality in function of the average degree

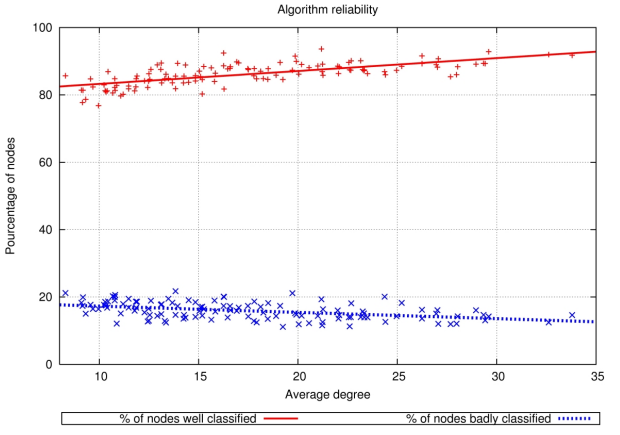


Fig. 8. Algorithm reliability

important is the density and more the index proximity able to distinguish the really *very close* nodes.

If we use this algorithm to know at which distance is a neighbor node, we should know if a neighbor selected as *close* or *very close* is indeed *close* or *very close* in the real world. To answer this question, we determined the number of neighbors belonging to the *close* and *very close* classes selected by the algorithm being indeed in the *close* and *very close* classes in a GPS-aware classification (red curve in Figure 8). Then we observe the number of nodes selected by the algorithm in these two classes and we note those which are not belonging to the GPS-aware classification *close* and *very close* (blue curve in Figure 8). More than 80% of nodes are well classified even for topologies with a low average degree.

5 Algorithm Application on Topology Control

In dense sensor networks it is often desirable to limit the vicinity to the closest neighbors. Several topology control algorithms exist like:

- Gabriel Graph [GS69]: an edge between u and v is selected if $disk(u, v)$ contains no another node inside.
- LMST [LHS03a]: Each node knows the location of its 1-hop neighbor and each node computes a MST in its neighborhood. The construction of the LMST topology is based on the construction of local MST by each node. An edge (u, v) is in the final LMST iif v is in the LMST(u) and u is in the LMST(v).
- RNG [Tou80]: Thanks to the position of the 1-hop neighbors, a node removes the longest links in the following way: given two neighbor nodes u and v, if there is a node w such as $d(u, v) > d(u, w)$ and $d(v, u) > d(v, w)$ then the edge (u, v) is deselected.

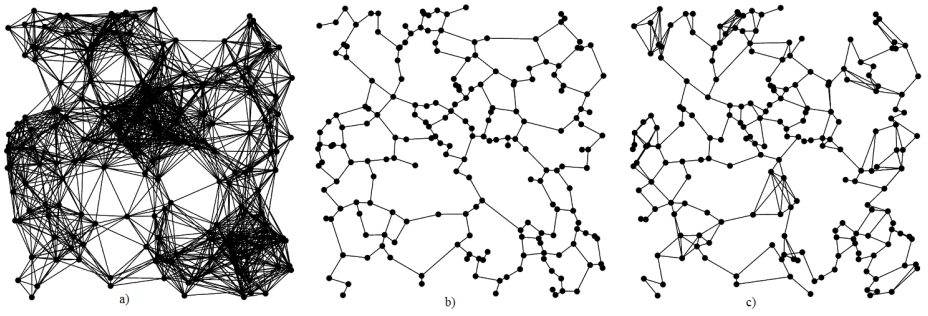


Fig. 9. a) Physical topology, b) Topology control (RNG, GPS) c) Topology control (RNG, Qualitative location)

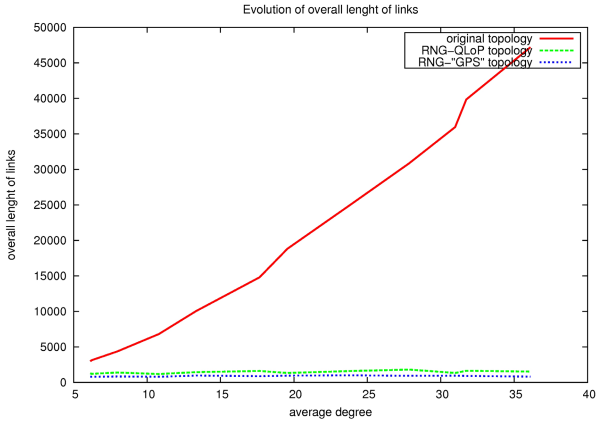


Fig. 10. Evolution of length of the topology links used

But those algorithms are generally based on the knowledge of the exact position of sensors (GPS, antenna array, RSSI, etc...). We applied our qualitative location algorithm to build a Relative Neighborhood Graph (see Figure 9, denoted as RNG-QLoP). Thanks to the proximity index of the 1 and 2 hop neighbors, a node removes the longest links in the following way: given two neighbor nodes u and v, if there is a node w such as $PI_u(w) > PI_u(v)$ and $PI_v(w) > PI_v(u)$ then the edge (u, v) is deselected. In Figure 10, we observe the effectiveness of the logical structure created by observing the overall length of the selected links: more the overall length is low, more the algorithm is relevant because of the energy saved. This analysis highlights two points: the performance of RNG-QLoP algorithm is very close to the RNG using GPS and more the density is important and more the performance of RNG-QLoP is important too. It is due to the information quantity increasing when the number of neighbors increases: it leads to a better precision.

6 Conclusions and Future Works

In this work we propose a qualitative localization algorithm using only local information. Our proposition does not use GPS information or any anchor or dedicated hardware. Based on the local informations from its neighborhood, a node can classify its neighbors as *very close* or *close* or *far* nodes. We have illustrated the behavior of our algorithm on a regular topology and on random one. A quadratic distance is computed to highlight the relevant classification provided. We apply this qualitative location algorithm for topology control (QLoP). A Relative Neighborhood Graph is computed using QLoP: the performances are very close the performances obtained when an absolute location (GPS) is used. Next, we will apply this qualitative localization algorithm to provide unicast routing protocol suited to wireless networks with interferences. Our idea is to favor paths made up of small hops and thus, to use *very close* nodes as relays because of their important signal-to-noise ratio.

References

[AKBD06] Akcan, H., Kriakov, V., Brönnimann, H., Delis, A.: Gps-free node localization in mobile wireless sensor networks. In: ACM MobiDE, pp. 35–42. ACM Press, New York (2006)

[BE02] Braginsky, D., Estrin, D.: Rumor routing algorithm for sensor networks. In: International Conference on Distributed Computing Systems, Vienna, Austria (2002)

[BHE00] Bulusu, N., Heidemann, J., Estrin, D.: Gps-less low-cost outdoor localization for very small devices. Personal Communications, IEEE 7(5), 28–34 (2000)

[BHvR05] Barr, R., Haas, Z., van Renesse, R.: Scalable Wireless Ad hoc Network Simulation, vol. 19, pp. 297–311. CRC Press, Boca Raton (2005)

[BOCB07] Benbadis, F., Obraczka, K., Cortés, J., Brandwajn, A.: Exploring landmark placement strategies for self-localization in wireless sensor networks. In: IEEE PIMRC, Athens, Greece (September 2007)

[CA06] Cao, Q., Abdelzaher, T.: Scalable logical coordinates framework for routing in wireless sensor networks. ACM Transactions on Sensor Networks 2(4), 557–593 (2006)

[CHH01] Capkun, S., Hamdi, M., Hubaux, J.-P.: Gps-free positioning in mobile ad-hoc networks. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences, vol. 9, p. 9008 (2001)

[CJ03] Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC 3626 (October, 2003)

[DT07] Desai, J., Tureli, U.: Evaluating performance of various localization algorithms in wireless and sensor networks. In: IEEE PIMRC, Athens, Greece (September 2007)

[GS69] Gabriel, G.R., Sokal, R.R.: A new statistical approach to geographic variation analysis. Systematic Zoology 18, 259–278 (1969)

[HE04] Hu, L., Evans, D.: Localization for mobile sensor networks. In: ACM MobiCom, pp. 45–57. ACM Press, New York (2004)

[HHB93] Hopper, A., Harter, A., Blackie, T.: The active badge system (abstract). In: CHI 1993: Proceedings of the INTERACT 1993 and CHI 1993 conference on Human factors in computing systems, pp. 533–534. ACM Press, New York (1993)

[HV07] Heurtefeux, K., Valois, F.: Topology control algorithms: a qualitative study during the sensor networks life. In: 3rd International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN 2007), in conjunction with MASS, Pisa, Italy (October 2007)

[HWLC01] Hofmann-Wellenhof, B., Lichteneeger, H., Collins, J.: Global Positioning System: Theory and Practice. Springer, Wien (2001)

[LHS03a] Li, N., Hou, J.C., Sha, L.: Design and analysis of an MST-based topology control algorithm. In: IEEE INFOCOM, San Francisco, USA (April 2003)

[LHS03b] Li, N., Hou, J.C., Sha, L.: Design and analysis of an MST-based topology control algorithm. In: IEEE INFOCOM, San Francisco, USA (April 2003)

[ML07] Magnani, A., Leung, K.: Self-organized, scalable gps-free localization of wireless sensors. In: IEEE WCNC, pp. 3798–3803, Hong Kong, China (2007)

[NJ07] Nawaz, S., Jha, S.: Collaborative localization for wireless sensor networks. In: IEEE PIMRC, Athens, Greece (September 2007)

[NN03] Niculescu, D., Nath, B.: Ad hoc positioning system (aps) using aoa. In: IEEE INFOCOM, San Francisco, USA (2003)

[SBK06] Saad, C., Benslimane, A., König, J.-C.: Mur: A distributed preliminary method for location techniques in sensor networks. In: IEEE WiMob, Montréal, Canada, pp. 61–68 (2006)

[Tou80] Toussaint, G.: The relative neighbourhood graph of a finite planar set. Pattern Recognition 12, 261–268 (1980)

[WABDB07] Watteyne, T., Augé-Blum, I., Dohler, M., Barthel, D.: Geographic forwarding in wireless sensor networks with loose position-awareness. In: IEEE PIMRC, Athens, Greece (September 2007)

[WAF02] Wan, P., Alzoubi, K., Frieder, O.: Distributed construction of connected dominating set in wireless ad hoc networks. In: INFOCOM, New York, NY, USA (2002)

[WAH97] Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Communications 4, 42–47 (1997)

[WL99] Wu, J., Li, H.: On calculating connected dominating set for efficient routing in ad hoc wireless networks. In: ACM DIALM, pp. 7–14. ACM Press, New York (1999)

[Yao77] Yao, A.C.: On constructing minimum spanning trees in k-dimensional spaces and related problems. Technical report, Stanford, CA, USA (1977)

A Lower Bound on the Capacity of Wireless Ad Hoc Networks with Cooperating Nodes

Anthony S. Acampora and Louisa Pui Sum Ip

Department of Electrical and Computer Engineering,
University of California, San Diego,
9500 Gilman Drive,
La Jolla, California, 92093, USA
acampora@cts.com, louisa.ip@sri.com

Abstract. In this paper, we consider the effects on network capacity when the nodes of an ad hoc network are allowed to cooperate. These results are then compared to the theoretical upper bound on the capacity of an ad hoc network without cooperation. For our cooperative model, two or more nodes are grouped together to cooperatively transmit information from the source node to the destination node. Here the lower bound is presented without frequency reuse and it is found that node cooperation can only help improve network capacity as the number of nodes in cooperation increases. However, the results of our cooperative network model show that without frequency reuse, node cooperation could not out perform a peer-to-peer network with frequency reuse. Furthermore, the improvement of network capacity might not be worthwhile beyond two nodes cooperating together. The three-node cooperation yields minimum gain, if not negligible, compared to the two-node cooperation.

Keywords: Ad Hoc Networks, Capacity Bounds, Cooperation, Peer to Peer Networks, Routing Algorithm, Spatial Diversity, Wireless Networks.

1 Introduction

Previously we have studied the information theoretic bounds on the capacity of peer-to-peer wireless ad hoc networks with hop-by-hop routing. We found the theoretical upper bound of an N x N network from the capacity matrix representing the point-to-point Shannon capacity that exists between two nodes and a relative traffic matrix [1],[2]. Each element of the capacity matrix represents the maximum rate at which information may be transferred between pair of two nodes with no co-channel interference. For the relative traffic matrix, each element represents the exogenous traffic between the source node and the destination node. Applying source switching entropy, we are able to bound the upper maximum factor by which the relative traffic matrix may be scaled. We use the result as a yardstick against any approach for peer-to-peer wireless networking and other cooperative networking schemes.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 230–240, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The area of single node hop-by-hop and single hop routing algorithm for wireless ad hoc networks has been well studied. These methods of transmission often required high transmit power, thus caused an increase of interference. One method to increase efficiency of the network is to increase spatial diversity by using multiple antennas in a node. However, multiple antennas in a simple node are often impractical and undesirable. Unlike wired infrastructures where packets may be directed to the recieving node only, packets in a wireless network are broadcast in the wireless medium. Other nodes near the transmitting node may receive the packets at no extra cost to the transmitting node. Therefore, another way to provide the spatial diversity is to allow individual nodes who are within receiving range of the transmitting node be grouped together and transmit cooperatively. There have been abundance of study done on the cooperation of paired nodes transmitting to a single receiver [3],[4],[5],[6], paired nodes transmitting to two receiver [5],[7],[8], but little attention has been placed on any higher order number of nodes cooperating together [9],[10]. In this paper, we examined the benefits of pairing up more than two nodes for cooperation and compared the results with paired nodes cooperation and traditional hop-by-hop networks.

The purpose of this study is to determine if there are any benefits in allowing more than two nodes to cooperatively send together and how does a simple time-share bound of a cooperative wireless ad hoc network compare to the capacity bound of a traditional hop-by-hop network with frequency reuse. We steered our attention to finding the lower bound of a similar network but with node cooperation. We first focus on a simple model of node cooperation with time-share bound. If this lower bound is reasonably close to our previous upper bound, it might be worthwhile to explore further. One obvious idea is to incorporate frequency reuse in place of our simple time share bound. On the other hand, if this lower bound is performing much worse than our previous upper bound, perhaps another method is needed or it could be that our previous upper bound was too loose. Maybe node cooperation is hindering, rather than expediting the transfer of packets.

This paper is organized in five sections. In section 2, we present the overall network model, the capacity matrix and the relative traffic model. In Section 3, we present the algorithm employed and how the optimum path is determined.

In Section 4, we present the results based on two simple cases; 9 and 25 nodes networks. The results show that without frequency reuse, node cooperation could not out perform peer-to-peer networks with frequency reuse as presented in our previous paper [1]. Even though the upper bound of the traditional hop-by-hop network capacity is much higher than what we have for lower bound with cooperation, nevertheless, for a simple time division strategy we achieved an order of magnitude 14.28 times at SNR equal to 5 and 5.67 times at SNR equal to 15 better than our hop-by-hop network with no frequency reuse, for a 9-node network with path loss, multipath fading and shadow fading. Furthermore, the three-node cooperation yields very minimum gain, if not negligible, compared to the two-node cooperation. Finally, in Section 5, we conclude with some numerical results and present future direction.

2 The Capacity Matrix and Traffic Matrix of the Network Model

The model for the N-node wireless network is shown in Figure 1. Each wireless node may send or receive information to or from another single node, paired nodes, three nodes, or up to $N-2$ nodes. The constraint placed on each node is that no node may send and receive simultaneously. For nodes with cooperation, we assume a phase shifter is built in with the model. Each source node may experience propagation impairments (e.g., shadow and multipath fading) that limit the rate at which information may be sent. Whenever a source node sends information to destination node via another single node, it behaves as a peer-to-peer network.

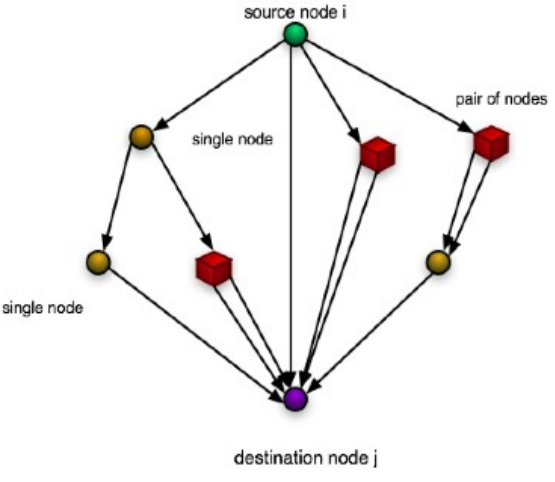


Fig. 1. Possible paths taken by source node for transmitting packets to destination node in an N-node wireless ad hoc network

The co-channel interference free capacity between a (or multiple) sending and receiving node(s) is defined as $C = W\log(1+\rho)$. W is the bandwidth available and it is set to one here for results in per Hz. ρ is the interference free signal-to-noise ratio (SNR). For each transmitting/receiving pair, the capacity matrix can be written as

$$\bar{\bar{C}} = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1N} \\ C_{21} & C_{22} & & \\ \vdots & & \ddots & \vdots \\ C_{N1} & C_{N2} & \dots & C_{NN} \end{bmatrix} \tag{1}$$

where C_{ij} means node i is the transmitting node and node j is the receiving node. Each node may not send a packet to itself, thus the diagonal of the capacity matrix is default to zero.

Our propagation model consists of path loss, flat multipath fading and shadow fading between each node. The path loss between each node is defined as

$$r = \left(\frac{d}{R}\right)^4 \tag{2}$$

where R is the relative distance between two nodes and d is the distance between two nodes as defined in the beginning of section 4. This defines the normalized SNR ratio if the transmitting and receiving nodes are separated by a distance d.

The flat multipath fading is defined as an exponential distribution with random variable X and mean one. Mathematically, multipath fading has little effect to the signal to noise ratio when paired with path loss. The shadow fading follows a log normal distribution with standard deviation of 6 dB.

The N nodes are randomly distributed over the service area and the SNR with propagation impairments is computed in the following fashion.

$$\rho_{ij} = \rho * \left(\frac{d}{R}\right)^4 * e^X * 10^{\frac{\epsilon}{10}} \tag{3}$$

where ρ is the SNR in linear scale and ϵ is Gaussian distributed, zero mean and standard deviation of six. ρ_{ij} is the SNR from node i to node j. The actual capacity with normalized bandwidth is then equates to

$$C = log(1 + \rho_{ij}) \; . \tag{4}$$

In the event where a node is sending to two or more nodes, the smaller of the capacity elements is selected. This capacity link is defined as

$$C_i^{jkl} = min\{C_{ij}, C_{ik}, C_{il}\} \tag{5}$$

for the case where node i is sending to nodes j, k and l.

To simplify the problem, we assume nodes are perfectly synchronized and allow superposition at the receiving node. Although it is important to consider the effects of an asynchronous network [8] and its performance tolerance to noncoherent scenarios [11], it is beyond the scope of this paper.

In the case where two or more nodes are cooperatively sending to a single node, the actual capacity would then be based on the sending nodes SNR. SNR is defined as the signal power over noise power, these two parameters are of units watt or volt^2. For example, if a link is sending with P watts, the volt would then be the square root of P. After adding the two signals in volts, we take the square of the resulting voltage to obtain watts. For example, if node j and node k are sending cooperatively to node i, we have

$$v_{jk}^i = v_{ji} + v_{ki} = \sqrt{\rho_{ji}} + \sqrt{\rho_{ki}} \; . \tag{6}$$

The SNR of this transmission is then defined as

$$(v_{jk}^i)^2 = (\sqrt{\rho_{ji}} + \sqrt{\rho_{ki}})^2 \; . \tag{7}$$

We may express the equation in watt,

$$\rho_{jk}^{i} = (\sqrt{\rho_{ji}} + \sqrt{\rho_{ki}})^2 \tag{8}$$

and the actual capacity is calculated as in equation (4). While we are assuming perfect time synchronization, it would be worthwhile to examine how slight imperfection of time synchronization would effect the signal quality at the receiver. It would be useful to examine the receiver's tolerance to imperfect time synchronization.

For two or more nodes to cooperate and send to a single node, we store these values in another capacity matrix. This pre-determined capacity table is similar to the idea of table-driven routing protocols used to maintain up-to-date routing information between the nodes within the network [12].

Using the same principal in which we defined the capacity matrix, we can define the Relative Traffic Matrix as

$$\bar{\bar{T}} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1N} \\ t_{21} & t_{22} & & \\ \vdots & & \ddots & \vdots \\ t_{N1} & t_{N2} & \dots & t_{NN} \end{bmatrix} \tag{9}$$

where t_{ij} is the exogenous traffic generated due to information delivered from node i to node j in fixed length packets per second as in our previous upper bound calculation [1]. As in the case with the capacity matrix, the diagonal elements of the traffic matrix are zeros. However, it is not necessary that the matrix is symmetric because the source and destination nodes could be of different equipment. We currently use a uniform traffic model to isolate the problem on how cooperation effect network capacity.

3 Routing Algorithm and Cost Calculation

Each element in the network may send information only when the network is free. The network has no frequency re-use and therefore presents the time-share bound.

The traffic may flow in any path available with the following constraint; a single node may transmit a packet to a pair of nodes or to another single node, but a pair of nodes may not transmit information to another pair of nodes. Figure 2 shows a diagram of how traffic elements flow sequentially.

When a node is free to send information from its queue, it may send to a conventional single node, two or more nodes. We used a simple iterative method in finding the best optimal pairing of nodes. We allowed the first node to be of any node from node 1 to node N with the exception that it may not be the source node or the destination node. We then pick the second node in the same manner except we start the search loop from one node after the starting point of the first node's search loop. This is because pairing of node 1 and node 2 is the same as pairing of node 2 and node 1. The decision on which path to take is

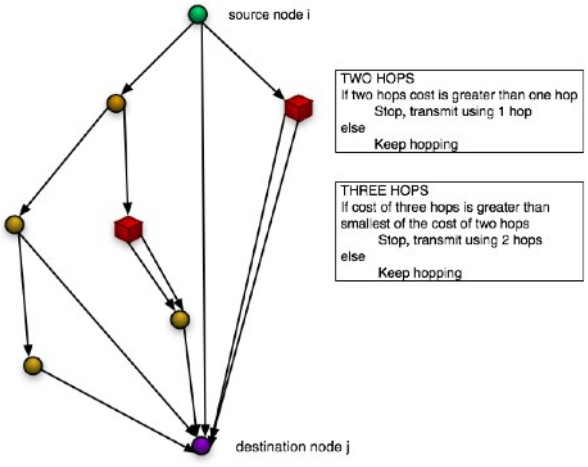


Fig. 2. Algorithm on how many hops is optimum for transmitting packets from source node to destination node

based on a simple algorithm of selecting the least time required path. The source node will send information through a node pair if and only if such transmission requires less time than a single node transmission path.

The number of hops is also determined based on the least time required for transmission. We defined the time required to send a packet from one node or nodes to another node as the cost for that hop. If a single hop transmission takes the least amount of time, the source node will directly send the packet to the destination node. However, if the source node detects that transmiting to the destination node via another single or paired node requires less net time than the one-hop method, the source node will take the latter path.

We define t_{ref} as the time it takes for a source node to send information directly to the destination node. If M bits were sent using capacity link C_{ij}, t_{ref} would be the M bits divided by the capacity link. This relationship can be expressed as follows:

$$t_{ref} \propto \frac{1}{C_{ij}} \, . \tag{10}$$

With two or more hops, we first determine if the transmission time from one node to another single node is less than, or more than from one node to a pair of nodes. We take the smaller of the two,

$$t_{route} = min \left[t_{single}, t_{pair} \right] \, . \tag{11}$$

t_{route} is calculated from the source node to intermediate nodes and arriving at the destination node. If a three-hop path requires less time than a two-hop path (perhaps due to higher capacity links between intermediate nodes), t_{route} is then the sum of the transmission time of the three hops in which the packet traveled

from the source to the destination node. While a single node may send to a pair of nodes, a pair of nodes is constrained to only sending to a single node.

The time it takes for a source node transmitting information to the destination node is the smaller of t_{ref} and t_{route}. The total time a network takes for each node to send its information is then the sum of each node's transmission time, or

$$t_{total} = \sum_{l=1}^{N} min\left[t_{ref}, t_{route}\right] \ . \tag{12}$$

The total capacity of the network is then

$$C_{network} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} t_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} t_{ij} t_{total}} \tag{13}$$

where t_{ij} is an element from the traffic matrix. This is the lower bound with no frequency reuse.

4 Results

As in our previous work [1], to generate numerical results, we consider a square service area, with each side of the square being length L and N nodes are placed in the square service area with distance d apart in all directions. There would be $\sqrt{N}$ nodes in each row and each column and the distance d between each node is

$$d = \frac{L}{\sqrt{N} - 1} \tag{14}$$

By doing so, if the transmitting node is separated from the source node by distance d, we may then define the SNR as the normalized signal to noise ratio in the absence of fading and interference. Further more, our model assumes a $(\frac{1}{r})^4$ path loss, flat multipath fading and log normal shadow fading with standard deviation of 6dB as with our previous work.

Figure 3 shows the performance for different quantities of nodes paired up as a group to cooperatively transmit packets they receive to another node over a 9-node network. The traffic matrix used here is uniformly distributed. We have included path loss, multipath fading and shadow fading in the simulation. A total of ten simulations were run, with each run corresponding to a different set of randomly distributed nodes over the service area.

It is clear that by including cooperation, the overall network performance improved significantly, by approximately a factor of 20 with SNR equal to 0 and the factor decreases as the SNR increases.

The plot on the right in Figure 3 is a close-up view of the performance curve for node cooperation with two, three, four, five and six nodes. Our results show that the capacity gain is not significant beyond grouping two nodes together. The three-node cooperation yields minimum gain, if not negligible, compared to the two-node cooperation. However, by increasing the number of nodes to be

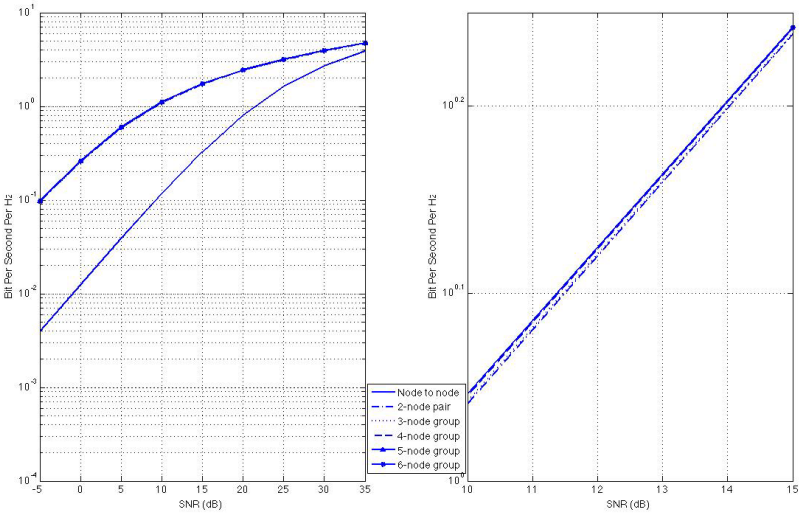


Fig. 3. Performances for a different number of nodes per group in cooperation based on the average of ten trials for a 9-node network, with uniform traiffic, path loss, multipath fading and shadow fading. The figure on the right is a close-up look at the curves with cooperation.

allowed to group together for cooperation, the network capacity could only be improved because we are picking the least transmission time path.

Whenever a node is added into a group to cooperatively transmit a packet, the complexity of transmission increases. For example, each transmitting node must receive the packet; synchronizes its clock before transmitting the packet out to the intended node. While performance does improve as we increase the number of nodes cooperating with each other, the increase of complexity might not be worthwhile for grouping more than two nodes together. Figure 4 shows the same result as Figure 3 but with a 25-node network. From our observations, the advantages of using node cooperation diminish as SNR increases, this is because the effects of the imperfection of the channel diminish with higher SNR.

Although the network setup with node cooperation has clearly out performed the network without cooperation, here we are calculating a network's time-share bound with no frequency reuse. From previous work, where frequency reuse is taken into consideration, our results for a peer-to-peer networks with frequency reuse still out perform the results presented here for the network using node cooperation and no frequency reuse.

We now compare our current results with no frequency reuse to our previous work of peer-to-peer networks with hop-by-hop routing and frequency reuse. We look at the case of a 25-node network with uniform traffic, with path loss, multipath fading and shadow fading as with figure 4. Although the new result for 25 nodes, peer-to-peer transmission resembles a linear line, but the curve tapers off slowly at higher SNR. Our information theoretic upper bound is shown in Figure 5 along with our current lower bounds with and without cooperation and

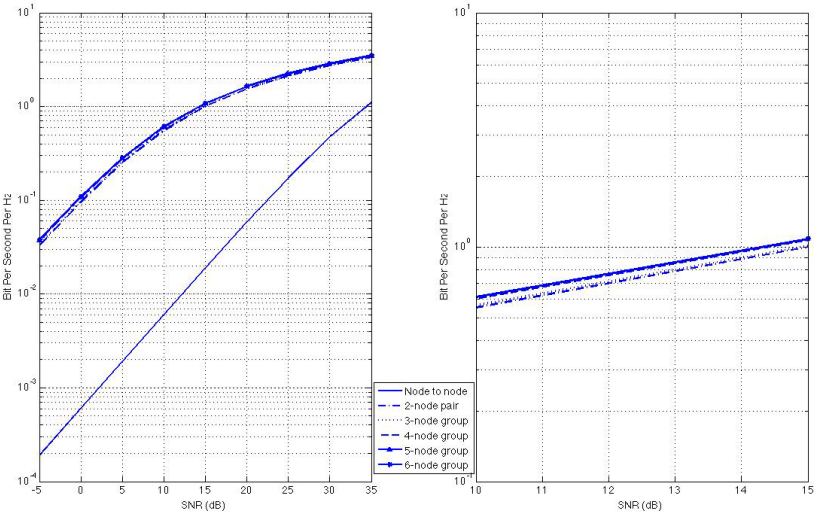


Fig. 4. Performances for a different number of nodes per group in cooperation based on the average of ten trials for a 25-node network, with uniform traffic matrix, path loss, multipath fading and shadow fading. The figure on the right is a close-up look at the curves with cooperation.

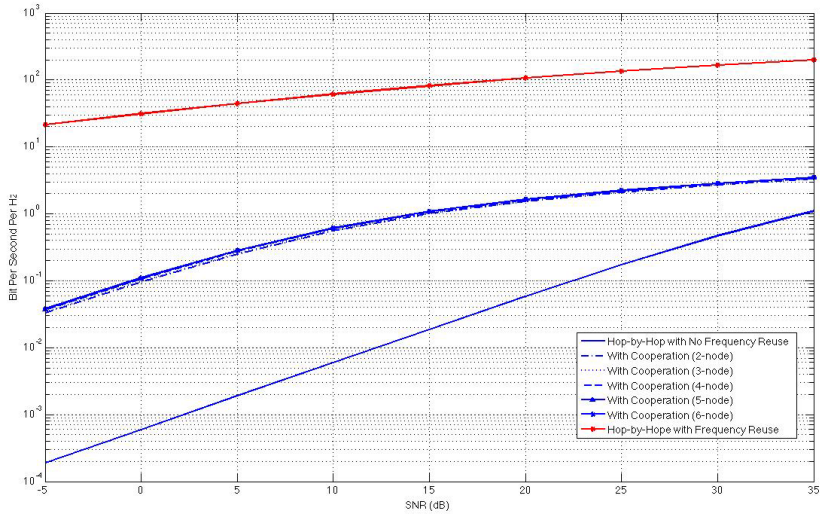


Fig. 5. Performance comparison of different number of nodes per group in cooperation to traditional hop-by-hop network(*top most solid line with "*"*), based on the average of ten trials for a 25-node network with uniform traffic matrix, path loss, multipath fading and shadow fading

no frequency reuse. We noticed that our lower bound with cooperation is far less than our upper bound without cooperation. This tells us either our upper bound is very lose, or there is substantial opportunity to devise a routing algorithm capable of much better performance than the simple strategy we have invoked in this paper. Our goal for the future is to investigate both of these. We expect cooperation would substantially out perform a network that uses simple hop-by-hop routing. Currently, the total information theoretic network capacity is roughly 300 times better than our current result with node cooperation at SNR equal to zero, roughly about 85 times better at SNR equal to 15 and 60 times better at SNR equal to 30. As SNR increases, cooperation between nodes would become less effective in improving network capacity. This is because the higher the SNR, the less effects the imperfections of the channel would cause. However, peer-to-peer transmission would never outperform two-node cooperation transmission based on the routing algorithm we employed.

5 Conclusions

By considering node-cooperation, we can significantly improve the performance of the network capacity. We have used a simple case with no frequency reuse to show the benefit of cooperation. The net capacity of the network has clearly improved over the traditional peer-to-peer network. However, even with node cooperation, a time-based bound could not out perform a simple peer-to-peer network if such network utilizes frequency reuse.

We have previously stated that with the upper bound found in [1], no media access protocol and no hop-by-hop routing algorithm can possibly produce an overall network capacity greater than this upper bound. Here, we show that with node cooperation, even the simplest case with two nodes cooperatively transmitting information, can out perform simple peer-to-peer node transmission without frequency reuse. What might be missing to allow cooperative network to perform better than our previous upper bond might be of the lack of frequency reuse or the lack of receiver cooperation. From [7], it showed that transmitter and reciever cooperation performs better than receiver cooperation or transmiter cooperation only. With this possibility, we next plan to explore the benefits offered by frequency reuse for our current network model with cooperation for two nodes pair. If our future model with frequency reuse still could not out perform our information theoretic upper bound, we might consider adding receiver cooperation to our model.

Acknowledgments. The authors sincerely thank the three reviewers for their constructive comments in the early stage of submission. The aurthors are also very thankful to Bryan Chavez, senior research engineer at SRI International and Dr. Michael Tan, for generously spending valuable time in reviewing this paper prior to our final submission.

References

1. Acampora, A.S., Ip, L.: Information theoretic bounds on the capacity of peer-to-peer wireless networks with hop-by-hop routing. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (September 2007)
2. Acampora, A.S., Tan, M., Ip, L.: Fundamental Bound on the Capacity of Ad Hoc Networks with Conventional Hop-by-Hope Routing. In: Wireless Communications CNIT Thyrrenian Symposium (September 2007)
3. Sendonaris, A., Erkip, E., Aazhang, B.: User Cooperation Diversity - Part I: System description. IEEE Transactions on Communications 51(11), 1927–1938 (2003)
4. Sendonaris, A., Erkip, E., Aazhang, B.: User Coperation Diversity - Part II: Implementation Aspects and Performance Analysis. IEEE Transactions on Communications 51(11), 1939–1948 (2003)
5. Ng, C.T.K., Jindal, N., Goldsmith, A.J., Mitra, U.: Capacity Gain From Two-Transmitter and Two-Receiver Cooperation. IEEE Transactions on Information Theory 53(10), 3822–3827 (2007)
6. Yazdi, K., El Gamal, H., Schniter, P.: On the Design of Cooperative Transmission Schemes. In: Proceedings of Allerton Conference on Communication, Control and Computing (October 2003)
7. Jindal, N., Mitra, U., Goldsmith, A.: Capacity of Ad-Hoc Networks with Node Cooperation. IEEE International Symposium on Information Theory (December 2003)
8. Stanković, V., Host-Madesen, A., Xiong, Z.: Cooperative Diversity for Wireless Ad Hoc Networks: Capacity bounds and code designs. IEEE Signal Processing Magazine 23(5), 37–49 (2006)
9. Elia, P., Kumar, P.: Constructions of Cooperative Diversity Schemes for Asynchronous Wireless Networks. In: IEEE International Symposium on Information Theorey, pp. 2724–2728 (July 2006)
10. Scaglione, A., Goeckel, D.L., Laneman, J.N.: Cooperative Communications in Mobile Ad Hoc Networks. IEEE Signal Processing Magazine 23(5), 18–29 (2006)
11. Chen, D., Laneman, J.N.: Noncoherent demodulation for cooperative diversity in wireless systems. In: IEEE Global Telecommunications Conference, vol. 1, pp. 31–35 (December 2004)
12. Royer, E.M., Toh, C.K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications (April 1999)

Attacks on CKK Family of RFID Authentication Protocols*

Zbigniew Gołębiewski[1], Krzysztof Majcher[2], and Filip Zagórski[2]

[1] Institute of Computer Science, Wrocław University
zbigniew.golebiewski@ii.uni.wroc.pl
[2] Institute of Mathematics and Computer Science, Wrocław University of Technology
krzysztof.majcher@pwr.wroc.pl,
filipz@im.pwr.wroc.pl

Abstract. At Pervasive 2008, Cichon, Klonowski, Kutylowski proposed a family of shared-key authentication protocols (CKK). Small computational and communication cost, together with possibility of efficient hardware implementation makes CKK attractive for low-cost devices such as RFID tags. In this paper we present a couple of attacks on CKK protocols, both passive and active.

Keywords: lightweight cryptography, RFID, authentication, HB, HB+.

1 Introduction

A lightweight cryptography becomes important nowadays. Such kind of cryptography is especially used in weak devices containing simple, low cost microchips. One of the examples of such a devices are RFID Tags, which use 8-bit processors, small memory (few hundred bits) and the possibility of low-bandwidth radio communication on short distances. Low-cost RFID systems are introduced as the successors of widely used barcodes. RFID Tags are commonly used as a small data storage of the objects which they are attached to, RFIDs allow their automatic identification.

RFID systems consist of the radio frequency tags and radio frequency reader. A tag usually does not have any battery and is induced by a signal sent by a reader. An activated tag can respond to the challenge sent by a reader and thus authenticate itself.

It is easy to see that because of the simplicity of the architecture used in RFIDs, designing secure and reliable authentication protocols is one of the main problems.

Because of very strong hardware (cost) limitations, one cannot use the battle-tested asymmetric cryptography protocols as RSA ([10]), ElGamal ([5]) and even symmetric cryptography protocols as AES ([3]) – i. e. the "smallest" implementation of AES contains over 10000 logic gates and because of the price it cannot be used in low-cost RFIDs.

Many recent papers describe variety of the lightweight authentications protocols dedicated for RFID systems. Many of them use about few hundred of logic gates. One of the security mechanisms was proposed by Vajda and Buttyán in ([11]) and then by Stephen

* This work was partially supported by EU within the 7th Framework Programme under contract 215270 (FRONTS).

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 241–250, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Weis et al. in ([12]). Those security mechanisms are not sufficient enough (i. e. protocols are too complicated or there were proposed attacks on those protocols (cf. [4])). The milestone in the research was done by Ari Juels and Stephen Weis in ([8]), where they have described HB and HB^+ authentication protocol. HB^+ is upgraded version of the human-to-computer authentication protocol designed by Hopper and Blum (HB) ([7]) and it is based on the "Learning Parity with Noise" (LPN) problem which was proved to be NP hard. The authors of HB^+ protocol claim that HB^+ is secure against passive and active adversaries. There are few papers in which authors analysed the security of the HB^+ ([1], [9], [6]) but all proposed there attacks are not practical in the point of view RFID systems. The HB^+ protocol has also few disadvantages. The main of them is that: if the authentication should be secure and reliable then a tag and a reader have to send many kilobytes of data. Therefore the authentication could last too long (even few seconds).

Another approach to the security mechanisms for RFID systems was proposed by Cichon, Klonowski and Kutyłowski in ([2]) (in the rest of this paper called CKK authentication protocol). They introduce few versions of a protocol that is based on pre-shared keys represented as the hidden subsets of random sequence. It is worth to mention that CKK tag needs few bits to be sent. For the case of the key length 128, a tag needs to send only 158 bits, while in the case of HB^+, for the same key length and noise parameter 0.05, number of transmitted bits is about 32000 (the higher noise parameter is the more bits need to be transmitted).

In this paper we perform security analysis of the versions of CKKprotocol, we describe possible passive and active attacks.

Organization of the paper. In the 2 section, we describe members of CKKfamily and introduce basic notation which is then use through the paper.

Section 3 presents passive attacks on CKK, CKK^2, CKK_p protocols. While in the Section 4, we present active attacks.

2 Protocols Description

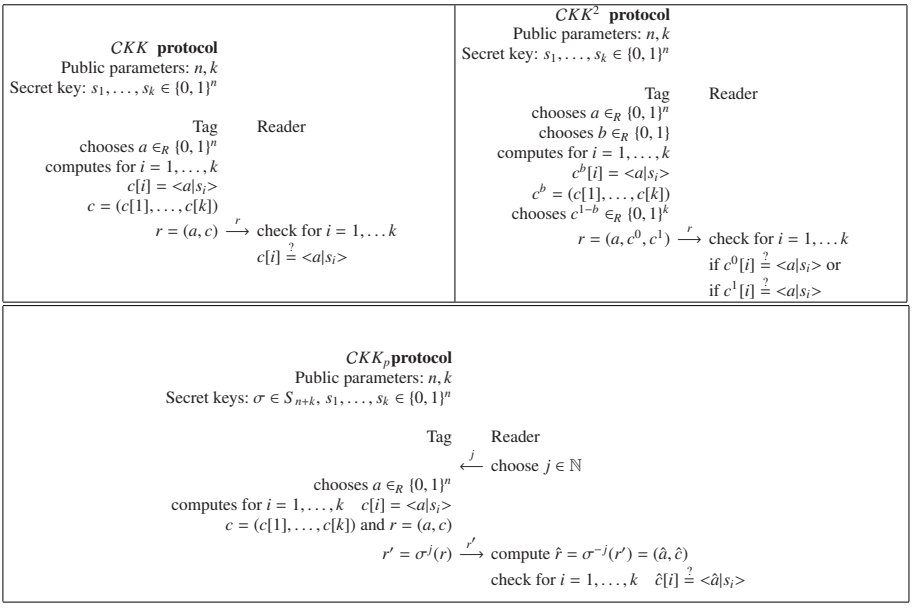
2.1 CKK Tags Description

In ([2]) there are three RFID authentication schemes proposed: CKK, $CKK^2$and CKK_p. In each of them, a Tag T shares with a Reader R secret – vectors: $s_1, \ldots, s_k$ of the length n (in the CKK_p also a permutation $\sigma \in S_{n+k}$). The differences between the schemes are in the way of a Tag responses.

For $n = 128$ parameter k is set to 30. For the rest of the paper we assume the following notation. $x[i]$ is the i-th bit of the vector x. $x \oplus y$ states for the bitwise XOR of vectors x and y. Finally, $<x|y>$ means the inner product of x and y. We write $x \in_R A$ for "x is being picked uniformly at random from the set A". For a vector x we write $\overline{x}$ for a vector with each bit being flipped (from 0 to 1 and vice versa).

By an observation of a tag we mean a sequence of vectors which one can see during tag authentication. For CKK protocol, an observation r of a tag is a par $r = (a, c) =$ (independent part, dependent part), for CKK^2 tag, observation r is a triple $r = (a, c^0, c^1) =$ (independent part, first answer, second answer). One of the c^0, c^1 is

Table 1. *CKK* protocols family description



correct, the second one is a random string. In the case of CKK_p, an observation is a vector of the length $n + k$ where bits of the independent and dependent parts are permuted.

For CKK and CKK^2 protocols, with a sequence of observations of a tag $O = \{r_1, \ldots, r_m\}$ we associate a set of independent parts of observations:

$$O_a = \{a_1, \ldots, a_m\} = \{a_i : a_i \text{ is an independent part of } r_i \in O\}$$

For a given set $B \subset O_a$, which is a basis of the $\{0, 1\}^n$, and a vector $a \in \{0, 1\}^n$, we define a set $rep_B(a)$ as a set of vectors from B, which occurs in the linear combination of representation of a, i. e. $a = \bigoplus_{b \in rep_B(a)} b$. The length of vector a in the basis B is the size of a set $rep_B(a)$.

We say that a vector a has *short representation* if $|rep_B(a)| < k$.

We say that an observation r is *in type* of a tag T if it has the same length as some observation of T and it could be generated as correct T authentication.

3 Passive Attacks on the *CKK* Family

In the current section we describe a passive attacks on each of the CKK protocol. For the rest of this section we use following notation.

In description of all passive attacks, we assume that a set of observations collected by an attacker, Alice:

$$O = \{r_1, \ldots, r_m\}$$

contains $m > n$ tuples. All passive attacks presented are linear of the length of n, but all of them require at least $m \geq n + 1$ to be collected.

3.1 Passive Attack on CKK

Let us assume that an attacker, Alice, have listened to $m \geq n$ executions of the CKK protocol performed by a Tag $\mathcal{T}$, thus collecting a set of observations:

$$O = \{r_1, \ldots, r_m\} = \{(a_1, c_1), \ldots, (a_m, c_m)\}.$$

Then with high probability (for $m = n = 128$ the probability equals to $p = 0.28$; for $m = n + 1$ it is $p = 0.57$, $m = n + 4 \rightarrow p = 0.938$, $m = n + 10 \rightarrow p = 0.999024$; see the appendix for exact formula) there exists a subset $B \subset O_a(B)$ such that $a_i \in B$ are a basis over $\{0, 1\}^n$. Then, by the linearity of the dependent part, Alice can generate proper answers $c(x)$ for any $x \in \{0, 1\}^n$ so that a pair $(x, c(x))$ is accepted by a Reader as proper answer of the Tag $\mathcal{T}$ (i. e. Alice finds a $rep_B(x)$).

More precisely:

Algorithm 1. *Passive attack on CKK protocol*

1. *Collect a set O of m observations of the CKKTag $\mathcal{T}$*
2. *Choose $B \subset O_a(B)$ such that B is independent over $\{0, 1\}^n$*
3. *For any x, compute $rep_B(x)$,*

$$c(x) := \bigoplus_{i \in \{j : a_j \in rep_B(x)\}} c_i$$

4. *Send a pair $(x, c(x))$ – a pair is in the type of $\mathcal{T}$*

Let us notice that if Alice collects m observations then she can generate at most $2^{min\{m,n\}}$ different authentication strings.

3.2 Passive Attack on the CKK^2 Protocol

Now, we assume that Alice has collected following set of observations of a Tag $\mathcal{T}$ authentications:

$$O = \{r_1, \ldots, r_m\} = \{(a_1, c_1^0, c_1^1), \ldots, (a_m, c_m^0, c_m^1)\}.$$

Again, we assume that $m > n$. Then, with overwhelming probability (for $m = n + 10$: $p = 0.999$; for $m = n + k : p \approx 1 - \frac{1}{2^k}$) there exists a subset $O_a(B) \subset O_a$ such that $B = \{a_i : a_i \in O_a(B)\}$ is a basis of $\{0, 1\}^n$.

Conversely to the case of the CKK, one cannot construct proper answers for the Tag $\mathcal{T}$ because one does not know values of b_i – so one does not know which of the values c_i^0, c_i^1 is correct for given a_i (correct in a sense: $c_i^{b_1} = <a_i|s_i>$).

If Alice wants to generate correct tuple (x, c^0, c^1), for any given x which is represented by the l vectors of the basis B ($|rep_B(x)| = l$), she has to pick correct values of b_i so the probability of correct answer is about $2^{-l} = 2^{-|rep_B(x)|}$ (it is not an exact result because sometimes wrong choices of the values b_i can lead to the correct answer). Let us notice that for randomly chosen $x \in \{0, 1\}^n$, $|rep_B(x)| \approx \frac{n}{2} > k$. So this kind of attack is not effective.

Basing on the following observations, we construct an algorithm, which performs passive attack on the CKK^2.

Observation 1. Let us notice that although expected number of basis vectors in representation of random vector from $x \in_R \{0,1\}^n$ is equal to $n/2$, random variable $|rep_B(x)|$ has Binomial distribution, i. e. $P(|rep_B(x)| = l) = \frac{1}{2^n}\binom{n}{l}$, so the probability that a vector x has short representation in base B is equal to:

$$P(|rep_B(x)| \text{ is smaller than k}) = \sum_{i=1}^{k-1} P(rep_B(x) = i)$$

If Alice takes $r = (a, c^0, c^1) \in O$ and $a \notin B$ and a has short representation in B (we assume that $|rep_B(a)| = L$) then Alice can perform exhaustive search of all combinations and find out correct values of b_i (at most 2^L).

Observation 2. If Alice chooses different vectors to a set B then a representation of vectors from observation will change. What is important for us, the length of the representation will also change. So, if Alice could not find any vector that has short representation, she can just pick another basis from $O_a(B)$ and try again.

Observation 3. If Alice takes two observations $r_{s_1} = (a_{s_1}, c^0_{s_1}, c^1_{s_1}), r_{s_2} = (a_{s_2}, c^0_{s_2}, c^1_{s_2})$ then one of the four possible dependent parts: $c^{f_1}_{s_1} \oplus c^{f_2}_{s_2}$, for $f_1, f_2 \in \{0,1\}$ is proper for a vector: $a_{s_1} \oplus a_{s_2}$.

Of course, by taking M observations, and xoring them together, Alice has to find out which of the 2^M combinations of $c^{f_1}_{s_1} \oplus \ldots \oplus c^{f_M}_{s_M}$ for $f_1, \ldots, f_M \in \{0,1\}$ is correct.

Algorithm 2. *Passive attack on CKK^2*

1. *collect a set O of m observations of authentications of the CKK^2 Tag $\mathcal{T}$*
2. *repeat*
   (a) *pick at random set $B \subset O_a$ until B is a basis of $\{0,1\}^n$; $C = \varnothing$*
   (b) *for $j = 1, \ldots, M$*
   check if there exists vectors $x_{f_1}, \ldots, x_{f_j} \in O_a \setminus B$
   such that:

$$|rep_B(x_{f_1} \oplus \ldots \oplus x_{f_j})| = L - |rep_B(x_{f_1} \oplus \ldots \oplus x_{f_j}) \cap \{a_i : i \in C \wedge a_i \in B\}| < k$$

   if short representation is found
   find correct values of b_i by checking at most 2^{L+j} possible cases; add indexes of b_i (index of b_i is i) into a set C
   until $|C| = n$

It should be noticed, that in the step 2 (a) of the algorithm, set B picked at random of the size n is be basis of $\{0,1\}^n$ with probability P_n equal to

$$P_n = \prod_{i=0}^{n-1} \left(1 - 2^{i-n}\right).$$

and it can be checked that $\lim_{n\to\infty} P_n = 0.2887\ldots$ and that the convergence of this sequence is very fast. For instance, we have $P_{10} = 0.28907$ (see the appendix for exact formula).

The step 2 (b) of the above algorithm, we are looking for the vector with short representation. The lower bound on the probability of finding such a short representation is P_s – the probability that in the set $O_a \setminus B$ there will be a vector that has the length of representation smaller than k. P_s is equal to

$$P_s = \sum_{j=1}^{M} \binom{t}{j} \frac{1}{2^n} \sum_{i=0}^{k-1-j} \binom{n}{i},$$

where $t = |O_a \setminus B|$. This is because the probability that a vector v picked at random has the representation of length l with probability $\frac{1}{2^n}\binom{n}{l}$, thus if we are looking for a vector that have the representation smaller than k then we have to sum these probabilities for $i = 0, \ldots, k-1$. The size of the set of vectors that take part in the test $O_a \setminus B$ is equal to t, so we can test $\binom{t}{j}$ vectors created by xoring together j vectors from $O_a \setminus B$.

For proposed in ([2]) tag with $n = 128$, $k = 30$, $M = 3$ and a set of observation of size 512 we have $P_s = 0.000037$, thus the expected number of vectors that should be tested is equal to $\frac{\binom{t}{M}}{P_s} = 2.4945 \cdot 10^{11}$. Simulations shows that home PC is able to find out correct solution in a couple of hours.

3.3 Passive Attack on CKK_p

Let us notice that cheap implementation of the permutations, like it is required by CKK_pprotocol, might be hard. Thus for the practical reasons, a possible scenario is when the Tag perform only one iteration of a permutation in the CKK_pprotocol. The following passive attack on the CKK_pworks only if the Tag permutes response bitstring exactly once in each authentication.

We assume that Alice has collected a set $O = \{o_1, \ldots, o_n\}$ of the observations (independent vectors) of the Tag $\mathcal{T}$. The set O can be treated as a matrix of size $(n + k) \times n$ where each row is a vector obtained from the different observation. Next we have to find any non-degenerate minor B of size $n \times n$ of a matrix O. From the definition of the minor we know that the rows of B are linearly independent. Therefore B is a basis of $\{0, 1\}^n$. Now, if we want compute the correct response bitstring for $v \in_R \{0, 1\}^n$, we need to find its representation in the basis B ($rep_B(v) = \{b_{i_1}, \ldots, b_{i_t}\}$). Next, we perform XOR operation of the vectors $\{o_{i_1}, \ldots, o_{i_t}\}$ form the set O that corresponds to the vectors from $rep_B(v)$. The result of this operation is a proper answer that allows Alice to perform successful authentication. As we can see, we do not need to know the permutation σ because it is hidden in the vectors $\{o_{i_1}, \ldots, o_{i_t}\}$. Form the linearity of the permutation we have $\sigma(w_1 \oplus w_2) = \sigma(w_1) \oplus \sigma(w_2)$ where $w_1, w_2 \in \{0, 1\}^n$, thus we know that result of XOR operation of the vectors $\{o_{i_1}, \ldots, o_{i_t}\}$ will retain the permutation.

4 Active Attacks on the CKK Protocols

In current section we present attacks in which, we allow Alice not only to listen to the communication between a tag and a reader, but also to perform other actions, i. e. retransmit modified messages.

4.1 Repetitive Attack on the CKK Protocol

Let us notice that *CKK* is not immune for the replay attack. Namely, let us assume that an attacker, Alice listens to one execution of the protocol *CKKAuth* and records $r = (a, c)$. Then Alice can act as tag T by sending the same, previously recorded value (a, c).

The only solution for that kind of attack is that the Reader remembers all values a sent by the Tag. Then the Reader can accept only those values which have never been used before. Let us notice that for any proper transmition $r = (a, c)$, c is in fact a linear function of a. If Alice have listened to m transmitions $r_1, \ldots, r_m$, she can send as an authentication string any linear combination of some of recorded values. So a reader should check if a value received from a tag is not a linear combination of the previously transmitted tokens. This kind of attack shows that the lifetime of *CKK*tag is limited by the length of the independent string, i. e. maximum number of secure transmitions is limited by n.

This leads us to the summary that *CKK* tags' lifetime is bounded by n transmitions (if every transmition is linearly independent from the previous ones then there are only n linearly independent vectors on $\{0, 1\}^n$).

4.2 Active Attack on CKK^2

If Alice collects a set of observations $O = \{r_1, \ldots, r_m\} = \{(a_1, c_1^0, c_1^1), \ldots, (a_1, c_m^0, c_m^1)\}$, she does not know which of the values c_j^0, c_j^1 is correct and which is a fake. But then, she can tell the fake from correct value in the following way.

Algorithm 3. *Active attack on* CKK^2

1. *Listen to m authentications of a Tag* $\mathcal{T}$
2. *For each* $i = 1, \ldots, m$ *send to a Reader tuples:* $(a_i, c_i^0, \overline{c_i^1})$, $(a_i, \overline{c_i^0}, c_i^1)$*; one of them will be accepted – remember* b_i

After execution of the algorithm presented above, Alice can generate correct values of a dependent part for any independent part a which is a linear combination of vectors from O_a.

5 Conclusions

We have presented a bunch of attacks (both active and passive) on the *CKK* family of the authentication protocols designed for RFIDs by Cichon et al. ([2]). Our paper shows that modified version of the *CKK* protocol has the same level of security (against active and passive attacks) as more complicated CKK^2 and CKK_p. So, there is no sense to use them anymore. Moreover, our work, together with the original paper ([2]) show that modified *CKK* protocol, where a reader remembers independent parts used by each tag, can be securely used exactly n times, where n is the length of the independent part. If dependent part is of the length k, we have shown, after n executions, an active attacker can easily act as a tag. But after any $i < n$ executions of a protocol every active attacker has only $\frac{1}{2^k}$ probability of successful authorisation.

References

1. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM 50(4), 506–519 (2003)
2. Cichon, J., Klonowski, M., Kutylowski, M.: Privacy protection for rfid's –hidden subset identifiers. In: Indulska, J., Patterson, D.J., Rodden, T., Ott, M. (eds.) PERVASIVE 2008. LNCS, vol. 5013, pp. 298–314. Springer, Heidelberg (2008)
3. Daemen, J., Rijmen, V.: The block cipher bksq. In: Schneier, B., Quisquater, J.-J. (eds.) CARDIS 1998. LNCS, vol. 1820, pp. 236–245. Springer, Heidelberg (2000)
4. Defend, B., Fu, K., Juels, A.: Cryptanalysis of two lightweight rfid authentication schemes. In: PERCOMW 2007: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 211–216. IEEE Computer Society Press, Washington (2007)
5. El Gamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, New York (1985)
6. Gilbert, H., Sibert, H., Robshaw, M.: An active attack against a provably secure lightweight authentication protocol. IEEE Electronic Letters 41, 1169–1170 (2005)
7. Hopper, N.J., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248. Springer, Heidelberg (2001)
8. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621. Springer, Heidelberg (2005)
9. Levieil, É., Fouque, P.-A.: An improved lpn algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006)
10. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
11. Vajda, I., Buttyan, L.: Lightweight authentication protocols for low-cost rfid tags (2003)
12. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)

A Appendix: Mathematical Facts

Let us assume that in one execution of the authentication algorithm, exactly one vector can be eavesdropped. Our aim is to calculate how many executions of the algorithm we need to gather a basis of the vector space created by the eavesdropped vectors.

Lemma 1. *Let V denote vector space created by the vectors of the length n. Let P_{n+k} be the probability that a basis of the vector space V is gathered after collecting $n + k$ vectors. Then*

$$P_{n+k} = \frac{\left(\prod_{i=0}^{n-1}\left(1 - 2^{i-n}\right)\right) \cdot \left(\prod_{i=0}^{k-1}\left(2^{i+n} - 1\right)\right)}{2^{kn} \cdot \left(\prod_{i=1}^{k}\left(2^{i} - 1\right)\right).} \tag{1}$$

Proof. Let us consider simple procedure $\mathcal{S}$. One round of $\mathcal{S}$ consists of three steps:

1. Draw uniformly random vector $v \in \{0,1\}^n$,
2. If v is linearly independent with vectors form set L then put v to the set L, otherwise go to the step 1,
3. If $|L| = n$ then end procedure, else go to step 1.

Our goal is to calculate the probability that the number of rounds in procedure $\mathcal{S}$ is equal to $n + k$.

Let v_i be the vector that is drawn in the i'th round and $B_{i,j}$ be the set of linearly independent vectors that was collected from i'th to the j'th round of the procedure $\mathcal{S}$. Let $A_{i,t}$ denote the event that the set $B_{1,i-1} \cup v_i$ is linearly independent and $|B_{1,i-1} \cup v_i| = t$. Then it is easy to see that

$$P_{n+1} = \sum_{i=1}^{n} P(A_{1,1} \cap \ldots \cap A_{i-1,i-1} \cap \neg A_{i,i} \cap A_{i+1,i} \cap \ldots \cap A_{n+1,n}).$$

It is obvious that the event $\neg A_{i,i}$ has not any impact on the events $A_{j,j}$ for all $j < i$. The event $\neg A_{i,i}$ has not an influence on the probabilities of events $A_{j,t}$ for all $j > i \wedge t \geq i$, because it does not change the size of set L. Therefore we can write as follows

$$P_{n+1} = \sum_{i=0}^{n} P(A_{1,1} \cap \ldots \cap A_{i-1,i-1} \cap A_{i+1,i} \cap \ldots \cap A_{n+1,n}) \cdot P(\neg A_{i,i}).$$

Next observation is that the linearly independent vectors are collected in some n rounds. The numbers of rounds in which vectors are included to the set L have not any impact on the value of probability that n linearly independent vectors are collected. Therefore the probability of collecting n linearly independent vectors is equal for any numbers of rounds in which those vectors are chosen. The value of this probability was introduced in ([2]) and it is equal to

$$p(n) = \prod_{i=0}^{n-1} \left(1 - 2^{i-n}\right).$$

So, we can simplify our formula for the probability that a basis of the vector space V is gathered after collecting $n + 1$ vectors and write it as follows

$$P_{n+1} = p(n) \cdot \sum_{i=1}^{n} P(\neg A_{i,i}).$$

The probability of event $\neg A_{i,t}$ (for all $t \geq i$) is equal to $\frac{2^{i-1}}{2^n} = 2^{i-1-n}$. Thus we can calculate $\sum_{i=1}^{n} P(\neg A_{i,i}) = 1 - 2^{-n}$.

Now we can generalize our reasoning to P_{n+k}. If the number of rounds that are needed to gather a basis of the vector space V is equal to $n + k$ then it will be k events $\neg A_{i_1,t_1}, \ldots, \neg A_{i_k,t_k}$. It should be notice that the order of events $\neg A_{i_j,t_j}$ is important in case of calculating P_{n+k}. Therefore we can write as follows

$$P_{n+k} = p(n) \cdot \sum_{i_1=1}^{n} P(\neg A_{i_1,t_1}) \cdot \left(\sum_{i_2=i_1}^{n} P(\neg A_{i_2,t_2}) \cdot \left(\sum_{i_3=i_2}^{n} P(\neg A_{i_3,t_3}) \cdot \ldots \right) \right).$$

Since $\sum_{a=b}^{n} P(\neg A_{a,t}) = \sum_{a=b}^{n} 2^{a-1-n} = 1 - 2^{b-1-n}$ we can simplify formula for P_{n+k} and write it as follows

$$P_{n+k} = p(n) \cdot \frac{\prod_{i=0}^{k-1} \left(2^{i+n} - 1\right)}{2^{kn} \cdot \left(\prod_{i=1}^{k} (2^i - 1)\right)}.$$

Now putting formula for $p(n)$ to this equation we get what we want to prove. □

Fact 4. *Let K denote the number of vectors gathered above n to collect a basis of vector space V. The expected value of K is given by formula*

$$E[K] = \sum_{k=0}^{\infty} k \cdot \frac{\left(\prod_{i=0}^{n-1} \left(1 - 2^{i-n}\right)\right) \cdot \left(\prod_{i=0}^{k-1} \left(2^{i+n} - 1\right)\right)}{2^{kn} \cdot \left(\prod_{i=1}^{k} (2^i - 1)\right)}$$

and it can be checked that $\lim_{n\to\infty} E[K] = 1.6067\ldots$ *and that the convergence of this sequence is very fast. For example, for* $n = 10$ *we have* $E[K] = 1.60572$*, for* $n = 20$ *we have* $E[K] = 1.60669$ *and for* $n = 25$ *we have* $E[K] = 1.6067$. □

On Backoff in Fading Wireless Channels

SeonYeong Han and Nael B. Abu-Ghazaleh

Computer Science Dept.
State University of New York at Binghamton
and
School of Computer Science
Carnegie Mellon University - Qatar
{shan6@,nael@cs.}binghamton.edu

Abstract. We consider the impact of transmission errors on the backoff algorithm behavior in the IEEE 802.11 protocol. Specifically, since the backoff algorithm assumes that all packet losses are due to collisions, it unnecessarily backs off when a packet is lost due to a transmission error. Two performance problems arise as a result: (1) low throughput, due to unnecessary loss of transmission time; and (2) unfairness when two competing links have different transmission error rates. In this paper, we characterize this problem and propose three solutions to it. The solutions aim to provide discrimination between transmission errors and collisions such that the sender can back off appropriately. The first algorithm relies on receiver discrimination and feedback; the receiving radio can in many instances differentiate between collisions and transmission errors. The second algorithm estimates the clear channel quality, and backs off if the observed quality deviates from the clear channel quality (indicating collisions). The third algorithm develops the probability of collision as a function of the number of observed idle slots during contention, and uses this probability to control the backoff algorithm. We show via simulation that the techniques significantly improve both performance and fairness of IEEE 802.11 in the presence of transmission errors.

1 Introduction

The IEEE 802.11 MAC protocol [1] is the *de facto* standard for wireless LANs, including ad hoc and mesh networks. It is a contention based protocol that uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to reduce the probability of collisions. In contention protocols, collisions cannot be eliminated; this is especially true for wireless networks due to the well-known hidden terminal problem [2,3]. Thus, an important component of contention MAC protocols is the backoff mechanism which is used to regulate the offered load to the shared channel in the presence of contention. Specifically, when a collision occurs *Binary Exponential Backoff (BEB)*, is invoked, typically doubling the size of the backoff window. BEB is used in other contention protocols such as Ethernet.

* This work is partially supported by NSF grant CNS-0454298.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 251–264, 2008.
© Springer-Verlag Berlin Heidelberg 2008

IEEE 802.11 interprets all packet losses as collisions and invokes the BEB algorithm. However, in the presence of wireless transmission errors, the BEB protocol is invoked unnecessarily (since a transmission error is not an indication of contention), leading to the following two performance problems:

(1) **Inefficient use of the available bandwidth:** this is a consequence of unnecessarily increasing the backoff window. This problem is exacerbated when one considers that contention occurs in IEEE 802.11 using the lowest transmission rate, to allow fair contention among connections with heterogeneous rates. As a result, at higher rates, data packet transmission time becomes shorter, but the backoff period stays the same as the lowest rate;
(2) **Unfairness:** when two links with different loss rates are in interference range of each other, unfairness arises. The weaker link backs off more frequently due to transmission errors, creating unfair competition for the medium and long-term unfairness. We characterize the impact of fading on IEEE 802.11 performance under different scenarios in Section 2.

This paper contributes three solutions for remedying these problems. A successful solution should discriminate between losses due to collisions and those due to transmission errors. This discrimination does not necessarily have to be at the granularity of the individual transmission; rather, the technique should provide insight into the probability of a loss in the aggregate being due to collisions. We investigate the following three solutions, which are presented in more detail in Section 3.

1. Receiver based discrimination: in this solution, the receiver uses any information available to it to determine the cause of the packet loss. Increasingly, the physical layer at the receiver is able to provide information about the transmission that is helpful in speculating on the reason for the loss. The speculation results are fed back to the sender on subsequent acknowledgments, allowing it to adjust its backoff window proportionately. This approach relies critically on the discrimination mechanism at the receiver and the information available from the wireless card.
2. Link Quality Estimation: in this solution, the sender maintains a running estimate of the clear channel link quality (the expected loss rate in the absence of contention from other sources). The backoff window is then increased in proportion to the loss rate being observed vs. that expected by the link quality. However, estimating the clear channel link quality while the network is active is difficult. We take an approach in which we use the minimum loss rate period over a period of time as an estimate of the clear channel link quality.
3. Idle Slot Collision Probability Estimation: Heusse et al. [4] observed that the number of idle slots in a contention period is indicative of the amount of local contention for the use of the channel. We adapt the approach to estimate the probability of collision as a function of the number of idle slots observed. With an estimate of the collision probability, we can estimate the

number of extra losses that are due to transmission errors and adjust the backoff accordingly.

Section 4 presents a simulation-based evaluation of the proposed approaches. The experiments show that all three approaches are able to address the problem, coming close to the performance of a perfect predictor. Section 5 overviews related work. Finally, Section 6 presents some concluding remarks.

2 Impact of Fading on Binary Exponential Backoff

The backoff mechanism regulates the offered load to the shared medium. Backoff algorithms maintain a contention window value in units of fixed-size slots, to determine how long to wait before transmission. In IEEE 802.11, there is a minimum contention window CW_{min} that is used after a successful transmission. CW is doubled whenever a packet loss occurs until it reaches CW_{max}. After every transmission a node picks a number of slots uniformly distributed in the range $[0,CW]$ as its backoff window.

The underlying assumption in these backoff algorithms is that all packet losses are due to collisions. This assumption holds true in wired shared media where transmission errors are exceptionally rare, but not in wireless environments where they are common. When transmission losses occur, backoff is invoked unnecessarily, leading to significant inefficiency and giving rise to long-term unfairness among links with different qualities.

We define *efficiency* to be the ratio of the observed throughput to the throughput of an ideal backoff algorithm that backs off when collisions occur, but not when transmission errors occur. Figure 1 shows the efficiency on a single hop link as a function of the link quality (the probability of successful transmission) for two different transmission rates and packet sizes. Clearly, there is a large drop in throughput as the link quality drops *beyond the loss that results from the loss of the packets to transmission errors*. Whenever a transmission error

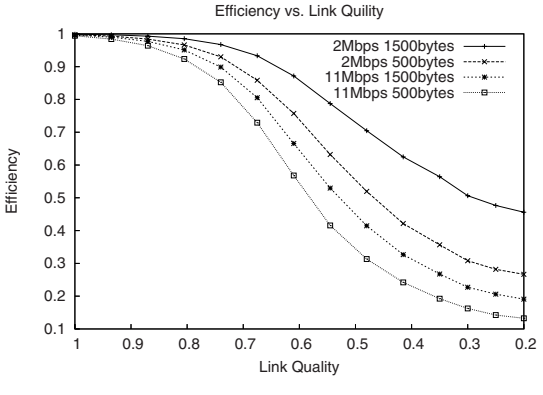


Fig. 1. Efficiency vs. Link Quality

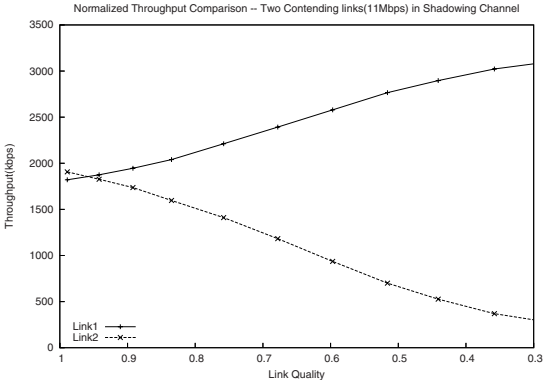


Fig. 2. Unfairness Problem

occurs, the contention window is doubled unnecessarily. The problem is worse when the packet size is small since the yield from each contention period drops. Further, since contention is carried out at the lowest transmission rate (for compatibility and fairness among transmitters with different rates), the problem also becomes worse when the transmission rate increases. Because the probability of successive transmission errors increases as the link quality degrades (leading to exponential backoff), the efficiency degradation is not linear.

Unnecessary backoff in response to transmission errors can also give rise to unfairness. When multiple links compete, if the backoff algorithm is not biased towards either, then long term fairness will be achieved. However, in the presence of transmission errors, two links with different loss rates experience different average backoff values. This causes unfair competition in accessing the medium and long term unfairness results. Figure 2 demonstrates this effect between two competing single hop flows. The link quality for the first flow is fixed at 0.95, while the quality of the second link is varied. The figure plots the normalized throughput (throughput divided by link quality); the normalization is done to remove the effect of the lost packets and provides an estimate of the actual transmissions that each flow receives. Clearly, as the link quality disparity increases, the higher quality links starts dominating the available bandwidth. Discriminating between transmission errors and collisions can mitigate this problem because the weaker quality link is not penalized by the backoff algorithm for transmission errors (which are not indicative of collisions).

3 Proposed Solutions

In this section, we discuss three approaches for informed back off for CSMA based wireless networks. The goal of our solutions is discriminate between transmission losses and collisions so that the back off algorithm increases back off only when collisions occur. An important observation is that this back off does not have to be at the granularity of the individual packet. Instead, an estimate of the

percentage of losses due to collisions is sufficient to guide the behavior of the back off algorithm.

Overall, discriminating between transmission errors and collisions is challenging as this information is rarely explicitly and directly discernible for a given transmission. However, often the combined views at the sender and receiver can be used to intelligently and speculatively determine the causes behind packet losses. From the sender's perspective, little information is available about a given packet transmission without receiver feedback. On the other hand, at the receiver, a given lost transmission may be undetected (e.g., due to a deep signal fade or high interference or noise) or partially detected (a part of the packet is corrupted). Both the sender and receiver may collect information about the channel via carrier sense when they are not transmitting to each other; however, the state of the channel at the receiver is more important than the state of the channel at the sender. In the remainder of this section, we propose three solutions to this problem.

3.1 Receiver Based Discrimination (RBD)

Receiver-based discrimination uses the information available at the receiver to identify the cause of a packet loss. At the physical layer, detailed information is available during the packet reception that can allow effective speculation on the reasons behind the packet loss (e.g., [5]). However, commercial wireless cards differ significantly in the information they expose to upper layers. Furthermore, some events are more difficult to detect than others (e.g., a collision or fade during the PLCP header causes complete loss of the packet and no information is available). Therefore, the available information, and the success rate for the speculation, varies significantly with the underlying hardware and drivers. It is possible to explore different alternative algorithms for discrimination based on the information available to the receiver.

Discrimination Mechanism: As an example for this approach, we use a mechanism suggested by Burns et al. for collision detection [6]. For this approach to be effective, the hardware of the receiver should be able to detect the existence of a new packet even though it is currently receiving another packet (indicating a collision). If the receiver is unable to detect the new packet header, it cannot decide if the loss is due to a collision or error. Other approaches for discrimination are possible (e.g., based on the observed RSSI). Once the receiver detects a collision, the speculation results are returned to the sender so that it can adjust its back off behavior. We feedback this information opportunistically by including it on subsequent ACKs.

Modified Backoff Algorithm: Once the collision information is received at the sender, the following approach is used to increase the contention window. The conditional collision probability (CCP) that a lost packet is lost due to a collision, rather than a transmission error, is estimated as follows. The receiver feeds back on the ACK N_{ecol}– the number of packets estimated to have been lost to collisions for a predetermined observation window (in time or number of packets).

The estimated CCP is $\frac{N_{ecol}}{N_{lost}}$, where N_{lost} is the total number of lost packets (to errors or collisions) in the same window. Since $N_{lost} = N_{transmit} - N_{success}$, CCP is $\frac{N_{ecol}}{N_{transmit} - N_{success}}$. When a transmission is lost, we back off with probability CCP.

Note that RDB generally underestimates CCP because it can fail to detect some collisions. Moreover, our implementation does not account for ACK losses (which in effect considers all of them to be non-collision losses). However, ACK losses due to collisions are relatively rare because of the small size of the ACK packet. Furthermore, nothing prevents ACK packet collision detection using the same approach.

3.2 Link Quality Estimation (LQE)

In this approach, we first estimate the clear channel link quality (CCLQ) which represents the loss rate on the channel in the absence of collisions. Once that is estimated, the expected probability of loss for each transmission can be developed. Over a certain window, again measured in terms of time or number of transmissions, we expect a number of transmission errors based on the number of attempted transmissions and the estimated link quality. Losses exceeding this number can be attributed to collisions and the back off window adjusted accordingly. Note that the approach can be made robust for different packet sizes and/or different transmission rates (e.g., by estimating the bit error rate instead of the packet loss rate). Different flavors of LQE may be developed based on the approach for estimating CCLQ, and how the contention window is adjusted.

Estimating Link Quality: CCLQ may be estimated off-line for static mesh networks by running clear channel measurements while the network is idle. However, this approach is not suitable for dynamic environments and does not adapt to the time-varying nature of link quality. The challenge in dynamically estimating CCLQ is that the channel is not idle while the network is active. Thus, simply tracking the percentage of packets received correctly counts both the losses due to transmission errors and collisions, under-estimating the link quality. We use the *highest observed* link quality value over a fixed number of measurement windows as the CCLQ. Our intuition is that this high link quality occurs due to a window with few or no collisions. However, the estimate remains approximate: if no period is free of collisions, then the quality is under estimated. Thus, the estimate of CCLQ is heuristic; the heuristic may be improved based on empirical evaluation. Further, LQE expects that the link remains stable over multiple windows and is therefore slow in tracking a dynamically changing window (e.g., due to mobility).

Modified Backoff Algorithm: In a given window the probability of loss is computed as the ratio of lost packets to total packets P_{loss}. If $P_{loss} \leq cclq$, we update CCLQ to be equal to P_{loss}. However, if $P_{loss} > cclq$, we have some collisions. To compute the conditional collision probability, note that

$$P_{loss} = cclq + P_c - cclq * P_c$$
$$P_c = \frac{P_{loss} - cclq}{1 - cclq}$$

where P_c is the probability of collision. The conditional collision probability (CCP) is then $\frac{P_c}{P_{loss}}$; when a transmission is lost, we back off with probability CCP.

3.3 Idle Slot Collision Probability Estimation (ISCPE)

Another approach to estimating the conditional collision probability (CCP) relies on the following observation due to Heusse et al [4]. Specifically, they observe that the degree of contention, and hence the probability of collisions, is a function of the number of idle slots after every successful transmission. They use this observation to derive an optimized back off algorithm called IdleSense. IdleSense significantly outperforms Binary Exponential Backoff, but is not compatible with it. Furthermore, IdleSense does not consider transmission losses.

Observing Idle Probability: Because back off algorithm pause the decrement of the back off counter whenever a busy channel is sensed, the length of a run of continuous idle slots is a good indicator of contention level around a receiver. Let the probability that each slot is assigned to some node be p, and $q = 1 - p$. Then, the probability that the k'th trial is the first success is

$$Pr(X = k) = q^{k-1}p \tag{1}$$

for $k = 1, 2, 3, ...$

The random variable X that indicates the number of trials before the first successful slot is geometrically distributed with expected value $E(X) = \frac{1}{p}$. The average observed length of a run of continuous idle slots, $L + 1$, is also $E(X)$. Then, $p = 1/(L + 1)$ and $q = L/(L + 1)$. Because q is the idle probability of a slot,

$$q = (1 - P_e)^N \tag{2}$$

where N represents the number of contending nodes, and P_e is the attempt probability per slot. From Eq. 2, $P_e = 1 - q^{\frac{1}{N}}$. The idle probability, P_i, is then $(1 - P_e)^N = q$.

Because each node senses the idle channel first and then takes the following slot to transmit if its backoff is zero, the minimum L is 1. Thus, $L = 1$ means that the obtained idle probability is overestimated; this is a limitation of idle slot based solution.

Estimating Collision Probability: According to [4], the successful transmission rate can be estimated as

$$P_t = N \cdot P_e(1 - P_e)^{N-1} \tag{3}$$

A slot where a collision occurs is one where more than one transmission occurs. More formally,

$$P_c = 1 - P_t - P_i \tag{4}$$

We can express P_c using P_i as

$$P_c = 1 - N(1 - P_i^{\frac{1}{N}})P_i^{\frac{N-1}{N}} - P_i \tag{5}$$

N can be found by sensing the channel and P_i can be found by observing the number of idle slot after every transmission. Thus, from the computed P_c we can estimate CCP, and adjust the back off with that probability on every lost packet.

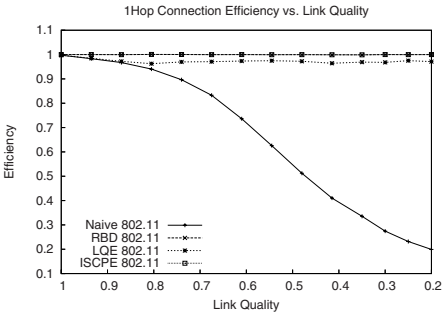
Estimating the Number of Neighbors: Though ISCPE mostly relies on local information, channel observation is still necessary to determine the number of neighbors, which is needed to calculate the transmission probability and collision probability from the idle probability. In a mobile environment, the observed number of neighbors may be either larger or smaller than the exact number because of tracking lag. The effect of inaccuracy in estimating the number of neighbors will be evaluated in the following section.

4 Performance Evaluation

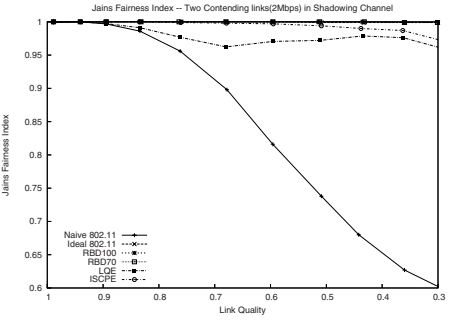
In this section we evaluate the proposed solutions using simulations. For all simulations, we use the NS-2 network simulator [7]. Unless otherwise indicated, we use the log-normal model to simulate a fading channel. We also investigated generating fading losses by training hidden Markov models trained with collected wireless traces, but no appreciable changes in the general trends were observed; therefore we elected to demonstrate the solutions using the more controllable log-normal model.

The first study revisits the case of a single hop flow with varying link quality. Figure 3(a) shows that all three solutions successfully address the backoff problem in the 1-hop case. Since there are no collisions, RBD achieves ideal performance since it detects no collisions and assumes that all losses are due to transmission errors (which is the case). Also, ISCPE will detect an always idle link, correctly predicting that there are no collisions. LQE does not achieve ideal behavior because it incorrectly guesses that a collision occurs whenever the number of losses in a window is above that in the lowest detected window. In other words, since errors do not occur at a constant rate, LQE mispredicts collisions, and therefore its performance is slightly worse than the other schemes.

In the second experiment, we compare the fairness achieved by two contending links using the different backoff algorithms. The quality of one link is fixed at 0.95, while the quality of the other is varied (x-axis). Figure 3(b) shows the fairness improvement of the three solutions. ISCPE and RBD achieve ideal fairness (Jain's index of 1). RBD calculates the CCD by the periodic observation of the number of transmitted, successfully received, and collided packets. The



(a) Efficiency vs. Link Quality



(b) Fairness Comparison in Two Contending Links (11Mbps)

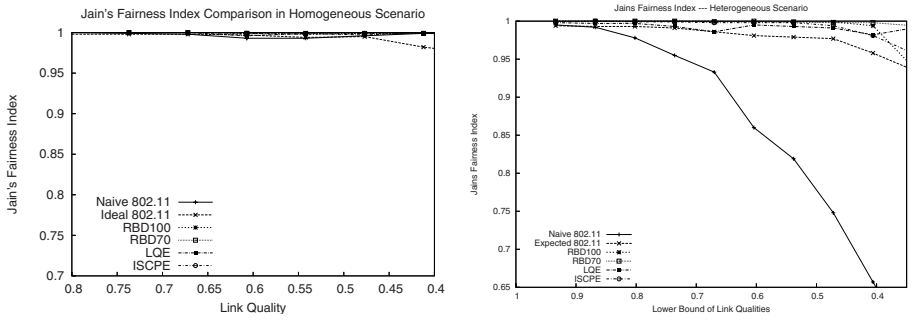
Fig. 3. Effect of the Solutions

level of accuracy of collision detection affects fairness,especially with low link quality. If only 70% of collisions are detected by the receiver,the CCD is underestimated, leading to a smaller contention window than ideal. Surprisingly, a lower accuracy of the RBD does not appreciably harm fairness. Moreover, LQE shows sensitivity to the asymmetric measurement error as link quality decreases. This is because LQE is a solution based on estimated link quality, which is inaccurate if collision loss events occur frequently. However, as can be seen in the figure,fairness improves significantly in all approaches.

We considered a WLAN scenario where five senders and five receivers are in range of each other. In this scenario, neither hidden terminals nor exposed terminals exist; however, collisions due to concurrent transmissions (two or more nodes attempting to transmit in the same slot) can occur [8]; persistent or transient unfairness due to hidden terminals cannot be solved using the schemes in this paper, which focus only on avoiding backoff when transmission losses occur. By fixing the number of transmitters and the transmission rate, the probability of collisions is fixed [8]. In a homogeneous scenario, all five links have the same link quality. In a heterogeneous scenario, link qualities are uniformly distributed in the range of $[Min(p_j), Max(p_j)]$. Each point represents an average of twenty simulation runs to tightly bound the confidence intervals.

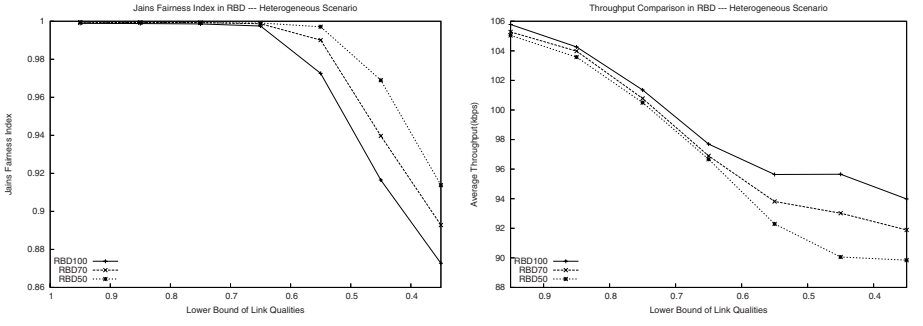
Figure 4(a) shows Jain's fairness index for the homogeneous scenario; unfairness is not present because all links have an equal opportunity to access a wireless channel and no persistent asymmetry exists. Jain's fairness index in a heterogeneous scenario is shown in Figure 4(b), where the x-axis represents the lower bound of link quality and the y-axis represents the fairness index. In this case, all three proposed solutions dramatically improve unfairness.

Achieving fairness may result in degradation of overall throughput. The primary reason behind this degradation is that by giving more chances to the weaker links to transmit, we end up losing more of the transmitted packets due to errors, harming overall throughput. Thus, aggregate throughput is reduced even though weak links achieve improved throughput.



(a) Homogeneous Scenario (b) Heterogeneous Scenario

Fig. 4. Jain's Fairness Index



(a) Jain's Fairness Comparison (b) Throughput Comparison

Fig. 5. Comparison in Several Accuracy Level of RBD

Figure 5(a) and 5(b) analyze the impact of discrimination accuracy on the performance of RBD in a heterogeneous scenario. RBDxx indicates that only xx%of collisions are detected in the receiver. For the case of RBD50, where only half of the collisions are detected, the fairness increases due to the small contention window but average throughput decrease due to collision losses. In fact, our simulation result showed that the number of collisions for RBD70 and RBD50 increase by 3.7% and 5% respectively compared to that for RBD100.

Figure 6 shows aggregate throughput of all proposed solutions. Because the strong links cannot dominate the channel in the solutions, the aggregate throughput of the proposed solutions is lower than that of the naive 802.11.

Figure 7(a) and 7(b) show the fairness index and throughput as the node density is increased when the lower bound of link qualities is fixed to 0.6. By increasing node density, we increase the collision probability. Naive 802.11 shows pretty stable unfairness through the various node density, though the aggregate throughput decreases. This is because the collision losses in a high density scenario happen fairly to the senders, while fading losses happen unfairly. However,

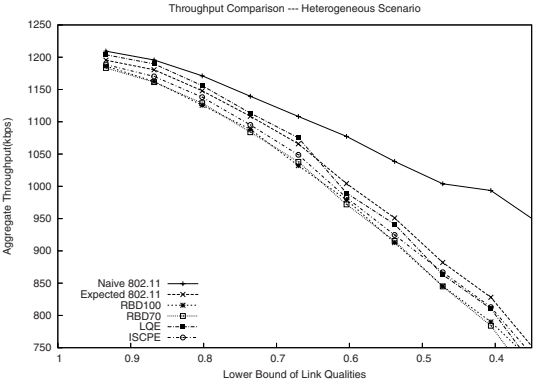
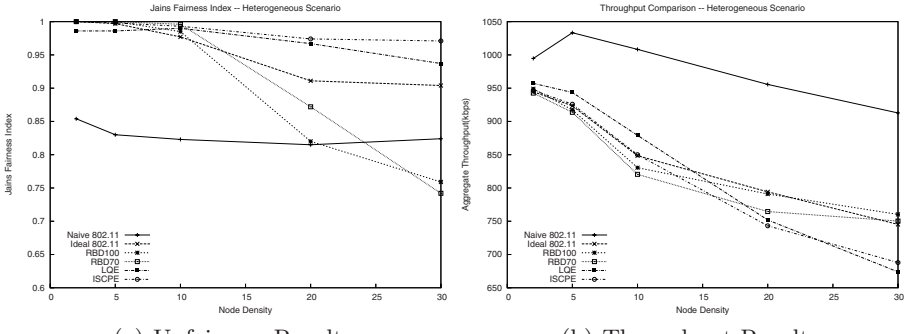


Fig. 6. Throughput Comparison of Solutions



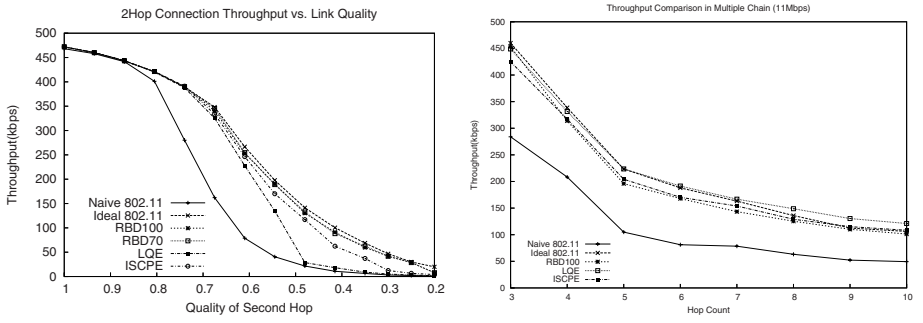
(a) Unfairness Result (b) Throughput Result

Fig. 7. Unfairness And Throughput Comparison As The Node Density Increases

the high collision loss due to high node density affects the performance of each solution. RBD needs to refer the returned feedback to estimate collision rate. As the node density increase, the feedback is more likely to be collided with, which results in incomplete feedback. LQE and ISCPE show stable improvement in fairness, because they do not depend on feedback.

Figure 8(a) presents a 2-hop connection throughput improvement for a 500 bytes packet size at 11Mbps, where the first hop has a 95% link quality and the second hop has various link qualities which are represented on the x-axis. A problem with unfairness in chains where a stronger link is upstream of a weaker link is that the stronger link wins more often, creating a mismatch between input and output at intermediate nodes and consequently packet drops. Increasing fairness, significantly improve performance by eliminating this effect.

The 2-hop scenario is extended to multiple chain connection as shown in Figure 8(b). Link qualities for each hop were assigned randomly. Each point represents an average of twenty simulation runs. All proposed solutions show improved throughput over any hop counts due to increasing fairness among hops,



(a) Throughput in a Two-hop Connection (b) Throughput in a Multiple Chain Connection

Fig. 8. Throughput Improvement in a Multi-hop Connection(11Mbps)

which provides further support for using mechanisms that intelligently back off in the presence of transmission losses.

5 Related Work

Chua and Lye discuss the need to distinguish collisions from errors in time-varying channels [9]. They observe that backoff algorithms should be modified to consider packets that fail due to channel errors, but offer no solution to the problem.

Nadeem et al [10] modify Bianchi's model to study noisy environments. They proposed $smart_{BEB}$, which adjusts the contention window in increments based on the loss probability p. The proposed solution needs time to achieve optimal value. Furthermore, it represents a completely different backoff algorithm that does not inter-operate with the standard IEEE 802.11. The IdleSense algorithm [4] is a similar algorithm to that proposed by Nadeem et al; it also is not inter-operable with IEEE 802.11.

Discrimination of wireless errors from congestion errors has received significant attention with respect to TCP. Since TCP uses packet loss events as an indicator of congestion, it also suffers from undetected wireless losses. Specifically, when a transmission error cause s apacket to be lost, TCP incorrectly activates its congestion control mechanism, resulting in poor performance. Much of the literature proposes end-to-end solutions that analyze the Round Trip Time of received packets [11,12] or use excplicit congestion notification to distinguish the cause of loss [13]. End-to-end solutions do not help to solve the unfairness problem in the MAC layer because they do not influence the backoff algorithm.

6 Conclusion

In this paper, we studied the impact of transmission losses on the backoff mechanism of IEEE 802.11. Specifically, the backoff algorithm treats all losses as

collision losses, leading to unecessary backoff. The problem leads to two negative side effects: loss of channel time and unfairness. We proposed three different solutions to the problem that attempt to discriminate between transmission errors and collisions then backoff only when a collision occurs. Specifically, Receiver Based Discrimination uses available information at the receiver to determine if a packet loss was due to collisions or errors, and feeds this information back to the sender. Link Quality Estimation estimates the clear channel link quality and backs off whenever the number of losses in a window exceeds the number expected by the link quality (the deviation indicating potential collisions). Finally, we proposed an Idle Slot Collision Probability Estimation mechanism that uses recent results that show that the number of observed idle slots can be used to estimate collision probability and thus to guide backoff behavior. Simulation that the proposed approaches significantly improve the problem and increase throughput and fairness overall.

References

1. IEEE 802.11 – The Working Group for Wireless LANs: Ieee 802.11 standard, edition (1999) Including IEEE 802.11a and IEEE 802.11b Extension (1999), http://www.ieee802.org/11/index.html
2. Tobagi, F., Klienrock, L.: Packet switching in radio channels: Part II: The hidden terminal problem in carrier sense multiple access and the busy tone solution. IEEE Transactions on Communication, 1417–1433 (1975)
3. Garetto, M., Shi, J., Knightly, E.W.: Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In: MobiCom 2005, pp. 200–214. ACM Press, New York (2005)
4. Heusse, M., Rousseau, F., Guillier, R., Duda, A.: Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. SIGCOMM Comput. Commun. Rev. 35(4), 121–132 (2005)
5. Woo, G., Kheradpour, P., Shen, D., Katabi, D.: Beyond the bits: Cooperative packet recovery using physical layer information. In: Proc. ACM International Conference on Mobile Computing and Networking (Mobicom) (2007)
6. Burns, L., Podell, A., Fisher, D., Ramachandran, R. (Radio based collision detection for wireless communication system) US Patent Issued on August 12 (1997)
7. Information Sciences Institute: NS-2 network simulator. Software Package (2005), http://www.isi.edu/nsnam/ns/
8. Bianchi, G.: Performance analysis of the ieee 802.11 distributed coordinationfunction. IEEE Journal on Selected Areas in Communications (2000)
9. Chua, K.C., Lye, K.: Backoff considerations in csma/cd lan with single time-varying channel. Electronics Letters 27(9), 747–748 (1991)
10. Nadeem, T., Agrawala, A.: Ieee 802.11 dcf enhancements for noisy environments. In: International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (2004)
11. Biaz, S., Vaidya, N.H.: Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. In: ASSET 1999: Proceedings of the 1999 IEEE Symposium on Application - Specific Systems and Software Engineering and Technology, p. 10. IEEE Computer Society Press, Washington (1999)

12. Li, Y., Su, F., Fan, Y., Xu, H.: End-to-end differentiation of congestion and wireless losses using a fuzzy arithmetic based on relative entropy. In: International Conference on Systems and Networks Communication (ICSNC 2006), p. 15 (2006)
13. Biaz, S., Vaidya, N.H.: "de-randomizing" congestion losses to improve tcp performance over wired-wireless networks. IEEE/ACM Trans. Netw. 13(3), 596–608 (2005)

TSLA: A QoS-Aware On-Demand Routing Protocol for Mobile Ad Hoc Networks

C. Mbarushimana and A. Shahrabi

School of Engineering and Computing
Glasgow Caledonian University
Glasgow G4 0BA, U.K.
{Consolee.Mbarushimana,A.Shahrabi}@gcal.ac.uk

Abstract. The complexity of Mobile Ad Hoc Networks (MANETs) has led to the extensive research in the development of their routing protocols as reported in literature. Although most of the proposed routing protocols are based on the shortest path algorithm, some other metrics like load and network congestion have also been considered in some other research. However, with the introduction of traffic differentiation in IEEE 802.11e, congestion effect becomes more distinct as the nodes with delay-sensitive multimedia applications tend to be busy for prolonged periods. This has received little attention in the literature to date. In this paper, we first expose that the performance of MANETs routing protocols is highly dependent on the type of traffic generated or routed by intermediate nodes. We then propose Type of Service and Load Aware routing protocol (TSLA), an enhancement to AODV that uses both the traffic load and the type of service as additional metrics. To our knowledge, TSLA is the first to avoid congestion by distributing the load over a potentially greater area and conducting the traffic through less busy nodes and, therefore, less congested routes. Our simulation study reveals a persistent improvement in throughput and packet delay of both low and high priority traffic.

Keywords: 802.11e, MANETs, QoS, routing protocol.

1 Introduction

Due to MANETs dynamic characteristics, their routing protocols have received a great deal of attention over the past few years. They are mainly classified as reactive (e.g., AODV and DSR) and proactive (e.g., DSDV and OLSR) routing protocols. In several studies carried out to evaluate their performance [1], [2], the negative points of reactive protocols were found out to be high delay and packet loss due to stale routes, whereas the performance of proactive protocols is very much affected by their routing overhead. The shortest path method used by most of the existing protocols in route selection does not provide optimal results, especially if the primary route is congested. This issue has led research on congestion and load aware routing protocols based on the fact that besides route failures, network congestion is the other main cause of packet loss in MANETs.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 265–278, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Today's networks are more prone to congestion due to large volumes of UDP-based multimedia traffic (e.g., voice, video). UDP flows do not typically back off when they encounter congestion. They aggressively consume more bandwidth than TCP flows. The traffic differentiation introduced by 802.11e assigns high priority to UDP-based delay-sensitive multimedia applications. This is exacerbated by the prolonged duration of data transmission in multimedia applications. For example, the PSTN is sized for average call duration of two minutes, but VoIP connections usually last longer than this.

Several load aware routing protocols for MANETs have been reported in the literature [3], [4], [5], [7], [10]. However, to our best knowledge, the effect of the type of service of the traffic hold in queues of the intermediates nodes has not been investigated. In this paper, we first explore how the type of traffic affects the congestion status of a node, along with the load. We then propose TSLA; a new routing protocol which is a cross-layer enhancement to AODV using both the traffic load and the type of service (ToS) as additional metrics. In TSLA, MAC layer notifies the network layer about the amount and type of service of traffic held in its queue. Based on this information, nodes can adjust and advertise their congestion level to neighbouring nodes. Using OPNET simulation, we then comparatively evaluate the performance of the proposed scheme and AODV.

The rest of the paper is organised as follows. Section 2 briefly reviews the related works. In Section 3, we explain our proposed routing protocol in details. We present the performance evaluation and simulations results in Section 4. Finally, some concluding remarks are given in Section 5.

2 Background

2.1 Related Work

Congestion avoidance routing has been investigated over the past years. Lee and Gerla proposed a dynamic load aware on demand routing protocol (DLAR) in [4]. The destination chooses the least congested path based on the load information attached in the RREQs and sends a RREP back to the source via the selected route. Another scheme is proposed in [9], which relies on intermediate nodes not to reply to route request messages when their load exceeds a certain threshold. A different approach of load balancing is used by the Dynamic Load-aware Based Load-balanced (DLBL) routing proposed in [11] to distribute the overhead among all intermediate nodes. Saigal et al. proposed load aware routing in ad hoc (LARA) in [6], which uses a metric called traffic density to represent the degree of contention at the MAC layer.

MAC layer channel contention information, number of packets in the interface queue, and the traditional hop count are the three metrics used for route selection by CSLAR (contention sensitive load aware routing protocol) proposed in [5]. A similar protocol, Contention and Queue-aware Routing (CQR) was proposed in [3], which bases its route selection on the queue size and the contention window. A congestion adaptive routing (CRP) in which a route is adaptively changeable based on the congestion status of the network is proposed in [7]. Similar to the others, the number of data packets in the node buffer is used to quantify its congestion status.

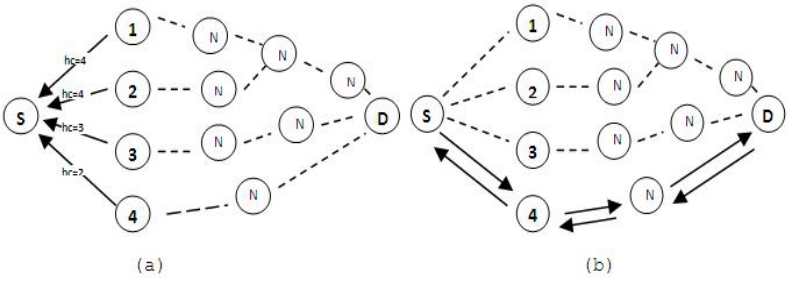


Fig. 1. AODV Route Selectiion

Some other approaches using metrics indirectly related to the network load have been proposed. Like the Load Balance Routing using Packet Success Rate proposed in and Lifetime-aware Leisure Degree Adaptive Routing protocol (L-LDAR) [10]. In [8], Ye et al. investigated the possibilities of spatially separating concurrent TCP connections using congestion aware routing.

The above reported congestion aware approaches converge in evaluating the nodes level of activity by measuring either the load or the delay. However, none of the reported studies has evaluated the effect the ToS of the traffic carried by the nodes has on the routing algorithm. This aspect is investigated in this paper.

2.2 QoS-Aware MANETs

There has been a tremendous increase in multimedia applications over the past few years. This type of applications requires QoS guarantees in terms of delay, bandwidth, packet loss and jitter. With the prospects of future MANETs commercial applications, it is desirable to support these services in MANETs as well.

The IEEE 802.11e EDCA provides a priority scheme to differentiate different access categories (ACs) by classifying the arbitration interframe space (AIFS), and the initial (CW_{min}) and maximum (CW_{max}) contention window sizes in the backoff procedures. EDCA uses different AIFS for each AC to achieve the access differentiation, where the $AIFS_i$ for a given AC_i is given by

$$AIFS_i = AIFSN_i \times \delta + SIFS \tag{1}$$

where the $AIFSN_i$ is an integer dependent on each AC and δ is the time interval of a slot. With small values AIFSN, high priority classes start decreasing their backoff counter earlier than low priority classes. The backoff interval (BI) is randomly chosen in the range [0, CW_i] where $CW_i=2^{k_i}\text{-}^1 CW_{min}$ (k is the backoff stage). High priority classes are given smaller values of CW_{min} and CW_{max}, which result in shorter backoff intervals. In real life, multimedia traffic like voice (AC_3) and video (AC_2) are assigned higher priority over best effort (AC_1) TCP based applications (e-mail, FTP).

2.3 Ad Hoc On-Demand Distance Vector Routing Protocol (AODV)

AODV minimizes the number of broadcasts by creating routes on-demand. Figure 1 illustrates a simple route discovery in AODV. The node S seeking a route to a

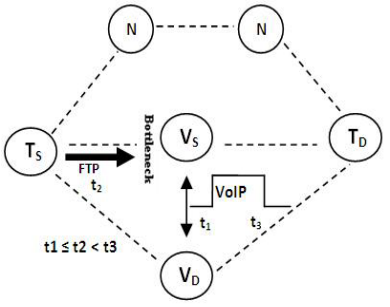


Fig. 2. Problem Description; Example Net work with AODV

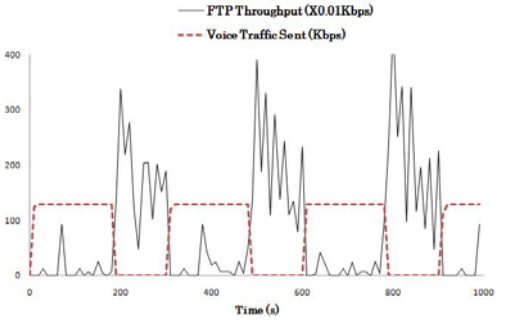


Fig. 3. FTP Throughput in presence VOIP traffic

destination D broadcasts a RREQ (route request) message to neighbouring nodes. In the simple case scenario where nodes 1, 2, 3 and 4 have a route to the destination, they reply with a RREP (route reply) message containing the number of hops (hop count hc) to the destination. AODV as a distance vector protocol that uses the hop count as the metric will choose the path through node 4 as it is the shortest.

3 TSLA Routing Protocol

The approaches discussed above converge in evaluating or assessing the level of activity in intermediate nodes by measuring either the load or the delay. However, none of the research reported has evaluated the effect that the type of service of the traffic carried by intermediate nodes has on the performance of routing protocols.

The problem can be clearly illustrated using the example scenario in Figure 2. Consider that the different nodes support IEEE 802.11e with the default parameters. At time t_1 a voice connection is opened between nodes V_S and V_D. While the connection is still active at time t_2, the node T_S generates FTP traffic destined for destination T_D. Based on AODV route selection criteria, the node T_S will establish a route to T_D through node V_S or node V_D as they provide the shortest path to T_D. T_S will therefore try to route the FTP traffic through the nodes that are already engaged in a VoIP conversation. With the limited bandwidth in MANETs, it is highly unlikely that T_S will have residual bandwidth to service the FTP connection as well. FTP traffic has a low priority compared to the voice traffic; it will queue at the source waiting for an opportunity to be transmitted. If the voice connection lasts for too long, this might results into buffer overflow and some packets might get dropped. Another important point to be taken into consideration is that in real life networks, high priority traffic conversations tend to last longer than lower priority traffic. Downloading a webpage or e-mail lasts just a few seconds, while voice calls and video streaming can last several minutes.

The graph in Figure 3 visualizes the throughput achievable by FTP in presence of voice traffic. It can be clearly seen that FTP throughput is very significantly decreased in presence of voice traffic, where its value drops to around one tenth of the expected

Tabel 1. Node congestion Level Classification

Load	ToS	Congestion Level (*n*)	Node Type	
0	0	0		Green
0	1	1		Yellow
1	0	2		Orange
1	1	3		Red

throughput. Similarly, if the node T_S had generated a high priority traffic, it will not get a chance to be transmitted through V_S or V_D as they are already busy with a similar priority traffic, and it is a know fact that nodes which are already transmitting tend to monopolise the channel.

In this paper, we suggest TSLA; a simple yet effective routing protocol to alleviate this problem and at the same time to achieve load balancing. TSLA is a cross-layer approach to enhance AODV by considering the effect of traffic ToS and coupling it with the existing congestion avoidance approach, which considers the load on intermediate nodes in the route selection process. The focus of TSLA is on the route discovery process. For a node wishing to transmit data, it broadcasts a RREQ like in AODV. On receiving a RREQ, a node checks its routing table for a route to the destination. In case it has a route to the destination and therefore wishes to generate a RREP, it first checks its congestion level. Using a similar colour scheme as in [7], nodes are classified into four categories; green, yellow, orange and red.

The load congestion level is determined based on the ratio between data currently buffered and the buffer size. This ratio can be adjusted dynamically, but in this study, a node with half buffer full is considered load congested. As for the ToS based congestion, nodes with best effort and background traffic are considered available whereas those with voice or video traffic are considered less available for new connections. The individual congestion levels are determined by the MAC layer of the node and given to the IP layer to determine the overall congestion level. The different possible combinations are shown in Table 1. A node with no traffic or with more than half buffer empty of delay-insensitive traffic is considered more open to receive more traffic, it is therefore labelled green. Whereas a node with more than half buffer full with delay-sensitive traffic is considered as a red node and therefore not available to accept new connections. A node with low load and delay-sensitive data is labelled yellow and therefore more open to accept traffic than a best effort traffic highly loaded node, which is labelled orange. This is because in QoS-aware networks, it is likely that the node with delay-sensitive traffic will get a chance to transmit it all before the node with best effort traffic. Moreover, it can be assumed that since delay-sensitive applications usually last longer, a half buffer is an indication that the communication is close to the end. Whereas a full buffer can indicate that, a communication is in process and likely to last for several minutes.

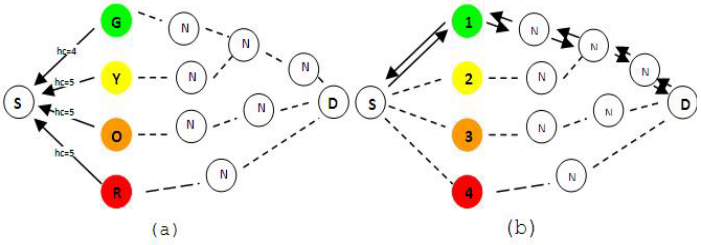


Fig. 4. TSLA Route Selection

After the intermediate node determines its congestion level, it generates a RREP packet. In this approach, we propose to modify the information contained in the RREP message so that it reflects the congestion status of the node. We therefore propose to add to the actual number of hops to the destination, additional hops proportional to the congestion level indicated by the node colour type. The resulting hop count included in the RREP is therefore given by

$$hc = actual_hc + (congestion_level \times n) \tag{2}$$

where n is a constant that can be varied depending on the network size, and therefore is proportional to the average hop count of the network. For small networks, a small value of n will be obtained. On receiving the RREPs, the destination will choose the route with the smallest number of hop count as in AODV.

Let us use the same example network of Figure 1 and consider that the one hop intermediate nodes congestion levels are labelled green to red as shown in Figure 4(a). As it is a small network, lets consider n=1. The intermediate nodes will therefore reply with RREP with the modified hop count values as shown in the figure. Applying the shortest path algorithm, TSLA will therefore choose to use the next hop as node 1 (Figure 4(b)), unlike AODV which chose the busiest node 4.

We mentioned earlier that TSLA is a cross layer solution that works in conjunction with the MAC layer. The MAC layer is therefore responsible for updating the IP layer whenever there is a change in either the traffic ToS or the buffer load. In our implementation, this was achieved by creating two interrupts at the MAC layer, one for the ToS and the other one for the load. The rising edge of the ToS interrupt notifies the IP layer that this node currently hold in its queues delay-sensitive traffic, while its falling edge indicates best effort or background traffic. Similarly, the rising edge of the load interrupt notifies the IP layer that the node is becoming overloaded and a falling edge indicates the node is lightly loaded. The two interrupts generated by the MAC layer are directly fed to the IP layer, which in turn will have to notify the MANET process. TSLA will handle them as explained above.

4 Performance Evaluation

In this section, we evaluate the performance of the TSLA routing protocol described in the previous section and we compare it to AODV. The performance of the two

routing protocols is assessed by analysing the network throughput, the packet end-to-end delay, the amount of traffic dropped and the routing load.

4.1 Simulation Setup and Parameters

Our simulations were conducted using OPNET Modeller 11.5. The simulations were run for 1000 seconds. We simulated a network consisting of 50 mobile nodes moving in a 1000x1000 m area and a nominal transmission range of 250m. The MAC layer protocol used is EDCA, the four standard access categories (ACs) are assigned priorities based on the default parameters for an IEEE802.11e physical layer. The network traffic consisted of long-lived FTP file transfers. The voice traffic was simulated by establishing G711 CBR connections between mobile nodes at some predefined time of the simulation. As the protocol (TSLA) proposed in this paper is a congestion avoidance routing protocol, we evaluated its performance compared to AODV's under different congestion levels. We also evaluated the two protocols under different mobility levels. The simulation results are averaged over five different seeds and the error bars represent 90% confidence intervals.

4.2 Simulation Results

4.2.1 Number of Sources

The performance of on-demand routing protocols is highly dependent on the number of nodes concurrently transmitting. TSLA is based on avoiding nodes highly loaded with high priority traffic; we therefore vary the number of voice traffic sources.

A. Throughput: First, keeping the number of FTP connections to 10, the voice connections were varied from 1 to 5. With the increase in number of VoIP nodes, FTP performance for the two routing protocols deteriorates as seen in Figure 5(a). We observe a decrease of 30% in FTP goodput when the number of voice connections is increased from 1 to 5. This is because more transmission opportunities are given to the nodes with delay-sensitive voice traffic, and less TCP traffic gets transmitted. The voice goodput on the other hand (Figure 5(b)) is increased when the number of voice connections increases. However, the increase in voice throughput is not a linear function of the number of voice connections. This is because self-contention exists between the voice connections themselves.

If we consider the difference in the performance of the two routing protocols, TSLA outperforms AODV in almost all the cases. With TSLA, new connections use only the least congested nodes and therefore the load is uniformly distributed across the network. More specifically, the best effort traffic avoids the nodes with voice traffic, therefore avoiding being dropped due to lack of transmission opportunities. It is also observed that it is not only best effort traffic that benefits from the TSLA load balancing approach. The voice traffic goodput is also higher in TSLA. Even though voice traffic is high priority, if a new voice connection has to be established while there is already an ongoing voice conversation, it is less likely to be transferred through the nodes that are already busy with voice traffic as they are of the same priority. It is better for the new connections to be routed through less busier paths even if they are longer, therefore avoiding same priority traffic crashes at some nodes.

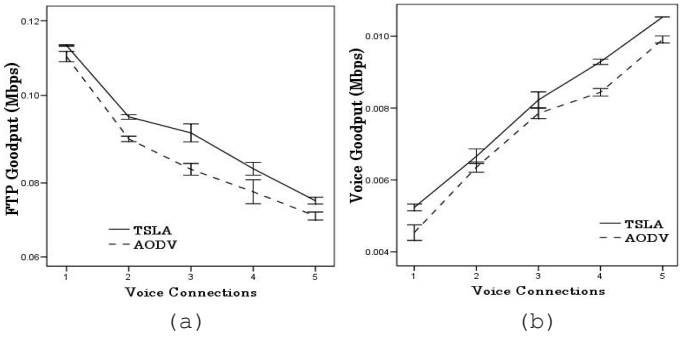


Fig. 5. Effect of Voice Connections on Goodput: (a) FTP, (b) Voice

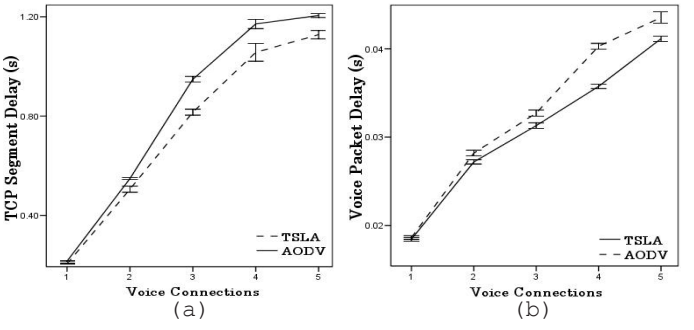


Fig. 6. Effect of Voice Connections on Packet Delay: (a) FTP, (b) Voice

B. Packet End-to-End Delay: The total delay experienced by any packet consists of queuing and propagation delay. The queuing delay of a specific packet will depend on the number of other, earlier-arriving packets that are queued and waiting for transmission across the link. In MANETs, the queuing delay also depends on the medium contention from neighbouring nodes as the medium access is through distributed mechanisms. The propagation delay on the other hand depends on the speed of the medium and the length of the path.

Most of the load balancing or load aware routing protocols developed for MANETs have been reported to achieve better delay than normal routing protocols. Similarly, TSLA is able to constantly deliver both TCP and voice traffic with delays smaller than AODV's. As expected, the end-to-end delay increase is observed in all the cases when the number of traffic sources increases (Figure 6(a) and 6(b)). This is a result of increased medium contention.

FTP traffic is routed avoiding nodes busy with delay sensitive voice traffic, but if they encounter best effort traffic on the chosen route, they might face a little bit of waiting since the priority is the same. Nevertheless, the waiting is shorter than being routed through nodes with higher priority traffic, hence the decrease in the TCP segment delay (Figure 6(a)). Similarly, TSLA voice delays are smaller than AODV in all the cases as seen in Figure 6(b). Another important point to note in dealing with the

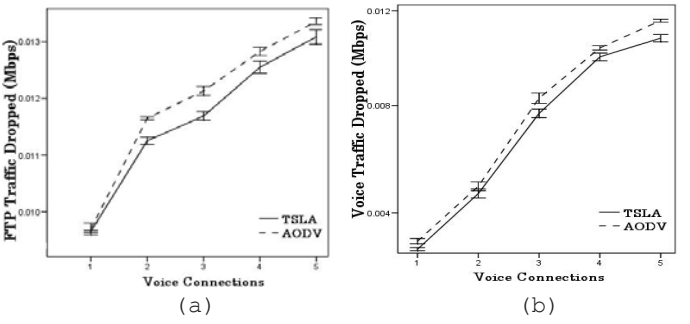


Fig. 7. Effect of Voice Connections on Traffic Dropped: (a)FTP, (b)Voice

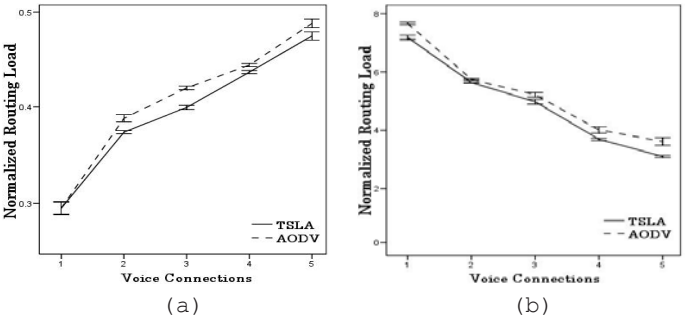


Fig. 8. Effect of Voice Connections on Routing Load Normalized on: (a)FTP, (b)Voice

end-to-end delay is that since TSLA packets are routed through longer paths, they would be expected to have higher propagation delay. However, as the chosen paths are the least congested, it is less likely that the packets will face long propagation delay. Moreover, since the queuing delay is much reduced, the overall packet (segment) delay is reduced.

C. Traffic Dropped: In 802.11e, a queue is held for each access category. The rate at which packets arrive at the MAC layer might exceed the rate at which they are transmitted. This is common in wireless networks due to the fierce way in which the stations contend for the medium. This would results into overflow of the buffer used to store the ACs data awaiting transmission in which case some of them might be dropped by the MAC layer itself. Moreover, in wireless networks, when the MAC ACK is not received, the source station retransmits the same frame repeatedly until the MAC ACK is received or it exceeds the limit of transmissions attempts allowed per frame.

The combined traffic dropped by the MAC layer for the two types of traffic when the number of sources is varied is shown in Figure 7. All the graphs show an increase in data traffic dropped for the two routing protocols due to increased congestion. Packet drops is one sign of congestions in any network. Load aware routing protocols

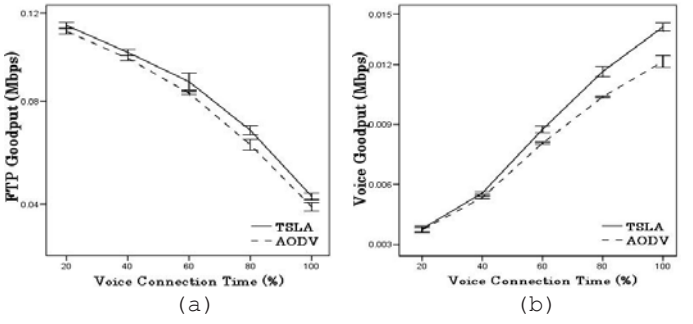


Fig. 9. Effect of Voice Connection Time on Goodput: (a)FTP, (b)Voice

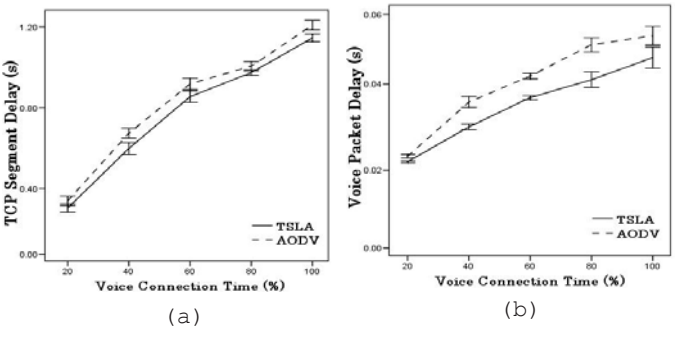


Fig. 10. Effect of Voice Connection Time on Packet Delay: (a)FTP, (b)Voice

are designed to avoid network congestion, thus reducing packets drop. TSLA is no exception, and it is able to achieve smaller numbers of packets drop compared to AODV. Both FTP and voice suffer from increase in the number of voice connections which results into more packets drops.

D. Routing Overhead: High routing load usually has a significant performance impact especially in low bandwidth wireless links. It is therefore important to evaluate how much routing load is produced by a reactive protocol. The routing load produced by reactive protocols is proportional to the generated data. For AODV and TSLA who have the same route discovery process, they would generate similar amount of routing overhead in the same network setting. They however are able to deliver different amount of data traffic. This paper evaluates the normalized routing load, which is the ratio between routing traffic generated to the successfully received data traffic.

The graphs shown in Figure 8(a) and 8(b) are the variation of normalised routing on FTP and voice goodput respectively. With increasing the number of voice connections, the FTP traffic generated still stays the same; therefore, the routing load produced stays similar as well. Nevertheless, as the FTP traffic received drops when the number of voice connections increases, the normalized routing load steadily increases as seen in Figure 8(a). As TSLA delivers more FTP traffic than AODV does (Figure 5(a)), its normalized routing load is consequently smaller than AODV's. The routing

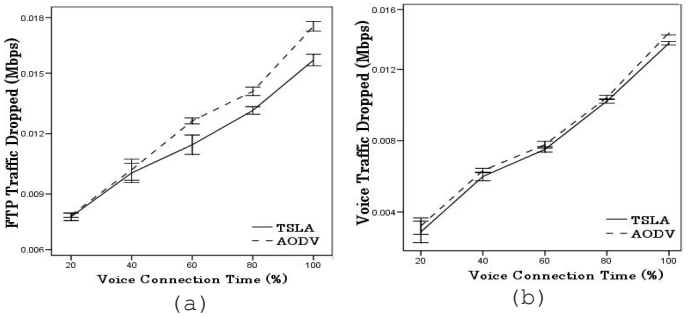


Fig. 11. Effect of Voice Connection Time on Traffic Dropped: (a)FTP, (b)Voice

load normalized on voice traffic decreases as the voice goodput increases as shown on Figure 8(b), and TSLA shows the lowest as it achieves better goodput than AODV.

4.2.2 Voice Connection Time

We mentioned earlier that in today's networks, multimedia traffic connections tend to last longer than best effort traffic. We therefore evaluate the effect of voice connection time on the performance of the two routing protocols. The voice connection time was varied as a percentage of the total simulation time.

The goodput achieved by FTP connections is very much affected by the voice connection time (Figure 9(a). As best effort opportunities to be transmitted when voice transfer is taking place are almost none, the longer the voice connections last, the poor the FTP goodput becomes. Although an increased number of voice connections were proven harmful to FTP goodput (Figure 5(a)), it is clear in this section that long-lived voice connections have the worst effect on FTP, whose goodput drops below 25% when there is constant voice traffic transfer.

Using TSLA as the routing protocol, the route selection tries to bypass those nodes with voice traffic. TSLA is therefore able to achieve a remarkable improvement of 30% in FTP goodput in case of long-lasting voice connections. As for the voice traffic, TSLA is also able to deliver a large amount compared to AODV (Figure 9(b)). In TSLA networks, new connections voice traffic is routed through less loaded nodes, or through nodes with best effort traffic, in which case they will be able to get through immediately. Whereas in AODV, if the voice traffic is routed through a node already with voice traffic, there will be self-contention and some might be dropped.

The packet delay behaviour mirrors that of the goodput. AODV and TSLA TCP segment delays are similar for short time connections, but as the voice connection time increases, the improvement in TSLA segment delay becomes remarkable as seen in Figure 10(a). As for the voice packet delay (Figure 10(b)), TLSA is constantly achieving better values than AODV (improved by 50%). The reason it is bigger in voice delay is that, for any packets which bypass red nodes, they are automatically transferred ahead any existing best effort traffic as they have higher priority, whereas the best effort packets will have to wait their turn for transmission in a FIFO fashion.

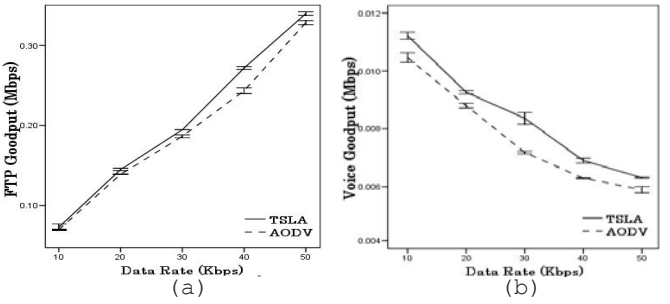


Fig. 12. Effect of Date rate on Goodput: (a)FTP, (b)Voice

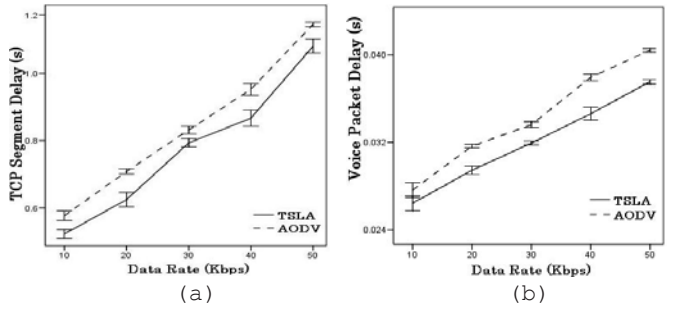


Fig. 13. Effect of Data Rate on Packet Delay: (a)FTP, (b)Voice

The packet dropped metric has characteristics similar to those of throughput and delay. The longer the voice connections last, the large the amount of traffic dropped (both FTP and voice as seen in Figures 11(a) and 11(b) respectively). For FTP traffic, time slots during which transmission is possible are reduced, the queues build up and more traffic end up being dropped. As for voice traffic, traffic dropped over long periods is logically bound to be more than shorter ones. TSLA networks drop fewer packets than AODV as congested nodes are bypassed, and only nodes less likely to drop packets are used on the primary path.

4.2.3 Data Rate

In TSLA implementation, the best effort traffic is not meant to affect significantly the performance of the routing protocol unless its load is high. We evaluated how the two protocols behave under different FTP load. In these scenarios, the FTP traffic rate is varied, and the voice traffic is generated in 60% of the total simulation time. As seen on Figure 12(a), there is an increase in FTP goodput, and TSLA is able to deliver successfully more packets than AODV. The difference is more significant at higher loads. This is because in TSLA, following the nodes highly loaded with voice traffic; the nodes highly loaded with best effort traffic are the next one to be avoided. This Similar to the previous cases, the increase in FTP data generation rate results into increase in network contention hence the deterioration of voice performance as seen on Figure 12(b). TSLA shows a better performance than AODV in all the cases.

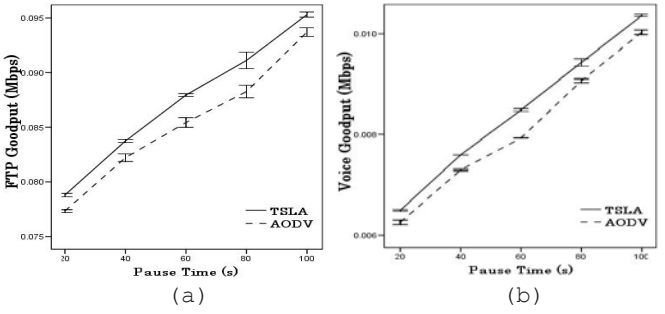


Fig. 14. Effect of Mobility on Goodput: (a)FTP, (b)Voice

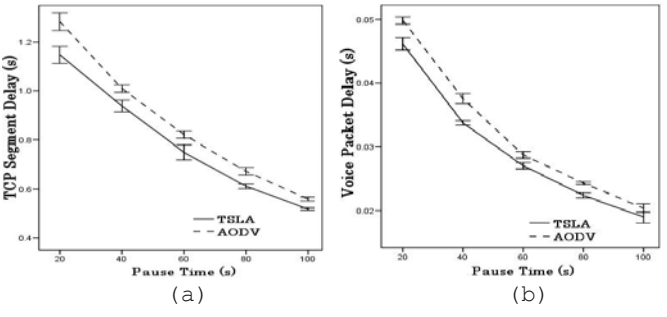


Fig. 15. Effect of Mobility on Packet Delay: (a)FTP, (b)Voice

Similarly, the packet delay for the two types of traffic is increased with FTP load. With congested networks, packets take longer to reach the destination and the queuing-delay is longer as well. A considerable improvement is observed in TSLA networks, for both FTP and voice traffic delay as shown on Figure 13(a) and 13(b) respectively.

4.2.4 Mobility

In MANETs, nodes mobility plays a significant role in determining the performance of routing protocols. We therefore evaluated how the two routing protocols perform under different mobility levels. The nodes were set to move following a random way-point mobility model, with an average speed of 10 m/s. Different mobility models were obtained by varying the pause time from 0 to 100 seconds.

The best effort traffic throughput and voice throughput are shown on Figure 14(a) and 14(b) respectively. We observe an improvement in network throughput in low mobility scenarios. This is because with large values of pause time, the routes are broken less frequently, the packet loss is reduced, hence the increase in the network throughput. The two protocols behave similarly, the difference in favour of TSLA being due to its use of congestion avoidance. Similarly, the packet end-to-end delay is reduced for the two protocols. As the route breakages are less frequent in low mobility scenarios, shorter time is spent in discovering and repairing routes, hence the decrease in TCP segment delay and voice packet delay as seen on Figure 15(a) and 15(b).

5 Conclusion

During the route discovery process in ordinary routing protocols in MANETs, nodes advertise themselves as capable of reaching the destination irrespective of the type of service and the load of the traffic in their queues. The new arriving traffic might therefore face long delay or get dropped failing to get transmitted ahead of existing high priority traffic. The adverse effect of this issue has been investigated in this paper.

As such incidents are common in QoS-aware MANETs that are concerned with QoS guarantees for delay sensitive applications, we then propose a new Type of Service and Load Aware (TSLA) routing protocol which avoids such nodes in the route discovery process. TSLA is a cross-layer congestion-avoidance routing protocol in which the routes through nodes engaged with large amount of delay-sensitive traffic for extended periods are only selected as the last resort, even when they are shorter. Avoiding intermediate nodes heavily occupied with high priority traffic can potentially alleviate congestion resulting in less packets drop and incurring shorter end-to-end delay. Our heavy simulation study has confirmed the advantages of TSLA over AODV in QoS-aware MANETs. Although TSLA has been implemented as an enhancement to AODV, the idea is applicable to any other reactive routing protocol; a scenario for our future study.

References

[1] Boppana, R., Dyer, T.D.: A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks. Mobihoc (October 2001)
[2] Clausen, T., Jacket, P., Viennot, L.: Comparative study of Routing Protocols for Mobile Ad Hoc Networks. In: The First Annual Mediterranean Ad Hoc Networking Workshop (September 2002)
[3] Gao, X., Zhang, X., Shi, D., Zou, F., Zhu, W.: Contention and Queue-Aware Routing Protocol for Mobile Ad Hoc Networks. In: WiCOM (2007)
[4] Lee, S.-J., Gerla, M.: Dynamic load-aware routing in ad hoc networks. In: IEEE ICC (October 2001)
[5] Li, Y., Man, H.: Three load metrics for routing in ad hoc networks. In: VCT (September 2004)
[6] Saigal, V., Nayak, A.K., Pradhan, S.K., Mall, R.: Load balanced routing in mobile ad hoc networks. Computer Communications 27, 295–305 (2004)
[7] Tran, D.A., Raghavendra, H.: Congestion Adaptive Routing in Mobile Ad Hoc Networks. IEEE Transactions on Parallel Distributed Systems 17(11), 1294–1305 (2006)
[8] Ye, Z., Krishnamurthy, S.V., Tripathi, S.K.: Use of Congestion-Aware Routing to Spatially Separate TCP Connections in Wireless Ad hoc Networks. In: MASS (2004)
[9] Yuan, Y., Chen, H., Jia, M.: An Adaptive Load-balancing Approach for Ad Hoc Networks. In: WCNM (2005)
[10] Zhang, X., Gao, X., Shi, D., Sung, D.K.: Lifetime-aware Leisure Degree Adaptive Routing Protocol for Mobile Ad hoc Networks. In: ICWMC (2007)
[11] Zheng, X., Guo, W., Liu, R., Tian, Y.: A New Dynamic Load-aware Based Load-balanced Routing for Ad Hoc Networks. In: ICCCAS (2004)
[12] Zou, F., Zhang, X., Gao, X., Shi, D., Wang, E.: Load Balance Routing Using Packet Success Rate for Mobile Ad hoc Networks. In: WiCOM (2007)

Query Dissemination with Predictable Reachability and Energy Usage in Sensor Networks

Zinaida Benenson[1,⋆], Markus Bestehorn[2], Erik Buchmann[2], Felix C. Freiling[1], and Marek Jawurek[3]

[1] University of Mannheim, Germany
[2] University of Karlsruhe, Germany
[3] Fraunhofer IESE, Germany

Abstract. Energy-efficient query dissemination plays an important role for the lifetime of sensor networks. In this work, we consider probabilistic flooding for query dissemination and develop an analytical framework which enables the base station to predict the energy consumed and the nodes reached according to the rebroadcast probability. Furthermore, we devise a topology discovery protocol that collects the structural information required for the framework. Our analysis shows that the energy savings exceed the energy spent to obtain the required information after a small number of query disseminations in realistic settings. We verified our results both with simulations and experiments using the SUN Spot nodes.

1 Introduction

Wireless sensor networks have been established in many important application areas from ambient intelligence over scientific research to industrial uses. Such sensor networks usually consist of numerous battery-powered nodes [3,2] equipped with sensing devices, low-power wireless communication and limited computational resources. In order to fulfill complex measurement tasks, the sensor-nodes use self-organization techniques to form ad-hoc networks where the nodes (1) forward queries from a central base station, (2) measure sensor values, (3) do in-network query processing and (4) return the results to the base station. In this paper, we focus on the query dissemination phase, i.e., the first step of query processing in sensor networks.

One of the most important optimization goals in sensor networks is to maximize their lifetime by minimizing the energy spent for communication. However, saving communication effort obviously may have a negative effect on quality-of-service parameters of the query. For example, if energy is saved by querying only 50% of the nodes, the accuracy of the query degrades. How much it degrades depends on many factors and is not very well understood. Quantifying this tradeoff

⋆ Zinaida Benenson was supported by Landesstiftung Baden Württemberg as part of Project "ZeuS" and by the Schlieben-Lange scholarship.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 279–292, 2008.
© Springer-Verlag Berlin Heidelberg 2008

between communication strategy and service quality for query dissemination is the topic of this paper.

Related Work. While numerous sophisticated in-network query processing techniques have been developed [11,12,18,19], they mostly focus on operator processing, optimization and aggregation techniques. The dissemination of the query from the base station into the network has either been disregarded or is done via simple flooding [8,14]. It is well known that flooding wastes energy. For example, analyses [13] have shown that a rebroadcast increases the area where the message is received by 61% at most, dropping to $\approx$ 20% for average networks. Therefore, most of the rebroadcasts will not result in additional nodes receiving the query. Furthermore, most nodes receive the query more than once, which results in additional energy consumption because receiving messages also consumes energy.

To avoid the disadvantages of simple flooding, several mechanisms for broadcasting in wireless networks have been proposed (see [17] for an overview). Generally, these approaches try to control which nodes rebroadcast a message in order to keep the number of nodes that receive the query more than once as small as possible. For example, in *counter-based flooding* schemes [13,17], if a node hears k or more of its neighbors rebroadcast the message, it suppresses its own transmission. In *neighbor knowledge* broadcasting schemes [10,15], nodes use local topology information to determine which nodes must rebroadcast a message. The advantage of these approaches is that the overlap of recipients can be reduced in a controlled manner, but this comes at the significant cost of acquiring and updating the neighborhood information. Furthermore, [16] has shown that finding a minimal set of rebroadcasting nodes can be reduced to the Dominating Set Problem, which is NP-complete [6].

A very promising approach are *probabilistic* or *epidemic* broadcast algorithms [13,5] where every node forwards a message with a predefined probability p. Compared to schemes using neighborhood knowledge, these methods do not induce the overhead of acquiring, storing and updating neighborhood knowledge. However, these schemes require information about the network in order to determine an optimal p. If p is set too high, the disadvantages of simple flooding arise, and if p is too low, the probability that all nodes receive the broadcast message decreases. In this paper we will focus on probabilistic flooding.

Contributions. In this paper, we study query dissemination techniques that can be seen as a combination between neighbor knowledge broadcasting and probabilistic flooding. Using extensive simulations we explore the tradeoff between energy, reachability and structural information required. We show that using very moderate structural information on the network it is possible to predict the number of nodes reached according to a certain broadcast probability p. Furthermore, the number of transmissions can be estimated in advance.

In particular, we make the following contributions:

1. We introduce an analytical framework to estimate the reachability and the number of transmissions in dependence to the rebroadcast probability p. Our

framework bases on connectivity information and a histogram containing the number of nodes reached with each rebroadcast, starting at the base station.
2. We describe a lightweight distributed topology discovery protocol which obtains the required information. Our analysis shows that gathering structural information and computing an optimal p saves energy after a small number of probabilistic floodings in realistic settings.
3. We conducted simulations with up to 425 nodes to verify the results of our framework for large numbers of nodes. Furthermore, we tested our findings on a testbed consisting of 17 Sun Spot sensor nodes.

Outline. In Section 2 we present a framework which estimates the number of nodes reached and energy spent by probabilistic flooding for a particular rebroadcast probability p. The framework depends on topological information. In Section 3 we show how to gather the required information efficiently. In Section 4 we present simulation and experimental results, and we conclude in Section 5.

2 Reachability and Energy Consumption Prediction for Query Dissemination

In this work we focus on probabilistic flooding where each node rebroadcasts queries with a fixed probability p. Parameter p allows to fine-tune the tradeoff between energy spent for query dissemination and the number of nodes reached. Moreover, in most (densely connected) sensor networks there exists a $p_0 < 1$ such that all nodes are reached by the base station. Thus, if the rebroadcast probability p is larger than p_0, more queries are rebroadcast than necessary, and the query dissemination can save energy by using p_0. On the other hand, if $p < p_0$, the query reaches only a fraction of nodes. This can be useful to trade energy with result quality.

Our goal is to develop a framework to predict for every p the number of reached nodes R and the energy E consumed by the query dissemination process. Knowing the dependencies between p, R and E allows the base station to estimate how many nodes can be reached using a fixed amount of energy, or at which p the reachability cannot be improved any more (at least, for reasonable energy cost). Obviously, energy usage prediction depends on reachability prediction, which in turn depends on the network topology. The more the base station knows about network topology, the more precise prediction can be made. On the other hand, gathering information about network topology consumes energy. Thus, we are interested in making predictions using a set of topological information which can be obtained without exhausting potential energy savings due to deriving an optimal p.

In the following we will present our framework for predicting reachability $R(p)$ and energy consumption $E(p)$ according to given topological information and a rebroadcast probability p. More specifically, $R(p)$ estimates the number of nodes reached, and $E(p)$ provides an estimate for the number of sent and received messages, which is proportional to the energy consumed.

2.1 Assumptions and Notations

Our estimation of the reachability bases on two assumptions:

- The sensor network is in a stable state while flooding the query, i.e., the number of nodes in each hop set does not change significantly between obtaining topology information and flooding.
- A node is either reached by a node that is one hop closer to the base station, or has the same hop distance to the base station.

A flooding disperses through a topology in multiple steps, beginning at the base station. The nodes which receive the query directly from the base station (1 hop) rebroadcast it, so that the query reaches the nodes two hops away from the base station in the next step. The procedure recurs until each node has forwarded the message once.

If a node A receives a previously unknown flooding message from a node B, we say that A is reached by B in this particular *flooding instance*. In addition, we will denote all nodes reached with h hops as *hop set* $H[h]$.

2.2 Topological Information

Our analytical framework depends on the following topological information (Section 3 will introduce a protocol that collects it efficiently):

- $histogram[h]$: stores the number of nodes reached at each hop from the base station, i.e., $\forall i \in \{1 \cdots n\} : histogram[i] = |H[i]|$.
- $connectivity[h]$ stores the average number of connections from one node in hop set $H[h]$ to a node from $H[h-1]$.
- $interconnectivity[h]$ stores the average number of connections between the nodes from the same hop set, i.e., the connections a node in $H[h]$ has to another node in $H[h]$.

Figure 1 illustrates this with an example. In this figure the hop set $H[i]$ consists of 3 nodes, the previous hop set $H[i-1]$ consists of 2 nodes. Edges connect the nodes that can hear each other's broadcast. Figure 2 shows the histogram and (inter-)connectivity for the example in Figure 1.

2.3 Reachability Prediction

Let $R_{direct}(h,p)$ be the number of nodes in hop set h which received their flooding message directly from a node in the hop set $H[h-1]$, and let $R_{indirect}(h,p)$ denote the number of nodes which received the flooded message from a node in the same hop set $H[h]$. Then the number of nodes reached at the h-th hop for a specific rebroadcast probability p can be computed as follows:

$$R(h,p) = min(R_{direct}(h,p) + R_{indirect}(h,p), histogram[h]) \tag{1}$$

The total reachability for some p is the sum over all hops:

$$R(p) = \sum^{h} R(h,p) \tag{2}$$

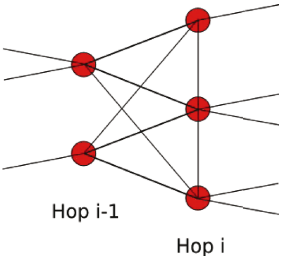


Fig. 1. Example for hop sets and their (Inter-)Connectivity

...	i − 1	i	...
...	2	3	...

(a) Histogram

...	i − 1	i	...
...	1.5	2	...

(b) Connectivity

...	i − 1	i	...
...	0	4/ 3	...

(c) Interconnectivity

Fig. 2. Histogram, Connectivity and Interconnectivity in Figure 1

Note that $R_{direct}(h,p) + R_{indirect}(h,p)$ can be larger than the actual number of nodes in the hop set $H[h]$, because rebroadcast messages can be received from nodes which might have received the message before. Thus the minimum function ensures that at most the actual number of nodes in the hop set is returned. $R_{direct}(h,p)$ can be computed recursively: $histogram[h-1]$ nodes could forward a message directly to a node in $H[h]$, but only $k = p * R(h-1,p)$ of $histogram[h-1]$ nodes have received the message in the previous step.

Let $P(\text{event})$ denote the probability for a certain event. Now we need the probability for the event "A node from hop set $H[h]$ receives its message from a node from hop set $H[h-1]$" The probability for this event is:

$$P(\text{reached directly}) = 1 - P(\text{not reached directly}) \tag{3}$$

The counter-event "not reached directly" can be obtained by considering the nodes which did not receive the message in the previous step. Thus, the problem corresponds to an urn model where k black and $n-k$ red balls are placed in an urn, and $P(\text{not reached directly})$ means to draw red balls only. Let $l = connectivity[h]$ be the number of connections a node in $H[h]$ has to the previous hop set $H[h-1]$ on average. The probability $P(\text{not reached directly})$ can be computed as follows:

$$P(\text{not reached directly}) = \prod_{l=0}^{\lceil connectivity[h] \rceil - 1} \frac{n-l-k}{n-l} \tag{4}$$

After having obtained this probability, we can calculate the number of nodes from hop set $H[h]$ receiving the flooding directly by multiplying the probability for the opposite case with the number of nodes in the hop set:

$$R_{direct}(h,p) = P(\text{reached directly}) * histogram[h] \tag{5}$$

The remaining nodes in hop set $H[h]$ can still be reached indirectly, i.e., by a subsequent broadcast by nodes from the same hop set. To calculate the number of nodes reached indirectly, we assume that the nodes which received the message are equally distributed over the hop set, i.e., if k from n nodes are directly reached, each node in the hop set obtained the message with probability $\frac{k}{n}$. Our experimental evaluation will show that this simplification is legitimate, i.e, it is not necessary to collect topological information in more detail. We calculate the number of neighbors of a node which directly received the flooding message and then rebroadcast it as:

$$n_{dr} = P(\text{reached directly}) * interconnectivity[h] * p \tag{6}$$

Finally, we estimate the number of nodes which received the flooding message indirectly:

$$R_{indirect}(h,p) = n_{dr} * histogram[h]. \tag{7}$$

2.4 Energy Consumption Prediction

After having estimated the number of nodes reached, we will estimate the energy required by probabilistic flooding. Therefore, we distinguish between sent and received messages. The number of messages sent in hop set $H[h]$ is as follows:

$$msgs_{sent}(h,p) = R(h,p) * p \tag{8}$$

Next, we estimate the number of messages received from the nodes of the previous hop:

$$Rec_1(h,p) = R(h-1,p) * p * \frac{connectivity[h] * histogram[h]}{histogram[h-1]} \tag{9}$$

$\frac{connectivity[h]*histogram[h]}{histogram[h-1]}$ calculates the average number of outgoing links from hop set $H[h-1]$ to $H[h]$. The number of all "receive" events induced at nodes of the hop set $H[h]$ and hop set $H[h-1]$ by the rebroadcast of reached nodes of hop set $H[h]$ can be calculated as follows:

$$Rec_2(h,p) = R(h,p) * p * (connectivity[h] + interconnectivity[h]) \tag{10}$$

Finally, the total number of received messages can be estimated as

$$msgs_{received}(h,p) = Rec_1(h,p) + Rec_2(h,p) \tag{11}$$

The total energy cost of the probabilistic flooding is calculated by vector multiplication of the tuple of sent and received messages with the vector of energy costs for sending and receiving and adding them up for every hop set:

$$E(p) = \sum_h (msgs_{sent}, msgs_{received})_{(h,p)} * \begin{pmatrix} energyPerSend \\ energyPerReceive \end{pmatrix} \tag{12}$$

3 Topology Discovery Protocol

We now describe the light-weight topology discovery protocol used in our experiments. It is an adaption of the well-known *echo algorithm* by Chang [4], i.e., it is structured in two waves: The first *expansion* wave of messages is flooded from the base station and is used to explore the network. When this waves reaches the borders of the network, a second *contraction* wave flows back to the base station, aggregating topology information / histograms on its way. The prediction formulas presented in Section 2 use these histograms to determine the parameter p for probabilistic flooding. Due to space limitations, we only present the general idea of the protocol here.

The base station initiates the topology discovery by broadcasting a *Topology Discovery Message* (`TDReq`), thus starting the expansion wave.

Expansion Wave. When a node receives a `TDReq` for the first time, the receiver must accomplish 4 steps:

1. Create an empty histogram data structure as described in Section 2.2 and mark the sender of the `TDReq` as its parent node. The receiver also extracts the hop number from the `TDReq` and stores it.
2. Start a timeout to ensure that the receiver does not wait forever for potential children.
3. Broadcast own request message with the receiver as sender, an incremented hop number, and parent id of the receiver.
4. Wait until the afore mentioned timeout expires. Note that the timeout should be sufficiently long to allow the children of the node to receive, process and rebroadcast their own `TDReq` messages. When the timeout expires, the contraction phase starts.

If a `TDReq` is received, then it could have three different originators. It could either be (1) a sibling of the node's parent, (2) a sibling of the node itself, or (3) a node in the subsequent hop set. Note that all three cases can be distinguished from the information contained in the `TDReq`. For example, in case (3) the request will contain the id of the receiver node. Depending on the case, the connectivity or inter-connectivity value in the histogram data structure is modified.

Contraction Wave. While a node waits for the timeout to expire, all incoming *Topology Discovery Responses* are recorded into the histogram data structure. On leaf nodes, the timeout expires without any incoming response messages, thus leaf nodes create response messages containing their hop number and appropriate values for connectivity and inter-connectivity[1]. Every leaf node sends such a response message to its parent and thereby starts the contraction wave.

[1] The average values for connectivity / inter-connectivity are stored as tuples to allow aggregation: The first value contains the sum of connections and the second stands for the number of nodes which have aggregated these connections. This allows the aggregation at every node and avoids floating point numbers in messages.

In case the node has children, the histogram lists of every response are aggregated in a way that the position i of the resulting list contains the sum of the histograms of the children. These aggregates histograms are always forwarded to the parent node and eventually reach the base station. Based on these values, the base station is able to predict reachability and energy consumption as described in Section 2.

Energy Cost and Message Size of Topology Discovery Protocol. As every node only broadcasts one Topology Discovery Request and only sends one Topology Discovery Response, the energy costs per node can be estimated as follows:

$$E_{Node} = E_{send}(b_1) + E_{send}(b_2) + AverageNodeDegree * (E_{rcv}(b_1) + E_{rcv}(b_2)) \quad (13)$$

The value b_1 stands for the number of bytes in the Topology Discovery Request of the node, b_2 stands for the number of bytes in the Topology Discovery Response.

Later we calculate energy consumption of Topology Discovery Protocol for a particular scenario and show after how many probabilistically flooded queries the protocol pays off.

4 Evaluation

In this section we evaluate the prediction framework with different node setups using simulations and a deployment of 17 Sun SPOT sensor nodes [2] in our faculty building. We compare the predictions made by our framework with the flooding of queries in simulated networks of up to 425 nodes and in the real sensor network, showing the following:

1. For all simulated networks and the real sensor network, the accuracy of the reachability prediction based on the topology information is sufficiently high.
2. Any inaccuracy related to the probabilistic flooding is clearly outweighed by the amount of energy saved through decreased communication overhead.

Our framework produces stochastic results for the average case, i.e., it works well for sufficiently dense networks or for large numbers of trials.Thus, we expect a deviation between the predicted values and experimental results. Nevertheless, our predictions can be successfully used for query optimization purposes, lifetime estimation or the computation of the rebroadcast probability with a small additional safety margin.

4.1 Simulations

For the simulation we used a custom *Karlsruhe Sensor Networking Simulator* which is interface-compatible to Sun SPOT sensor nodes, thus enabling us to deploy the prediction framework as well as the topology discovery protocol in both the simulated environment and the real deployment.

Simulations Setup. We considered the following simulation scenarios: uniform and Gaussian distributed nodes, a scale-free distributed scenario, and a real world set up from the Intel Lab Website [1]. Due to space limit we only present the results for the first two scenarios below.

Uniform node distribution. All topologies of this scenario distribute the sensor nodes uniformly in a circular area around the centre where the base station is located. The parameter of this scenario is the average number of neighbors of every node. The radius of the simulation area is fixed, and the number of nodes is adjusted accordingly to obtain the respective average node degree. We used networks of node degrees 4, 8, 12 and 16, and generated for each node degree 40 different topologies. For each topology we ran 100 experiments.

Table 1. Average node degree in Uniform scenario and resulting amount of nodes

Average Node Degree	Used Sensors
4	125
8	225
12	325
16	425

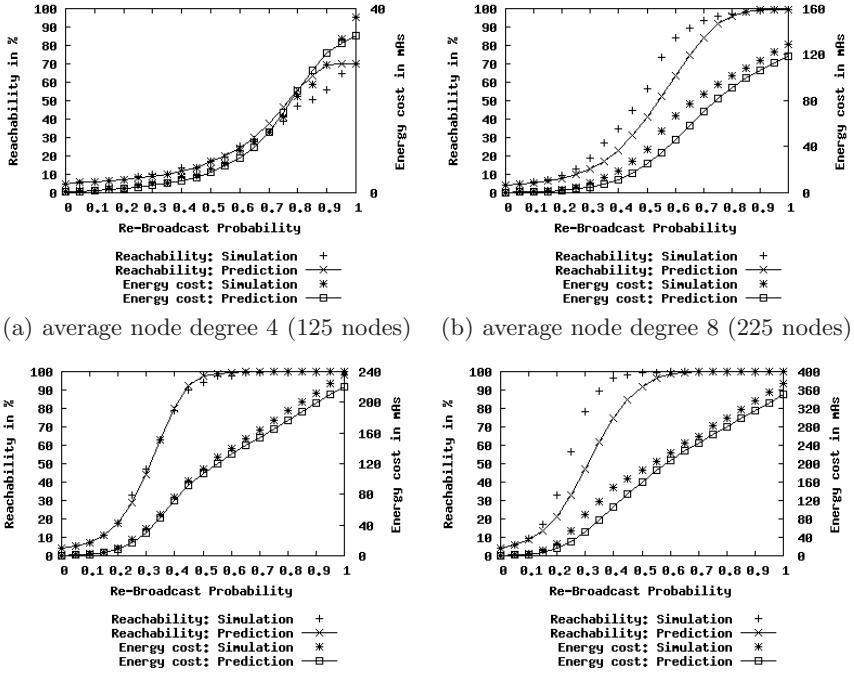
Gaussian node distribution. In this scenario all sensor nodes are distributed using a Gaussian distribution over an area with a fixed radius of 30 units. The coordinates of the nodes are taken from a Gaussian sampling with the centre of the environment as mean and a standard deviation of 18 units. By choosing this standard deviation most of the sensors are placed in the target area, only few nodes were placed beyond. Most nodes are located close to the centre, and the further away from the base station the lower the node density. This scenario has the number of sensor nodes placed as parameter. In order to compare the results from the uniform scenario with this scenario, we generated instances with the same average node degrees for scenarios with the same number of nodes (see Table 1). As in uniform scenario, for each of the four network sizes we generated 40 topologies and run 100 experiments per topology.

Reachability and Energy Consumption. For this series of experiments, we assume a message payload size of 28 bytes for the query. According to an analysis [9] of MICAz [3] sensor nodes the energy consumption Formulae 14 and 15 were determined. Parameter b specifies the number of bytes sent/received.

$$EnergyForSending(b) = 0.185191mAs + (b - 28byte) * 2.48461mAs * 10^{-5} \quad (14)$$

$$EnergyForReceiving(b) = 0.042mAs + (b - 28byte) * 2.47915mAs * 10^{-5} \quad (15)$$

The energy consumption was firstly measured for standard TinyOS [7] message payload of 28 bytes, and then the energy consumption for sending (receiving) b additional bytes was determined. The results of evaluation of our



(a) average node degree 4 (125 nodes) (b) average node degree 8 (225 nodes)

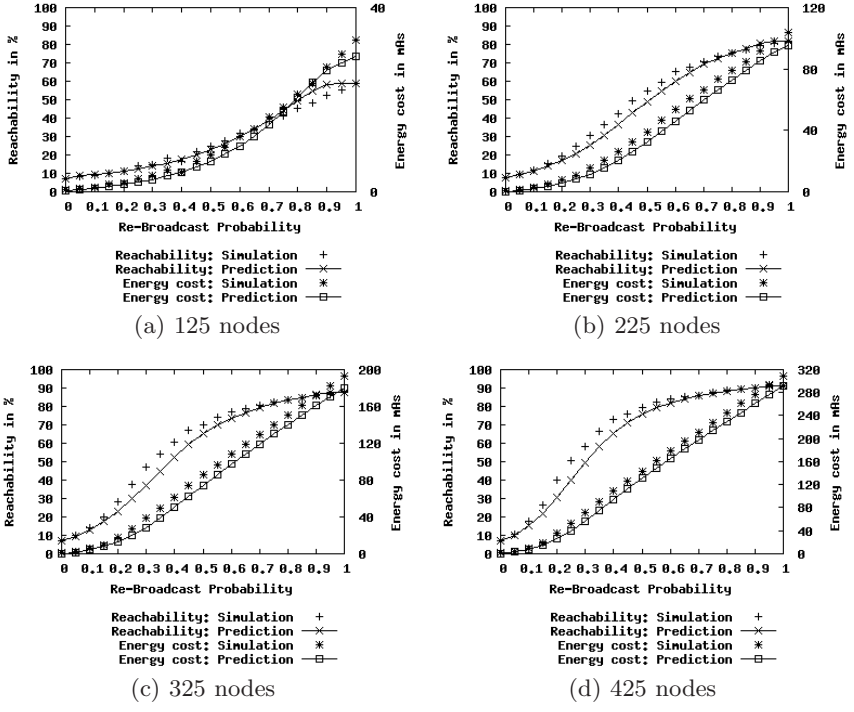
(c) average node degree 12 (325 nodes) (d) average node degree 16 (425 nodes)

Fig. 3. Comparison of simulated reachability/energy cost in uniform scenarios

reachability and energy consumption prediction framework are presented in Figure 3 for the uniform scenario, and in Figure 4 for the Gaussian scenario.

One can see that our framework works reasonably well in sufficiently dense scenarios. It systematically underestimates reachability and energy consumption, but it still allows to save a large amount of energy. For example, in Figure 3(b–d), although the full reachability is achieved with smaller rebroadcast probabilities than predicted, flooding with the predicted probability still allows to save from 10 (b) to 37 (d) percent of energy. Moreover, reachability and energy consumption predictions for the Gaussian scenario follow the simulated results so closely that they allow very accurate determination of the rebroadcast probability needed to reach a particular amount of nodes. Note that in Gaussian scenarios, some nodes are placed so far from the base station that the network becomes disconnected.

Topology Discovery and Reachability Prediction Payoff. Assuming a uniform scenario with 425 nodes, average node degree 16 and a reachability of about 99%, up to 150 mAs can be saved using our prediction framework (see Figure 3(d)). Using rebroadcast probability $p = 0.6$ only approximately $220mAs$ are consumed in comparison to the simple flooding which consumes



(a) 125 nodes (b) 225 nodes

(c) 325 nodes (d) 425 nodes

Fig. 4. Comparison of simulated reachability/energy cost in Gauss scenarios

$370mAs$. However, some energy was previously spent for Topology Discovery Protocol. Using Formula 13 for energy consumption of the Topology Discovery Protocol, and Formulas 14 and 15 for energy consumption of MICAz nodes, we estimated that in the above scenario, the Topology Discovery Protocol has approximate costs of $722mAs$ (we omit the computations due to space limit). Thus, the Topology Discovery Protocol would have paid off after 5 probabilistic query floodings.

4.2 Sun SPOT Deployment

After having provided simulation results, we tested our framework together with the topology discovery protocol in real testbed. Figure 5 shows a map of 17 Sun SPOT sensor nodes (circles) and a base station (square) that are deployed in the offices at the Institute for Programming Structures and Data Organization (IPD) of the University of Karlsruhe. On each node we counted incoming and outgoing messages, as well as the sizes of the messages in bytes. These values were stored in the memory of each node and collected after the experiments were finished.

To assess the quality of the flooding prediction, the following experiment was repeated 10 times:

1. A simple flooding of a query was executed to determine the number of reached nodes for simple flooding.
2. Using the topology discovery protocol, the information required for the prediction was collected.
3. Using the topology information, the parameter p for the probabilistic flooding was computed with the aim of disseminating a query to all nodes of the network. Thus, we tried to determine the lowest p for which a reachability of 100% was predicted.
4. Based on the computed value of p, a query message was flooded into the network using probabilistic flooding.

Despite minor changes between the different experiments within the topology information, which can be attributed to environmental influences (e.g. open/closed doors in the used offices), the topology information was consistent throughout our experiments.

Table 2 shows the average results for the 10 experiments: Generally, the accuracy of the prediction is sufficient, even though there is a small difference between the 16.3 nodes reached by simple flooding compared to the probabilistic flooding with 15.4 nodes reached on average.

Table 2 shows messages required by the simple and the probabilistic flooding: The number of messages sent and received when the probabilistic flooding is used, is by far lower than the amount used by the simple flooding. Thus the amount of saved energy due to reduced communication clearly outweighs the small inaccuracy of the prediction.

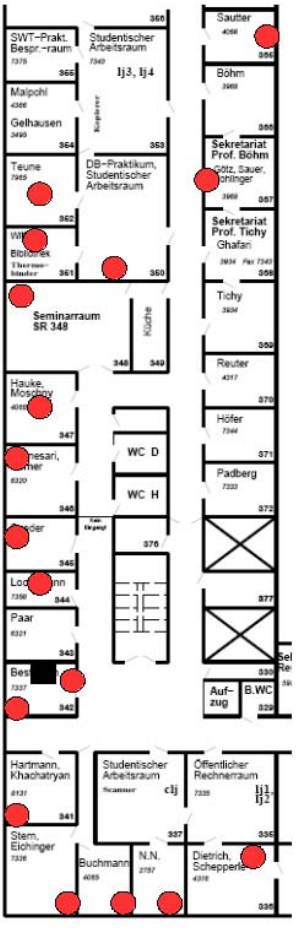


Fig. 5. Map of 17 Sun SPOTs and a Base Station deployed at the IPD

Table 2. Result of the flooding experiment using the Sun SPOT deployment

Flooding	Avg. Reached Nodes (of 17)	Messages Sent	Messages Received
Simple	16.3	16.3	63.8
Probabilistic	15.4	10.2	34

5 Conclusions and Future Work

It is challenging to realize energy-efficient query dissemination with predictable reachability and energy usage in sensor networks: Unnecessary transmissions should be generally avoided in order to save energy. On the other hand, it requires knowledge about the sensor network to find out which transmissions are actually required, but obtaining these information comes with an additional communication overhead.

In this paper we have used probabilistic flooding as a model to explore the relations between (1) energy consumption of the query dissemination phase, (2) the number of nodes reached and (3) the energy spent to gather structural information about the network which are required to parameterize probabilistic flooding. In particular, we have introduced an analytical framework that enables the base station to estimate the reachability and energy consumption of probabilistic flooding according to based on connectivity information. Furthermore, we have shown how to gather such information efficiently, and we have computed the break-even between energy saved and energy spent to obtain structural information. Both experiments with a simulator and an implementation with a testbed consisting of 17 SUN Spot nodes validate our findings.

As part of our future work we plan to consider "back links" in flooding, and other query dissemination strategies. In addition, we are interested in the relations between the energy spent for query dissemination and the accuracy of the query result returned.

References

1. Intel berkeley research lab data, http://db.csail.mit.edu/labdata/labdata.html
2. SUN Microsystems Inc., Small Programmable Object Technology (SPOT)
3. Xbow technology inc. wireless sensor networks
4. Chang, E.J.H.: Echo algorithms: Depth parallel operations on general graphs. IEEE Transactions on Software Engineering 8(4), 391–401 (1982)
5. Eugster, P.T., Guerraoui, R., Kermarrec, A.-M., Massoulieacute, L.: Epidemic information dissemination in distributed systems. Computer 37(5), 60–67 (2004)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness (1990)
7. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.S.J.: System architecture directions for networked sensors. In: Proc. 9th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (2000)
8. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw. (2003)
9. Kellner, S., Pink, M., Meier, D., Blaß, E.-O.: Towards a realistic energy model for wireless sensor networks. In: WONS 2008 (to appear) (January 2008)
10. Lim, H., Kim, C.: Multicast tree construction and flooding in wireless ad hoc networks. In: MSWIM 2000: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (2000)
11. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. In: SIGOPS (2002)

12. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tinydb: an acquisitional query processing system for sensor networks. In: ACM TODS (2005)
13. Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network. In: MobiCom 1999: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (1999)
14. Obraczka, K., Viswanath, K., Tsudik, G.: Flooding for reliable multicast in multi-hop ad hoc networks (2001)
15. Peng, W., Lu, X.-C.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: MobiHoc 2000: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing (2000)
16. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: HICSS 2002: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), vol. 9 (2002)
17. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC) (2002)
18. Yao, Y., Gehrke, J.: The cougar approach to in-network query processing in sensor networks. In: SIGMOD Rec. (2002)
19. Yao, Y., Gehrke, J.: Query processing in sensor networks. 2003. In: CIDR 2003: Proceedings of the First Biennial Conference on Innovative Data Systems Research (2003)

A Prediction Based Cross-Layer MAC/PHY Interface for CDMA Ad Hoc Networks

Pegdwindé Justin Kouraogo, François Gagnon, and Zbigniew Dziong

Department of Electrical Engineering, École de technologie supérieure,
Montréal, Québec, Canada H3C 1K3
{pegwindejustin.kouraogo.1,
francois.gagnon,zbigniew.dziong}@etsmtl.ca

Abstract. Variable environments in ad hoc networks require a joint control of physical (PHY) and medium access control (MAC) layers resources in order to optimize performance. In this paper, we propose a framework to perform such cross-layer control and optimization. The PHY layer and cross-layer engine estimate and predict the channel variations to select the users that will meet the signal-to-interference-noise ratio (SINR) requirement in the next time slot, for MAC layer optimization. We consider high capacity code division multiple access (CDMA) ad hoc networks working at fixed quality of service (QoS) requirement where nodes are equipped with matched filter receivers.

Keywords: Ad hoc network, cross-layer interface, prediction, CDMA systems, medium access control.

1 Introduction

In ad hoc networks, PHY layer variations strongly affect all the higher layers. For the optimization of the access to the wireless medium, MAC layer designers need a knowledge of the distribution over the channel fluctuations, in terms of the packet error rate (PER), the transmission rate, and the required transmission powers of the users in contention. Moreover, in high capacity CDMA ad hoc networks, each node can simultaneously decode several transmissions and the complexity of the problem grows with the number of the received users. A cross-layer study of the problem is a good approach to cope with such an issue.

A scheme for user's access optimization along the traffic variations in CDMA ad hoc networks was proposed in [1]. Voice activity process was modeled by a Markov chain to predict data users' capacity. We address the issue of the access optimization along the change of PHY layer rather than the change in voice process. The aim of this paper is to predict the channel variations, the user capacity for the target PER and the transmission powers in order to provide sufficient information to the MAC layer for the choice of an optimal operating point. The originality of this work relies on integrating this task in an entity called cross-layer engine composed of a processing part and a cross-layer interface (CLI) to ensure the information transport from PHY layer to MAC layer.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 293–306, 2008.
© Springer-Verlag Berlin Heidelberg 2008

On the cross-layer design area, some analogies may be established to a series of works published on time division multiple access (TDMA) ad hoc networks [2-5]. The authors introduced an original cross-layer study to control MAC queues stability over a fading channel. The proposed cross-layer framework works with a Spatial-TDMA scheduling that admits users transmissions only at a certain distance from the active receivers [6]. Our framework works with a color based CDMA protocol described in [7].

The remainder of this paper is organized as follows: In section 2 we present the MAC layer architecture used. In section 3 the general problem and the characterization of the logical and the physical channel parameters are presented. Section 4 provides details of the signal processing part of the framework. In section 5 and 6 the QoS modules and the cross-layer interface are provided, respectively. Section 7 presents the simulation results. Finally, section 8 concludes the paper.

2 Medium Access Control Structure

2.1 Architecture

The considered network supports two types of CDMA channels: a common CDMA channel to exchange connectivity informations, and several dedicated CDMA channels for scheduling and data transmission. Time is divided into super-frames. Each super frame is split into one connectivity frame and 10 data frames. When a node enters the network, it waits for the connectivity frame to exchange information with the neighboring nodes. The other active nodes execute a distributed algorithm to choose a dedicated CDMA code and update their bases of neighbors' transmission codes [7]. The data frame is composed of a scheduling slot and a data transmission slot as depicted in Figure 1. The scheduling slot is split into three mini-slots. In the first mini-slots, a node assignment algorithm performs an initial scheduling to set certain users in transmission mode and the others in reception mode. The second mini-slot gives an opportunity to the isolated nodes which lost the transmission/reception mode contention, to establish connections. In the last mini-slot, all the transmitters confirm the connection request by sending a request-to-send (RTS) packet with their QoS demand. The receivers decode the packets to extract the demands for MAC schedulers.

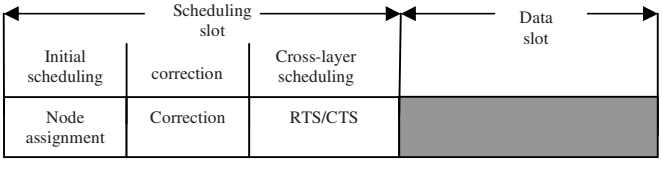


Fig. 1. Data frame

Cross-layer scheduling is then performed to allocate the required PER and transmission rate, and the users' responses are sent back by a clear-to-send (CTS) packet. The informations provided to the cross-layer scheduler from the PHY layer are the following:

- The users that meet the required SINR level
- The required powers of the transmitting nodes
- The channel gain level
- The out-of range interferences level

The cross-layer scheduling principle is illustrated in figure 2.

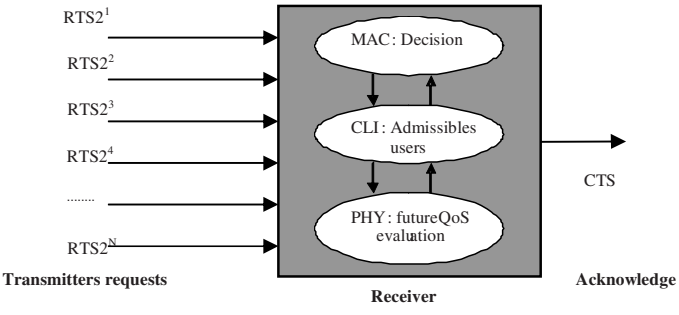


Fig. 2. Cross-layer scheduling principle

To transport the QoS demand to the receiver's MAC layer, the structure of the RTS packet is modified by adding some fields to carry the packet error rate (PER) and data rate the transmitter requires. These two fields are followed by a third additional field that transports a sequence of training bits for the channel estimation and prediction (see Figure 3a). Each CTS packet includes a series of two additional fields to transport the receivers' acknowledgement/non-acknowledgment (ACK/NACK) and the required transmission power, *P*, to meet the demand (see Figure 3b).

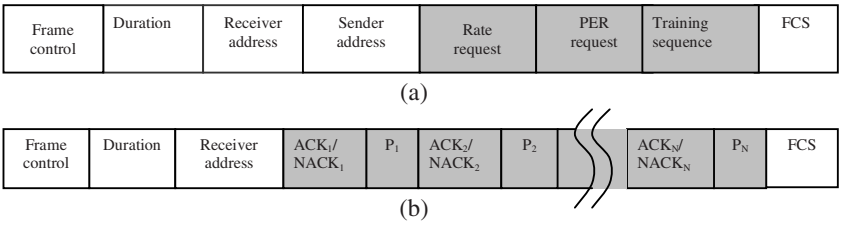


Fig. 3. a) RTS control packet b) CTS control packet

2.2 Neighborhood Topology

Although the nodes' positions change due to their possible motion, the network snapshot of the connections is fixed from the cross-layer scheduling mini-slot until the end of the data transmission slot. This configuration is the basic neighborhood topology of our study (Figure 4). It consists of a set of receivers connected with certain transmitters in the neighborhood. With multi-user reception capability, the receiver decodes the intended signal and further listens to the unintended ones. Note that by *multi-user reception* it is meant that the nodes use a conventional matched filter and *multi-user detection* when a supplementary detection algorithm is implemented after the matched

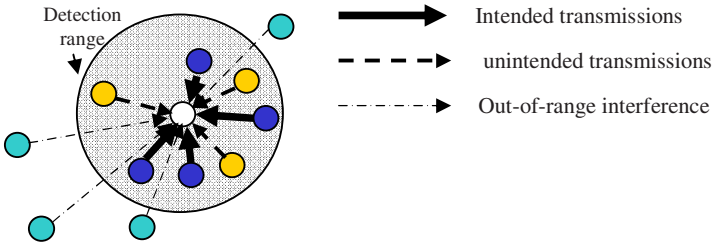


Fig. 4. Basic receiver configuration

filters. It is also important to mention that the unintended transmissions are listened to, in order to reduce the interferences in the network.

3 General Approach

3.1 Cross-Layer Framework

Our approach consists of the designed framework depicted in figure 5. Four main modules have to be developed.

Observation/estimation module. The observation and estimation module extracts the demodulated samples from the transceiver to estimate the PHY layer parameters. Because multi-user reception deals with multiple access interferences, the main parameters to be estimated are the channel gain, $h(t)$, and the out-of- range interferences, I_0 .

Prediction module. The prediction module forecasts the parameters for the next data slot. The predicted parameters are used by the QoS module and the cross-layer interface to process the admissibility of transmitting nodes.

QoS module. The QoS module calculates the QoS parameters of the different links and their required transmission powers based on the channel prediction. A more detailed description of the parameters will be given in section 5.

Cross-layer interface (CLI). CLI ensures the communication between the two layers. It transports the MAC layer synchronization information to the PHY layer to allow the pilot bits extraction for the estimation and prediction. It also provides the predicted values from the PHY to MAC layer for the optimization purpose. MAC layer optimization consists to select the best configuration of transmitters based on the future state of the channel, the interferences, the packet delay constraints and the availability of the QoS. This topic will be address in the next work.

3.2 MAC Logical Channel Characterization

Logical channels at the MAC layer are characterized by a transmission rate, R , a maximum tolerable PER, and a maximum tolerable delay, τ . A given PER corresponds to a target BER at the PHY layerThe BER is well approximated by the

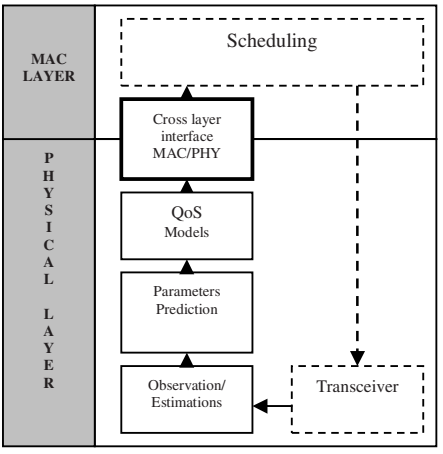


Fig. 5. Cross-layer framework

probability of error P_b, which depends on the modulation used, and the received SINR. Several techniques of modulation may be used to transport the bits stream over the channel. We adopt here the simplest: the binary phase shift keying (BPSK). The probability of error of the BPSK is a Q-function of the SINR.Finally, MAC required PER is converted into a targer SINR for the calculations as follows:

$$\mathrm{SINR} = \mathrm{Q}^{-1}\left(\frac{\mathrm{BER}}{\sqrt{2}}\right), \tag{3.1}$$

where $\mathrm{Q}(\mathrm{x}) = \frac{1}{\sqrt{2\pi}}\int_{\mathrm{x}}^{\infty} \mathrm{e}^{-\mathrm{t}^2/2}\mathrm{dt}$.

In addition to the SINR control mechanism considered here, the other link error control mechanisms such that the automatic repeat request (ARQ), the forward error correction (FEC) or the interleaving may be considered to improve the quality of the packets reception. Theses topics will be investigated in the future researches.

3.3 Physical Channel Characterization

The SINR reflects the physical impairments introduced by the wireless channel on the signal. Since logical channels work on frame of few milliseconds, packets received are mainly affected by small scale channel fading. This type of fading follows a Rayleigh distribution [8-10]. The standard computer model used to generate the channel coefficients is Jake's simulator [9, 10]. Consequently, Rayleigh fading coefficients are computed by summing M delayed sinusoids to take into account the scattering and the Doppler effects. The equations of channel gain used in our simulations are covered in [10]:

$$h(t)=\frac{1}{\sqrt{M}}\left(e^{j(2\pi f_d t+\phi_n)}+e^{-j(2\pi f_d t+\phi_n)}\right)+\frac{1}{\sqrt{M}}\left(\sqrt{2}.\sum_{n=1}^{M_0}\left[e^{j\left(2\pi f_d\cos\frac{2n\pi}{M}t+\phi_n\right)}+e^{-j\left(2\pi f_d\cos\frac{2n\pi}{M}t+\phi_n\right)}\right]\right), \quad (3.2)$$

where M is an odd integer such that $M_0=\frac{1}{2}.\left(\frac{M}{2}-1\right)$. The term $\frac{1}{\sqrt{M}}$ is a normalization factor to normalize the average power to unity, f_d the maximum Doppler frequency such as $f_d=\frac{f_c.v}{c}$, f_c the carrier frequency and v represents the mobile speed. The n-th sinusoid is received with a delay τ_n and a phase ϕ_n given by $\phi_n=2\pi f_c.\tau_n$ such that ϕ_n is random and uniformly distributed over $[0,2\pi]$.

4 Signal Processing

4.1 Parameters Observation and Estimation

Observation. We take advantage of the knowledge of the training bits we have in the RTS packet in order to estimate the channel gain. The observation samples are acquired from the output of the matched filter's bank. At the i-th bit instant, user *k* sample is expressed as:

$$y_k(i)=h_k(i)A_k(i)+I_{0k}(i)W+\eta_{0k}(i)\ ,\ k=1,...,K\ \text{users}\ , \quad (4.1)$$

where $h_k(i)$ is the channel gain. $A_k(i)$ is the transmitted signal amplitude defined by $A_k(i)=d_k(i)\sqrt{P_{max}}$, assuming that control packets are sent at the maximum power level P_{max}. $d_k(i)$ represents the known training symbol. $I_{0k}(i)$ and $\eta_{0k}(i)$ are the out-of range interference level and the additive white Gaussian noise samples after the correlation applied by the matched filter.

Estimation of the channel gain. A simple estimate of the channel gain $he_k(i)$ can be calculated as follows:

$$he_k(i)\approx\frac{y_k(i)}{d_k(i)\sqrt{P_{max}}}\ ,\ k=1,...,K\ \text{users}\ . \quad (4.2)$$

4.2 Parameters Prediction

With the knowledge of the N_{Pilot} bits of RTS packet, a collection of N_{Pilot} channel samples is buffered at the output of the estimator, to feed the input of the predictor. This involves a bank of *K* predictors for *K* transmitters, providing a supplementary complexity to be considered for the choice of the prediction algorithm. Another dominant criterion in this choice is the long range prediction capability. For this purpose we prefer linear prediction, over the heavy computation methods presented in

[9, 10]. Linear prediction is well described in [11]. Channel gain process $he_k(i)$ $i = 1, 2, ...$ is modeled as an auto-regressive (AR) process. Then, we determinate the finite impulse response (FIR) filter coefficients a_l, where $l = 0, 2, ..., L-1$, that minimize the mean square error (MSE) between the estimated and the predicted samples. L is the prediction order of the filter. The channel gain $hp_k(i)$ at bit i is then computed using the L past samples and the filter coefficients as follows:

$$hp_k(i) = \bar{a}_k(i) \cdot \bar{he}_k(i) = \sum_{l=0}^{L-1} a_l^k(i).he_k(i-l), k = 1, ..., K, \quad (4.3)$$

where $\bar{he}_k(i) = [he_k(i), he_k(i-1), ..., he_k(i-L+1)]^T$ is the vector of the L past samples of the k-th user's channel gain. The vector $\bar{a}_k(i) = [a_0^k(i), a_1^k(i), ..., a_{L-1}^k(i)]^T$ represents the filter coefficients. The coefficients $\bar{a}_k(i)$ that minimize the MSE are obtained by the orthogonality principle [11], and computed using the following matrix inversion:

$$\bar{a}_k(i) = [R_k(i)]^{-1}.\bar{r}_k(i), k = 1, ..., K, \quad (4.4)$$

where $[R_k(i)]$ is the correlation matrix of the channel coefficient at the instant i such that $[R_k(i)]_{n,m} = E\{he_k(i-n).he_k^*(i-m)\}$, $n, m = 0, 1, ..., L-1$. The sign $(*)$ represents the complex conjugate operation. The vector $\bar{r}_k(i) = [r_0^k(i), r_1^k(i), ..., r_{L-1}^k(i)]^T$ contains the L samples of channel autocorrelation so that $r_{l,k}(i) = E\{he_k(i).he_k^*(i-l)\}$, $l = 0, 1, ..., L-1$.

To compute the K users' channel coefficients, K matrix inversion is performed at each bit. In order to minimize the impact of processing time on the network delay jitter, we choose to adaptively update the FIR filter coefficients by the least mean square (LMS) algorithm. Finally the module is typically a bank of K LMS predictors which work in two phases:

- During *the training phase* the prediction coefficients are updated at each instant according to the prediction error between the estimated and predicted samples as follows:

$$\bar{a}_k(i) = \bar{a}_k(i-1) + \mu_k \cdot e_k^*(i-1) \cdot \bar{he}_k(i-1), k = 1, ..., K, \quad (4.5)$$

where μ is the step size parameter that guides the convergence of the algorithm, and takes its value in the interval $[0,1]$. $e_k^*(i) = hp_k^*(i) - he_k^*(i)$ is the complex conjugate of the prediction error given by the difference between the predicted and the estimated sample, $\bar{he}_k(i)$ the vector of the L past samples of the channel estimate.

- *The tracking phase* goes from the beginning of the first bit after the training sequence of the RTS packet to the end of the data packet. During this period there is

no knowledge of the channel estimates. We estimate the desired channel value by interpolating it by the last predicted sample as follows:

$$\tilde{he}_k(i) = hp_k(i-1),\ k = 1,...,K, \tag{4.6}$$

Accordingly, the error used for the adaptation is calculated by the expression:

$$e_k^*(i) = \left(hp_k^*(i) - \tilde{he}_k^*(i)\right) = \left(hp_k^*(i) - hp_k^*(i-1)\right), k = 1,...,K, \tag{4.7}$$

Assuming a data packet of N_{data} bits, the slot level channel gain is obtained by taking the average of the N_{data} predicted samples over the data slot:

$$h_k = \frac{1}{N_{data}} \sum_{i=N_{RTS}+N_{CTS}}^{N_{RTS}+N_{CTS}+N_{data}} hp_k(i), \tag{4.8}$$

N_{RTS} and N_{CTS} designate the RTS and CTS packets' lengths, respectively. We compare our prediction scheme to the current estimation method that interpolates the channel gain on the next data slot by averaging the estimated samples during the training sequence acquisition:

$$h_k = \frac{1}{N_{RTS}} \sum_{i=1}^{N_{RTS}} hp_k(i). \tag{4.9}$$

5 QoS Management/Admission Parameters

One class of traffic is considered in this study. Each link is characterized by: a data rate R_i, a required SINR γ_i, a transmission power P_i, and a maximum allowable power P_{max}. Several objectives exist to optimize the medium access. The simplest way is to minimize the interference level in order to improve the system's capacity. Minimum level of interference is achieved when all users send their packets with the strict minimum power to meet MAC QoS demand with equality. Converting the QoS demand in term of SINR constraint, the condition is expressed by [12]:

$$\frac{W}{R_i} \frac{h_i P_i}{\sum_{j \neq i} h_i P_i + (I_0 + \eta_0)W} = \gamma_i \quad i = 1,...,N, \tag{5.1}$$

where W is the transmission bandwidth, $h = [h_1, h_2, ..., h_N]$ the channel gain vector, η_0 the spectral density of the noise, and I_0 the out-of-range interferences. Rewriting the equation (5.1) for *N* users and proceeding to some transformations lead us to the receiver admissibility condition given by [12]:

$$\sum_{j=1}^{N} \frac{1}{\left(\frac{W}{R_j} \cdot \frac{1}{\gamma_j} + 1\right)} \leq 1 - \frac{(I_0 + n_0).W}{\min_i \left[P_i^{max} h_i \left(\frac{W}{R_i \gamma_i} + 1\right)\right]_{i=1}^{N}}, \tag{5.2}$$

5.1 User's Admission Parameters

For a fixed transmission rate, users' admission at the receiver is limited by the following criteria [13]:

$$\frac{N}{\left(\frac{W}{R_0}\cdot\frac{1}{\gamma_0}+1\right)} \le 1-\delta_i^{max}, \tag{5.3}$$

where parameter $\delta_i^{max} = \dfrac{(I_0+n_0)W}{\min_i\left[P^{max}h_i\left(\frac{W}{R_0\gamma_0}+1\right)\right]_{i=1,\ldots,N}}$ is the ratio of the out-of-range interferences plus the noise power over the minimum value of the links' parameters, g_i, where $g_i = P^{max}h_i\left(\frac{W}{R_0\gamma_0}+1\right)$ for $i=1,\ldots,N$. The parameter $b=\frac{1}{\frac{W}{R_0\gamma_0}+1}$ represents the *"bandwidth coefficient"* that the receiver can assign to each admissible user. The statement (5.3) shows that the total *bandwidth* is affected by the maximum value of δ_i^{max}, corresponding to the weakest link's parameter. This value can be predicted since it depends on the users channel gain and the interference term. The number of signals that can meet the QoS request is calculated by:

$$N=\left(\frac{W}{R_0\gamma_0}+1\right).\left(1-\delta_i^{max}\right)_{i=1,..,N}, \tag{5.4}$$

One can achieve this limit when the parameter δ_i^{max} goes to zero i.e. $P^{max}\to\infty$. So the maximal capacity is attained at unlimited transmission power such that the capacity is $N^{max}=\left(\frac{W}{R_0\gamma_0}+1\right)$. The non-admissibility of a set of N candidate users occurs when the total *bandwidth* of the admissible users' exceeds the difference between unity and δ_i^{max}. In addition, due to the channel variations, the difference $\left(1-\delta_i^{max}\right)$ may be reduced more. Based on the predicted value of the parameter δ_i^{max} and the knowledge of the *bandwidth coefficient* b, we can determine for the next time slot the admissible set of users. This task is done by the cross-layer interface which uses the links parameters δ_i $i=1,\ldots,N$ to perform iteratively the described users' admission.

5.2 Required Transmission Powers

The CTS packet transport the required powers calculated by the following expression adopted from reference [13]:

$$P_i=\frac{(I_0+n_0)\cdot W}{h_i\left(\frac{W}{R_0\gamma_0}+1-N\right)}. \tag{5.5}$$

6 Cross-Layer Interface

In reception mode, nodes execute the following tasks:

Step 1: Listen to the entire neighborhood
Step 2: Analyze RTS packets headers to find the destination addresses, extract the QoS requests of the transmitters desiring to establish connections, detect the out-of-range node for interference cancellation.
Step 3: Estimate and predict the channel gain h_i, the out-of-range interferences I_0, and calculate the QoS parameters b and δ_i $i = 1,...,N$.

Step 4: Sort in ascending order users according to the parameter δ_i^{max} and execute the following algorithm to search the admissible users:

Initialization: $N_{users} = 1$

Repeat until $\left(\dfrac{N_{users}}{\left(\dfrac{W}{R_0} \cdot \dfrac{1}{\gamma_0} + 1 \right)} > 1 - \delta_{N_{user}}^{max} \right) \text{or} \left(N_{users} > N^{max} \right)$

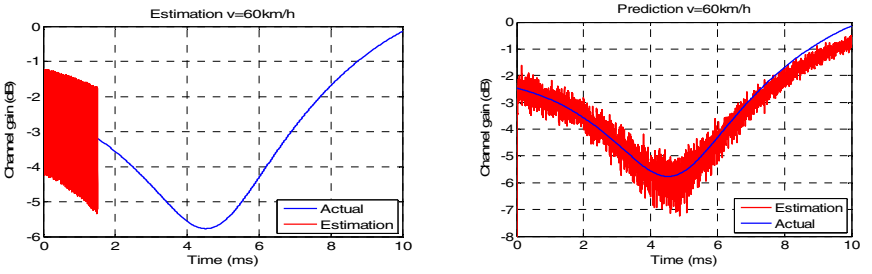
$N_{users} = N_{users} + 1$

End repeat

Step 5: According to the allowable delay, the evolution channel and interferences in the next slots, MAC layer schedules the best configuration of transmitter in the set of the N_{users} links.

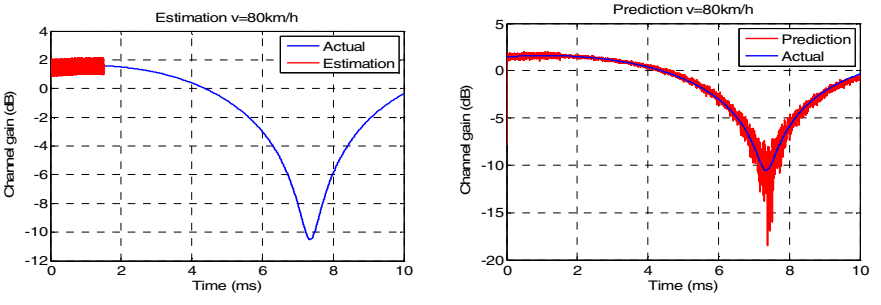
7 Simulation Results

The foundation of the framework relies on the signal processing algorithms. As the algorithms are destinated to be used in the real time systems, the simulation appears to be a more accurate tool than the analytical methods to evaluate the performances. We consider a system where nodes share a bandwidth of W=450MHz, and transmit RTS packet at $P_{max} = 5W$. We assume a spectral density of the background noise of $N_0 = 10^{-9}$, an out-of-range interferences level of $I_0 = -3dB$, a spreading gain of $G = 128$ such that the bit rate is $R = 3.5Mbits/s$. For simplicity, we define the frame to be a block of RTS-CTS and Data packets. Each frame has a length of $L_{Frame} = 35000bits$ with 10ms duration and the observation window $L_w = 20 \times 264$ pilot bits. LMS algorithm is implemented with a step size parameter $\mu = 0.006$ and a prediction order of $L = 20$. We first simulate the estimation and prediction for several node's speeds. The estimation is performed during the acquisition of the $20 \times 264bits$ of the RTS packet as showed in figure 6.a, 7.a, and 8.a. In figures 6.b, 7.b and 8.b we can see the prediction over the whole frame. These results also show the attenuation experienced by the frames due to the channel fading. The maximum attenuations are -3 dB and -10dB, in figure 6 and figure 7 respectively.



(a) (b)

Fig. 6. (a) Channel estimation, (b) prediction



(a) (b)

Fig. 7. (a) Channel estimation, (b) prediction

The performance of the predictor is evaluated by averaging over 200 realizations the MSE between the actual channel gain and the prediction on one data packet. A comparison of the prediction is performed with the classical estimation method. The results are plotted for a range of Doppler frequencies in Figure 9. By fixing the Doppler frequency to $f_d = 100\text{Hz}$, we then plot the average MSE for different signal-to-noise ratio (SNR) in Figure 10 and several observation windows in Figure 11 Figure 9 shows that estimation gives better performances than prediction for a Doppler frequency inferior to 16.7 Hz as the channel gain is flat at low frequencies. When the frequency increases, more fading occurs on data packet and the prediction scheme is superior to the classical estimator. In Figure 10, we can observe that the estimator performance exceeds the predictor's when the SNR is lower than 4 dB, this is essentially due to the propagation of the error induced by the noisy observations in the LMS algorithm. However, over 4dB the predictor gives better performances. Figure 11 illustrates the superiority of the prediction scheme over the estimation when we variate the observation windows. Moreover we can see that prediction error decreases when the observation window is enlarged.

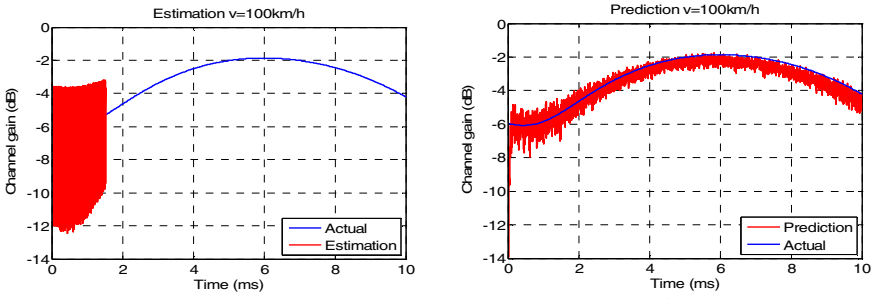


Fig. 8. (a) Channel estimation, (b) prediction

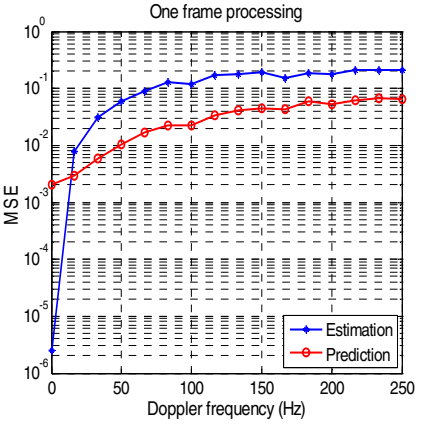


Fig. 9. Average MSE as a function Doppler frequency

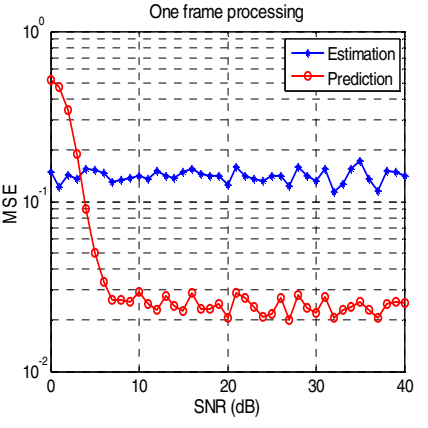


Fig. 10. Average MSE as a function of signal-to-noise ratio

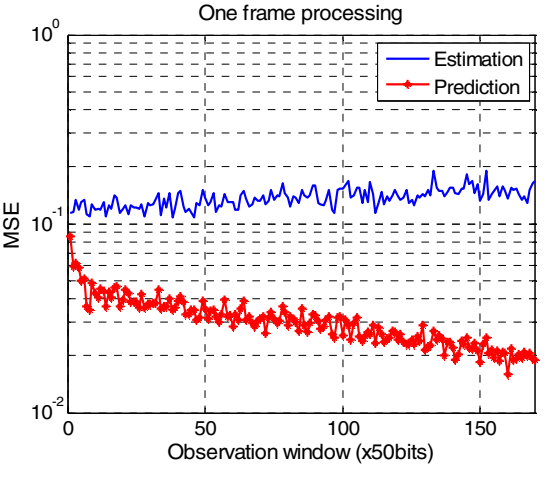


Fig. 11. MSE as a function of Observation

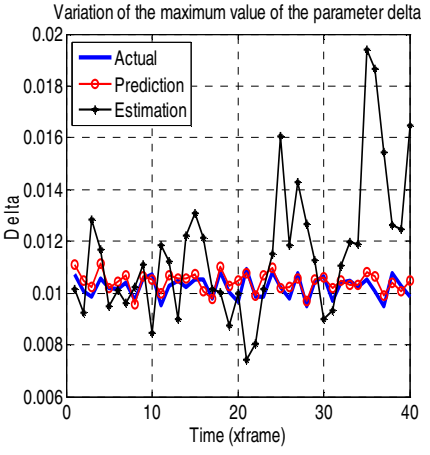


Fig. 12. Variation of parameter δ_i as a function of time, $W = 83.5\text{MHz}$, $R = 1\text{Mbits/s}$, $f_d = 80\text{Hz}$ $L_{Frame} = 8696\text{bits}$, $L_w = 400\text{bits}$

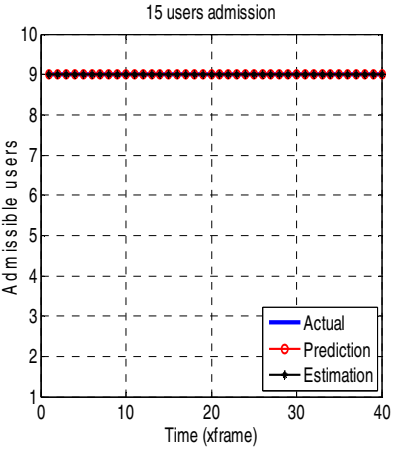


Fig. 13. Number of admitted transmitters as a function of $W = 83.5\text{MHz}$, $R = 1\text{Mbits/s}$, $f_d = 80\text{Hz}$ $L_{Frame} = 8696\text{bits}$, $L_w = 400\text{bits}$

Figure 12 shows the variation of the parameter δ_i in time. The predictor gives a better result than the estimator. In figure 13, we plot 15 users' admission. Due to the high number of transmitters, many of them experience high channel gain, and the maximum capacity is easily achieved when the algorithm uses the actual, the predicted and the estimated values of the parameter delta.

8 Conclusion

In this paper we presented a cross-layer interface between MAC and PHY layer. We take advantage of the samples produced by the matched filter to predict the channel change for MAC layer optimization process. As the capacity is influenced by the channel fluctuations, a cross-layer algorithm determines the set of admissible users based on minimum interference criteria. Then simulation results presented showed the superiority of the prediction over the classical estimation scheme. In the future work, the users' admission will be study more extensively in the case of different ad hoc scenarios, with more realistic radio propagation models. MAC layer optimization process will be also addressed.

Acknowledgment. The authors wish to thank Dr. Mohamed Haidar for his insightful comments and remarks.

References

1. Comaniciu, C.: Integrated Access Control and Detection for QoS Multimedia CDMA Networks. In: Electrical and Computer Engineering Department, Graduate School-New Brunswick New Jersey (2002)

2. Liu, Q., Zhou, S., Giannakis, G.B.: Cross-Layer Modeling of Adaptive Wireless Links for QoS Support in Multimedia Networks. In: Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE 2004). IEEE Computer Society, Los Alamitos (2004)
3. Qingwen, L., Shengli, Z., Giannakis, B.G.: Queuing With Adaptive Modulation and Coding Over Wireless Links: Cross-Layer Analysis and Design. IEEE Transactions on Wireless Communications 4(3) (2005)
4. Liu, Q., Zhou, S., Giannakis, G.B.: Cross-Layer Scheduling With Prescribed QoS Guarantees in Adaptive Wireless Network. IEEE Journal on Selected Areas in Communications 23(5), 1056–1066 (2005)
5. Qingwen, L., Shengquan, H., Giannakis, B.G.: Cross-Layer Scheduling Algorithm With QoS Support in Wireless Networks. IEEE Journal on Selected Areas in Communications 55(3) (2006)
6. ElBatt, T., Ephremides, A.: Joint scheduling and power control for wireless ad hoc networks. IEEE Transactions on Wireless Communications 3(1), 74–85 (2004)
7. Zhang, J., et al.: Performance Evaluation of A Multiuser Detection Based MAC Design for Ad Hoc Networks. In: IEEE VTC2007-Fall, Baltimore, USA, September 30 – October 3, 2007, pp. 294–298 (2007)
8. Rappaport, T.S.: Wireless Communications Principles and Practice. In: Rappaport, T.S. (ed.) Communications Engineering and Emerging Technologies Series. Prentice Hall, Englewood Cliffs (2002)
9. Duel-Hallen, A., Hu, S., Hallen, H.: Long Range Prediction of Fading Signals: Enabling Adaptive Transmission for Mobile Radio Channels
10. Kurniawan, A.: Predictive Power Control in CDMA Systems, in Institute for Telecommunications Research, Division of Information Technology, Engineering and the Environment, The University of South Australia (2003)
11. Haykin, S.: Adaptive filter theory. In: N.J.P.H. (ed.), Upper Saddle River (2002)
12. Ashwin, S., Sarath, K.P., Jack, M.H.: Power Control and Resource Management for a Multimedia CDMA Wireless System. In: PIMRC 1995 (1995)
13. Ramakrishna, S., Holtzman, J.M.: A Scheme for Throughput Maximization in a Dual-Class CDMA System. IEEE Journal on selected areas in communications 16(6), 830–844 (1998)

Utility-Based Uplink Power Control in CDMA Wireless Networks with Real-Time Services

Timotheos Kastrinogiannis, Eirini-Eleni Tsiropoulou, and Symeon Papavassiliou

Network Management & Optimal Design Laboratory (NETMODE)
School of Electrical & Computer Engineering
National Technical University of Athens (NTUA)
9 Iroon Polytechniou str. Zografou 15773, Athens, Greece
{timothe,eetsirop}@netmode.ntua.gr, papavass@mail.ntua.gr

Abstract. In this paper we address the problem of efficient power allocation in the uplink of CDMA wireless networks, emphasizing on the support of real-time services' QoS prerequisites. The corresponding problem is formulated as a non-cooperative game where users aim selfishly at maximizing their utility-based performance under the imposed physical limitations. A user's utility reflects its degree of satisfaction with respect to its actual throughput performance, QoS requirements fulfillment, and the corresponding power consumption. The existence and uniqueness of a Nash equilibrium point of the proposed Uplink Power Control (UPC) game is proven, where all users have attained a targeted SINR value or transmit with their maximum power, leading essentially to an SINR-balanced network. The properties of equilibrium in a pure optimization theoretical framework are studied, and the tradeoffs between users' overall throughput performance and real-time services strict QoS requirements in channel aware resource allocation processes are revealed and quantified. Finally, a distributed iterative algorithm for computing UPC game's equilibrium is proposed and its efficiency is illustrated via simulation and analysis.

Keywords: Wireless networks, utility-based resource allocation, QoS, real-time services.

1 Introduction

Considerable research efforts have been devoted to the combined problem of power and rate allocation for the downlink [1], [2] of a code division multiple access (CDMA) system. Furthermore, users' selfish behavior as well as the necessity of efficiently supporting their various QoS requirements, allows the formulation of the power and rate control problem in the uplink of such systems as a non-cooperative game, where each node wishes to maximize its own level of satisfaction (as expressed by an appropriately defined utility function) [3]-[5].

In this paper, we study the problem of power allocation in the uplink of CDMA wireless networks focusing on the support of real-time services. We propose a generic utility-based framework for assigning users' transmission powers, which maximizes the efficiency of the system in terms of user's utility-based degree of satisfaction, which

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 307–320, 2008.
© Springer-Verlag Berlin Heidelberg 2008

accounts for their actual throughput expectations and QoS requirements, as well as minimum power consumption. Initially, we formulate the Uplink Power Control (UPC) problem as a non-cooperative game, where each user aims at maximizing its performance. The existence and uniqueness of a Nash Equilibrium (NE) of the proposed UPC game is proven while a distributed, iterative algorithm for reaching equilibrium is presented. Then, the properties of Nash Equilibrium in the proposed UPC game are analyzed and the strong correlations among real-time users' QoS requirements, system's modulation and coding schemes and their power limitations are revealed.

In [6], a utility-based approach for allocating resources to real-time (RT) users in the uplink of a CDMA system is also adopted. However, in that work, even if RT users' utilities reflect their services' satisfaction with respect to the achievable throughput performance, their corresponding energy consumption has not been taken into account and hence, utilities abstract definitions avoid reflecting properly their QoS prerequisites. In [5], linear utilities of users' achieved goodput are considered and thus, real-time services' QoS requirements are expressed as statistical delay constraints. Such an approach is not always efficient [7], since even if the delay constraints of a real-time user are satisfied, the degradation of their service quality can not be avoided due to possibly bad channel conditions and variations.

The rest of the paper is organized as follows. In section 2, the system model and necessary background information is presented. In section 3, real-time users' QoS properties are studied and mapped to appropriate utility functions. In section 4, the proposed uplink power control non-cooperative game formulation is described, while in section 5 its solution is presented. Section 6, discusses the potential use of users expected QoS performance properties at equilibrium towards the design of a node's self-optimizing approach. Finally, some numerical results are provided in section 7 while section 8 concludes the paper.

2 System Model and Background Information

We consider the uplink of a single cell time-slotted CDMA wireless system with $N(t)$ continuously backlogged users at time slot t, where $S(t)$ denotes their corresponding set. A time slot is a fixed interval of time and could consist of one or several packets. Users' channel conditions are affected by shadow fading, fast fading and long-time scale variations and thus, can be modeled as a stationary time-varying stochastic process. Let as denote by $G_i(t)$ the corresponding path gain of user $i \in S$ at time slot t. In the following, assuming fixed users' channel conditions within the duration of each time slot, we omit the notation of the specific slot t in the notations and definitions we introduce. At the beginning of each time slot t, users' Uplink Power Control (UPC) mechanisms make decisions on their transmission power and resulting rate in a distributed manner. Note that a node's transmission power and rate are also fixed within the duration of a time slot.

Let us denote by P_i the uplink transmission power for user i in the slot under consideration which however is limited by its maximum power value P_i^{Max}. Let us also denote by $\gamma_i \triangleq E_b/I_o$ the bit energy to interference density ratio for user i and by R_i the achievable uplink transmission rate. Therefore, the received γ_i at the base station for each user i is given by:

$$\gamma_i\left(R_i, P_i, \overline{P}_{-i}\right) = \frac{W}{R_i}\frac{G_i P_i}{\theta\sum_{j=1}^{N} G_j P_j - \theta G_i P_i + I_0} = \frac{W}{R_i}\frac{G_i P_i}{I_{-i}(\overline{P}_{-i})} \tag{1}$$

where θ denotes the orthogonality factor, W is the system's spreading bandwidth, $\overline{P}_{-i}$ denotes the users' power allocation vector excluding user i and I_0 includes the background noise and intercell interference. Thus, $I_{-i}(\overline{P}_{-i})$ actually denotes the network interference and background noise at the base station when receiving data from user i and is given by:

$$I_{-i}(\overline{P}_{-i}) = \theta\sum_{j=1}^{N} G_j P_j - \theta G_i P_i + I_0 \tag{2}$$

To align real-time (RT) users' various services flow characteristics under a common optimization framework each mobile user is associated with a suitable utility function U_i which represents his degree of satisfaction in relation to the expected tradeoff between its utility-based actual uplink throughput performance and the corresponding energy consumption per time slot. Therefore, this can be expressed as:

$$U(R_i^*, P_i, \overline{P}_{-i}) = \frac{T_i(R_i^*, P_i, \overline{P}_{-i})}{P_i} = \frac{T_i(R_i^F \cdot f_i(\gamma_i), P_i, \overline{P}_{-i})}{P_i} \tag{3}$$

where $R_i^* \equiv R_i^F \cdot f_i(\gamma_i)$ denotes user's i actual uplink transmission rate (i.e. goodput) at the under consideration time slot, R_i^F is its fixed designed transmission rate and f_i denotes its efficiency function. The latter represents the probability of a successful packet transmission for user i, and is an increasing function of his bit energy to interference ratio γ_i at any time slot. A user's function for the probability of a successful packet transmission at fixed data rates depends on the transmission schemes (modulation and coding) being used, and can be represented by a sigmoidal-like function of its power allocation for various modulation schemes [1], as Fig.1 illustrates. Therefore, a user's i, efficiency function f_i has the following properties:

1) f_i is an increasing function of γ_i.
2) f_i is a continuous, twice differentiable sigmoidal function with respect to γ_i .
3) $f_i(0) = 0$ to ensure that $T_i = 0$ and $U_i = 0$ when $P_i = 0$.
4) $f_i(\infty) = 1$.

The validity of the above properties has been demonstrated in several practical scenarios with reasonably large packet sizes M (i.e. $M \geq 100$bits) [4], [5].

Finally, $T_i(R_i^*, P_i, \overline{P}_{-i})$ is a sigmoidal function of user's i actual data rate R_i^* and reflects its degree of satisfaction in accordance to its service actual throughput expectations and QoS requirements fulfillment at every time slot. In the following section, we first study and specify the desired properties and QoS prerequisites that characterize

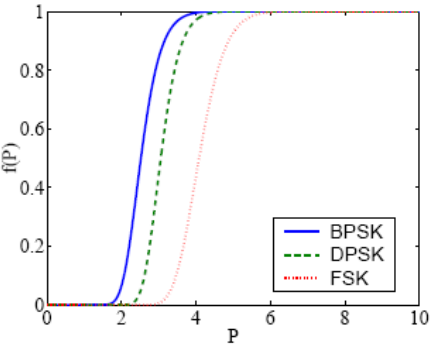


Fig. 1. Probabilities of packet transmission success for BPSK, DPSK, and FSK modulation schemes

the performance of RT users' services and then, we analyze and justify our proposed methodology for mapping a real-time user's degree of satisfaction with respect to its corresponding service performance into a proper actual throughput utility function T_i.

3 Satisfying Real-Time Services QoS Requirements

Considering a soft QoS requirements framework [6], in the following we assume that real-time services' requirements consist mainly of a constant target actual rate per time slot and an appropriate elasticity factor. The target actual uplink rate $R^*_{T,i}$, indicates the ideal value of its actual transmission rate per time slot, at which its service QoS requirements are fulfilled, while the Elasticity Factor (EF_i) determines the expected bounds of its actual achieved throughput deviations with respect to the target actual rate.

Specifically, a RT user's elasticity factor determines the minimum acceptable levels of its expected actual throughput (i.e. $R^*_{Min,i} = R^*_{T,i} - EL_i$). Moreover, we argue that when a user's achieved actual transmission rate remains within the range of $[R^*_{Min,i}, R^*_{T,i})$, his actual throughput utility must be a slowly decreasing function of his actual data rate. On the other hand, when $R^*_i < R^*_{Min,i}$ then his actual throughput utility should be a rapidly decreasing function of R^*_i, indicating its priority in occupying additional system resources. The previous design option gives to a user's power and rate control mechanism, operating over fast fading channels environment, the enhanced flexibility of decreasing its actual uplink throughput up to a certain level ($R^*_{Min,i}$), if required due to its potentially bad instantaneous channel conditions, without however excluding the user from transmitting data at that corresponding time slot. The later would occur, if a step function of a RT user's actual transmission rate was used to reflect the user's corresponding degree of satisfaction.

Furthermore, we argue that an additional increment of a RT user's actual data rate from its target one, must not contribute to an analogous increment of its actual

throughput degree of satisfaction and thus, the latter should tend asymptotically to its maximum value, as $R_i^* \to \infty$ (i.e. $T_i(\infty) = 1$). The previous argument is based on the observation that since RT service QoS prerequisites are fulfilled when its required target bit rate is achieved, an additional improvement of user's actual throughput performance will not further improve its degree of satisfaction.

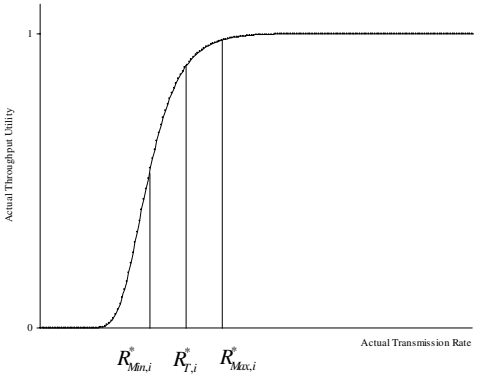


Fig. 2. Actual throughput utility function T_i, for real-time services

Moreover, by restricting up to a specific level ($R^*_{Max,i}$) the achieved actual uplink data rates of RT users, that due to their temporarily good transmission environment can obtain more recourses than required, we get the advantage of reallocating the excess system resources to the un-favored RT users (e.g. those with temporally bad transmission environment) in order to increase not only their performance satisfaction but also the overall number of RT users that can be simultaneously served. For practical reasons, due to users' hardware limitations, we regard that a RT user's actual throughput utility has reached a value close to the maximum when $R^*_{Max,i} = R^*_{T,i} + EL_i$ (i.e. $T_i(R^*_{Max,i}) = 1 - \varepsilon$, where ε is an arbitrarily small positive number, e.g. $\varepsilon = 5\ 10^{-5}$).

With respect to the previous discussion and analysis, a RT user's i actual throughput utility T_i has the following properties (Fig.2):

1) T_i is an increasing function of R_i^* .

2) T_i is a continuous, twice differentiable sigmoidal function of R_i^* , with unique inflection point $R^*_{Infl,i}$ determined by:

$$R^*_{Infl,i} = \left\{ R_i^* : \left. \frac{\partial^2 T_i(R_i^*)}{\partial (R_i^*)^2} \right|_{R_i^* = R^*_{Min,i}} = 0 \;\middle|\; R^*_{Min,i} = R^*_{T,i} - EL_i \right\} \tag{4}$$

3) $T_i(0) = 0$ to ensure that $T_i = 0$ when $R_i^* = 0$,

4) $T_i(\infty) = 1$ and

5) $T_i(R^*_{Max,i}) = 1-\varepsilon$, where $\varepsilon = 510^{-5}$.

Finally, in accordance to (3), by setting $R_i^F = R^*_{Max,i}$ a RT user's i utility function is defined as follows:

$$U_i(P_i, \overline{P}_{-i}) = \frac{T_i(R^*_{Max,i} \cdot f_i(\gamma_i))}{P_i} \tag{5}$$

where $R_i^* \in \left[0, R^*_{Max,i}\right]$, since $f_i(\gamma_i) \in [0,1] \ \ \forall \gamma_i \geq 0$.

4 The Non-cooperative Uplink Power Control Game

As the system evolves, at the beginning of each time slot a user's uplink power control (UPC) mechanism is responsible for determining an appropriate uplink transmission power level towards maximizing its overall degree of satisfaction, which is reflected by the corresponding values of its utility. In addition to the maximization goal, mobile user's hardware limitations as well as instantaneous system characteristics mast be taken into account. In this section, the main goals of the proposed PRC mechanism are analyzed and formally defined as a generic optimization problem in a game theoretic framework.

Since, each user in the network aims at the maximization of the expectation of its utility U_i, the corresponding goal of the UPC mechanism can be defined as the maximization of the objective function:

$$\begin{aligned} &\max_{P_i} \ E[U_i(P_i, \overline{P}_{-i})] \\ &s.t. \ \ 0 \leq P_i \leq P_i^{Max} \end{aligned} \tag{6}$$

Two crucial observations need to be made. As the channels of the communication links are assumed to be independent and identically distributed (i.i.d.) the maximization of the average utility in equation (6) is obtained by maximizing the utility at every time slot t. Moreover, due to the users' selfish operation, in terms of pursuing optimal power values towards their individual utility maximization, the overall network uplink power control problem at each time slot can be formulated as a non-cooperative uplink power control game (UPC game).

Let $G = [S, \{A_i\}, \{U_i\}]$ denote the proposed non-cooperative game, where S is the set of users/players and $A_i = [0, P_i^{Max}] \times \Re^N$ is the strategy set of the i^{th} user. Each strategy in A_i can be written as $a_i = (P_i)$. Furthermore, the resulting per time slot non-cooperative game can be expressed as the following maximization problem:

$$\max_{a_i} \ U_i = \max_{P_i} \ U_i(P_i, \overline{P}_{-i}) \quad for \ i = 1, ..., N. \tag{7}$$

under the constraint of non-negative, upper bounded powers.

5 Towards a Nash Equilibrium for the Non-cooperative UPC Game

For the non-cooperative UPC game proposed in the previous section, we adopt Nash Equilibrium approach towards seeking its solution, which is most widely used for game theoretic problems. A Nash Equilibrium point is a set of power vectors, such that no user has the incentive to change its power level, since its utility cannot be further improved by making any individual changes on its value, given the powers of other users.

Definition 1: The power vector $\overline{P}^* = \left(P_1^*, ..., P_N^*\right)$ is a Nash Equilibrium of the UPC game, if for every $i \in S \quad U_i(P_i^*, \overline{P}_{-i}^*) \geq U_i(P_i, \overline{P}_{-i}^*)$ for all $P_i^* \in A_i$.

Prior to the investigation of the existence and uniqueness of an equilibrium in UPC game, we study the properties of RT users' actual throughout and overall utility functions, over their corresponding definition domains. The following lemma determines the form and features of a RT user's actual throughput utility T as a function of the achieved SINR. It is noted that due to space limitation the proofs of the corresponding lemmas are omitted.

Lemma 1: Given: a) an efficiency function $f(\gamma)$, which is a sigmoidal function of γ and has a unique inflection point γ_{Infl}^f, and b) an actual throughput utility function $T(R^*)$, which is a sigmoidal function of R^* and has a unique inflection point R_{Infl}^*, where $R^* \equiv R_{Max}^* \cdot f(\gamma)$, then function $T(\gamma) \equiv T(R_{Max}^* \cdot f(\gamma))$ is also a sigmoidal function of γ ($\gamma \geq 0$) with a unique inflection point $\gamma_{Infl}^{\mathrm{T}}$ for which:

$$\gamma_{Infl}^{\mathrm{T}} : \begin{cases} f^{-1}\left(\dfrac{R_{Infl}^*}{R_{Max}^*}\right) \leq \gamma_{Infl}^{\mathrm{T}} \leq \gamma_{Infl}^f & when \ f^{-1}\left(\dfrac{R_{Infl}^*}{R_{Max}^*}\right) \leq \gamma_{Infl}^f \\ \gamma_{Infl}^f \leq \gamma_{Infl}^{\mathrm{T}} \leq f^{-1}\left(\dfrac{R_{Infl}^*}{R_{Max}^*}\right) & otherwise \end{cases} \tag{8}$$

where $f^{-1}\left(R_{Infl}^* / R_{Max}^*\right)$ is the mapping of function's $T(R^*)$ inflection point at the access of γ.

Lemma 1 states that the composed function of two sigmoidal functions, as in the case of a user's actual throughput utility, is also a sigmiodal function. Henceforth, the inflection point of the new function is laying between the inflection points of the generator functions. As it will be analyzed extensively in the following section, the previous property plays a key role in the attributes of UPC game's equilibrium and therefore, in RT services' QoS requirements satisfaction. We can now characterize the utility maximization of a single user's overall utility when other transmission powers are fixed.

Lemma 2: User's $i \in S$ utility function $U_i(P_i, \overline{P}_{-i})$ is a quasi-concave function of its own power $P_i \in [0, P_i^{Max}]$. Moreover, considering other users' transmission powers fixed, $U_i(P_i, \overline{P}_{-i})$ has a unique global maximization point:

$$P_i^* = \min\{\frac{\gamma_i^* R_{Max,i}^* I_{-i}(\overline{P}_{-i})}{WG_i}, P_i^{Max}\} \tag{9}$$

pursuing a unique target SINR value γ_i^*, which is the (positive) solution to $\frac{\partial T_i(\gamma_i)}{\partial \gamma_i} \cdot \gamma_i - T_i(\gamma_i) = 0$.

Equation (9) indicates that if a user's maximum transmission power is not sufficient for reaching the targeted γ_i^*, due to its potentially bad channel conditions, then the best policy is to transmit with maximum power. Moreover, lemma 2 reveals that users' goal to maximize their utility-based performance can be translated in a constant attempt of meeting specific SINRs (at the base station), which eventually leads to an SINR-balanced network.

The following proposition asserts the existence and the uniqueness of a Nash Equilibrium point of the proposed uplink power control game and hence, determines nodes' transmission power vector at equilibrium.

Proposition 1: The Nash Equilibrium of the non-cooperative game (7) is given by P_i^*, where $P_i^* = \min\{\frac{\gamma_i^* R_{Max,i}^* I_{-i}(\overline{P}_{-i})}{WG_i}, P_i^{Max}\}$, $\forall i \in S$. Here, γ_i^* results from the unique positive solution of equation $\left(\partial T_i(\gamma_i)/\partial \gamma_i \cdot \gamma_i\right) - T_i(\gamma_i) = 0$. Furthermore, the equilibrium exists and is unique.

Proof: In accordance to lemma 2, the power level that corresponds to the maximization of user's *i* utility-based performance, given other users' power levels, equals to the power level that maximizes the utility in (5) when setting as a designed transmission rate $R_i^F = R_{Max,i}^*$ and thus, is given by $P_i^* = \min\{\gamma_i^* R_{Max,i}^* I_{-i}(\overline{P}_{-i}) / WG_i, P_i^{Max}\}$, where γ_i^* is the unique positive solution of $\left(\partial T_i(\gamma_i)/\partial \gamma_i \cdot \gamma_i\right) - T_i(\gamma_i) = 0$.

So far, we have shown that at Nash Equilibrium (if exists) each user's transmission power is pursuing a targeted SINR value which depends not only on the modulation, coding, and packet size being used (expressed through its appropriate efficiency function f_i) but also, is affected by RT user's QoS requirements (reflected by its actual throughput utility).

Following [4], [5], the existence of a Nash Equilibrium of the game in (7) can be shown via the quasi-concavity of each node's utility function in its own power. As shown in lemma 2, $U_i(P_i, \overline{P}_{-i})$ is a quasi-concave function in $P_i \in [0, P_i^{Max}]$, and hence, Nash Equilibrium always exists.

Moreover, for an S-shaped actual throughput utility function T_i, $\left(\partial T_i(\gamma_i)/\partial\gamma_i \cdot \gamma_i\right) - T_i(\gamma_i) = 0$ has a unique solution γ_i^*, which is the unique global maximizer of user's i utility function. Because of the uniqueness of γ_i^* and the one-to-one relationship between uplink transmission power and corresponding SINR, the above Nash Equilibrium is unique. ■

Concluding this section, we present an iterative and distributed uplink power control algorithm for reaching the Nash Equilibrium for the UPC game G at every time slot t.

UPC Algorithm

(1) At the beginning of time slot t, user i transmits with maximum power. Set $k=0$ and hence, $P_i^{*(0)} = P_i^{Max} \ \forall i \in S$.

(2) Given the uplink transmission powers of other users, which is implicitly reported by the base station when broadcasts its overall interference $I^{(k)}(\overline{P}^{(k)})$, the user computes $I_{-i}^{(k)}(\overline{P}_{-i}^{(k)})$ and refines its power level, i.e. computes $P_i^{*(k+1)}$ in accordance to (9).

(3) If the powers have converged (i.e. $\left|P_i^{*(k+1)} - P_i^{*(k)}\right| \le 10^{-5}$) then stop.

(4) Set $k=k+1$, go to step 2.

6 On Real–Time Users QoS Performance at Equilibrium: Towards a Self-optimization Approach

In this section, the properties of Nash Equilibrium in the proposed uplink power control game are analyzed and discussed, in terms of users' power levels at equilibrium, the role of their physical limitations, their actual throughput and overall utility-based performance. It is shown, that by giving users the autonomicity of controlling their modulation and coding schemes, as well as their transmission power levels, their services' QoS performance optimization can be successfully self-controlled.

Following the pure optimization analysis of the previous section, we initially point out that any γ_i^* that corresponds to the maximization of a RT user's i utility U_i, must be greater than $\gamma_{Infl,i}^{\mathrm{T}}$, that is such an γ_i^* must be in the interval over which $T_i(\gamma_i)$ is concave i.e.

$$\gamma_{Infl,i}^{\mathrm{T}} < \gamma_i^* \tag{10}$$

Additionally, in accordance to lemma 1, the inflection point of $T_i(\gamma_i)$, as a function of γ_i, lays always between the values of the inflection points of its generator functions (i.e. $f_i(\gamma_i)$ and $T_i(R_i^*)$ as a function of R_i^*). Therefore, from (8) and (10), it is apparent that the following property regarding γ_i^* holds:

$$\gamma_i^* = \begin{cases} f_i^{-1}\left(\frac{R_{Infl,i}^*}{R_{Max}^*}\right) < \gamma_i^* & when\ f_i^{-1}\left(\frac{R_{Infl,i}^*}{R_{Max,i}^*}\right) \le \gamma_{Infl,i}^f \\ \gamma_{Infl,i}^f < \gamma_i^* & otherwise \end{cases} \tag{11}$$

Relying on the previous statements, the following proposition determines the influence of an existing modulation and coding scheme on a RT user's i service performance.

Proposition 2: If for real-time user $i \in S$, $f_i^{-1}\left(R_{Infl,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f$; then at UPC game's Nash Equilibrium, where $P_i = P_i^*$ and hence $\gamma_i = \gamma_i^*$, the achieved actual throughput rate R_i^* is always greater than its service minimum acceptable actual throughput $R_{Min,i}^* = R_{T,i}^* - EL_i$ i.e.

$$\text{if } f_i^{-1}\left(R_{Infl,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f \text{ then } R_i^* > R_{Min,i}^*$$

Proof: When $f_i^{-1}\left(R_{Infl,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f$ then according to (11) $f_i^{-1}\left(R_{Infl,i}^*/R_{Max}^*\right) < \gamma_i^*$ and thus, $R_{Infl,i}^* < R_{Max}^* \cdot f_i\left(\gamma_i^*\right)$ since f_i is a continuous increasing function of γ_i when $\gamma_i \ge 0$. Moreover, since from definition, a RT user's utility inflection point is set as $R_{Infl,i}^* = R_{Min,i}^*$, with respect to it service QoS requirements, and due to the fact that $R_i^* \equiv R_{Max,i}^* \cdot f_i(\gamma_i)$, we can conclude that:

$$\text{if } f_i^{-1}\left(R_{Infl,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f \text{ then } R_i^* > R_{Min,i}^* \tag{12}$$

and hence,

$$\text{if } f_i^{-1}\left(R_{Min,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f \text{ then } R_i^* > R_{T,i}^* - EL_i \tag{13}$$

when P_i^{Max} is large enough for reaching γ_i^*. ■

The previous proposition identifies a strong correlation among RT users' QoS requirements, system's modulation and coding schemes (efficiency functions) and their power limitations. Moreover, it asserts that if user's *i*, P_i^{Max} is large enough for reaching γ_i^*, then at games' equilibrium, not only its overall utility-based performance is maximized but also its service actual throughput expectations are simultaneously fulfilled, if $f_i^{-1}\left(R_{Min,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f$.

From a RT users' perspective, given a specific efficiency function and its corresponding inflection point (i.e. for a certain modulation scheme), we can point out that:

A. If $f_i^{-1}\left(R_{Min,i}^*/R_{Max,i}^*\right) \le \gamma_{Infl,i}^f$, then it is assured that its actual transmission rate will always be within the bounds determined by its target actual uplink rate $R_{T,i}^*$ and its elasticity factor EF_i.

B. The greater a RT user's service elasticity factor, the higher the probability of satisfying its actual throughput QoS requirements, since f_i^{-1} is an increasing function of the ratio $R^*_{Min,i} \big/ R^*_{Max,i} = (R^*_{T,i} - EL_i) \big/ (R^*_{T,i} + EL_i)$.

C. On the other hand, the stricter its QoS requirements are (i.e. when setting small values for its elasticity factor which indicates that only small deviations from its target actual throughput are allowed), the harder its throughput prerequisites are fulfilled. Moreover, if user's i, EL_i value is such that $\gamma^f_{Infl,i} < f_i^{-1}\left(R^*_{Infl,i} \big/ R^*_{Max,i}\right)$, then, even when its overall utility maximization is achieved, its actual throughput may potentially be smaller than $R^*_{Min,i}$.

In accordance to the previous analysis, we argue on using the boolean criteria of proposition 2 as a decision indicator on a RT user's initial acceptance in the system (i.e. call admission control criteria).

From the system's perspective, assuming the arrival of a new RT user with fixed QoS requirements (predefined proper actual throughput utility T_i and its corresponding parameters $R^*_{T,i}$, EL_i), its modulation and coding scheme, reflected to function f_i, should be chosen such that (13) is satisfied. From (13), we can observe that the stricter a user's QoS requirements are (i.e. small values for its elasticity factor) the higher the value of the inflection point of the chosen f_i should be. Therefore, as Fig.1 illustrates, a poorer modulation scheme, with respect to the achievable maximum transmission rate under that scheme, should be chosen in such case. Thus, the inevitable tradeoff between users' overall throughput performance and RT users' strict QoS requirements in channel aware resource allocation mechanisms comes into sight.

The previous controllable criteria favor the design and implementation of autonomous users' mechanisms that will not only control their power levels towards maximizing their overall utility-based performance (by adopting the proposed UPC mechanism), but will also control their services' performance experience by: a) constructing their proper actual throughput utility functions and b) adjusting users' modulation and coding schemes in accordance to predefined criteria (e.g. criteria introduced in proposition 2), towards optimizing their QoS prerequisites satisfaction.

7 Numerical Results and Discussions

In this section, we provide some initial numerical results illustrating the operation and features of the proposed framework and UPC algorithm. Throughout our study we consider the uplink of a single cell time-slotted CDMA system, supporting $N=10$ continuously backlogged real-time users. Moreover, each simulation lasts 10000 time slots, while we set P_i^{Max} =2 Watt, $W = 10^6$ Hz and $I_0 = 5*10^{-16}$. We model users' path gains as: $G_i = K_i \big/ d_i^a$, where d_i is the distance of user i from the base station, a is the distance loss exponent ($a=4$), and K_i is a log-normal distributed random variable with mean 0 and variance $\sigma^2 = 8(dB)$, representing the shadowing effect. Furthermore, for

each user i, $d_i=d_{i-1}+250\ (m)$ for $i=2,..,10$, where $d_0=300\ (m)$. In this way, we emulate a scenario where users' average channel conditions are worse as their ID value (i.e. $i=1,\ldots,10$) increases.

Two types of real-time users are considered. Users *1* to *5* require actual uplink rate $R^*_{T,i}$= *64 (Kbps)*, while users *6* to *10*, $R^*_{T,i}$=*128 (Kbps)*. For both types of users the elasticity factor is set EL_i = *10 (Kbps)* for i= *1,…, N*. Fig.3, Fig.4 and Fig.5 illustrate for each user its average utility-based performance, the corresponding average power consumption and its achievable average actual throughput, under two different scenarios with respect to the characteristics of the examined efficiency function $f_i(\gamma_i)=(1-e^{-\gamma_i})^M \quad \forall i=1,...N$. In the first scenario (black columns), where M = *100,* for both types of users $f_i^{-1}\left(R^*_{Infl,i}/R^*_{Max,i}\right)\geq\gamma^f_{Infl,i}\forall i=1,...,N$, while in the second scenario (grey columns), where M = *500,* we have $f_i^{-1}\left(R^*_{Infl,i}/R^*_{Max,i}\right)\leq\gamma^f_{Infl,i}\forall i=1,...,N$.

The results indicate that as users' average channel conditions become worse, their overall utility based performance decreases (Fig.3), while their power consumption increases (Fig.4). On the other hand, their actual uplink data rates are in line with their predefined actual uplink rates R^*_T, especially when M=*500*, which clearly indicates not only the proposed algorithm's efficiency but also the proper functionality of user's actual throughput utilities (T_i) on satisfying real-time users' QoS requirements.

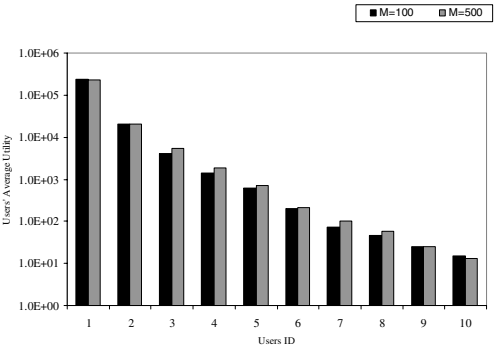


Fig. 3. Users' average utility-based performance

Moreover, the results show that the selection of an appropriate modulation scheme can result in RT users' QoS expectations satisfaction. Specifically, we can observe that when M=*500*, all users' actual uplink data rates are within their expected bounds (i.e. $R^*_{Min,i}< E[R^*_i]<R^*_{Max,i} \quad \forall i\in S$ due to the fact that $R^*_{Min,i}< R^*_i < R^*_{Max,i}$ for each time slot t), even for the second type of users (note that in Fig.5 red ($R^*_{Min,i}$) and green ($R^*_{T,i}$) lines represent the respective bounds), while the latter

observation does not hold when *M=100*. These results further demonstrate each user's ability to control and adapt its modulation scheme towards the efficient satisfaction of the corresponding QoS requirements. Moreover, after the initial construction of its actual throughput utility in accordance to its service QoS requirements, a user's modulation scheme selection should always be in line with the criteria in proposition 2, in order not only its overall performance to be optimized but also its QoS constraint to be achieved.

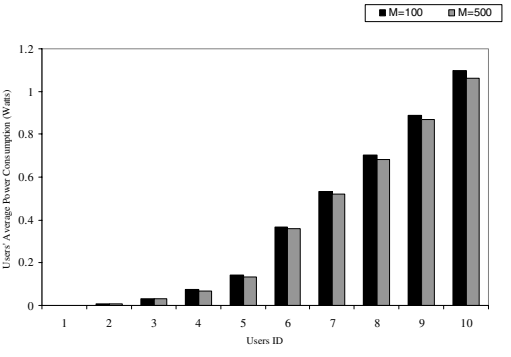


Fig. 4. Users' average power consumption

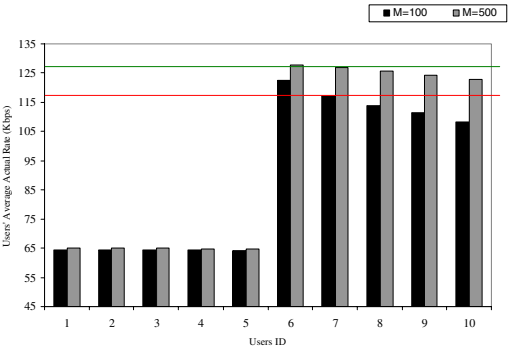


Fig. 5. Users' achieved average uplink actual throughput

8 Conclusion

In this paper we considered the issue of efficient power allocation in the uplink of CDMA wireless networks, emphasizing on the support of real-time services' QoS prerequisites. The corresponding problem has been formulated as a non-cooperative game and solved through a low-complexity algorithm, which reaches game's unique Nash Equilibrium point, taking into account the imposed physical limitations. The existence and uniqueness of Nash Equilibrium point of our proposed game was proven and thus, the properties of equilibrium as well as the tradeoffs between users'

overall throughput performance and real-time services strict QoS requirements were revealed.

Generalizing this work, we are currently studying a concrete uplink recourse allocation utility-based framework, which will accommodate both real-time and non-real time services using the appropriate utility functions. Moreover, the introduced framework and proposed approach provides a first step towards the realization of autonomic wireless networks, where user self-adaptive mechanisms allow the control and facilitate the satisfaction of their QoS constraints.

Acknowledgements. This work has been partially supported by EC EFIPSANS project (INFSO-ICT-215549).

References

[1] Lee, J.-W.R., Mazumdar, R., Shroff, N.B.: Joint resource allocation and base-station assignment for the downlink in CDMA networks. IEEE/ACM Trans. Netw. 14(1) (February 2006)

[2] Kastrinogiannis, T., Papavassiliou, S., Kastrinogiannis, K., Soulios, D.: A Utility-Based Resource Allocation Approach for the Downlink in CDMA Wireless networks with Multimedia Services. In: Proc. of the 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (September 2007)

[3] Meshkati, F., Poor, H.V., Schwartz, S.C., Balan, R.: Energy-Efficient Resource Allocation in Wireless Networks with Quality-of-Service Constraints. Princeton University Press, Princeton (preprint 2005)

[4] Meshkati, F., Poor, H.V., Schwartz, S.C.: A Non-Cooperative Power Control Game in Delay-Constrained Multiple-Access Networks. In: Proc. of the IEEE International Symp. on Info. Theory (ISIT), Adelaide, Australia (September 2005)

[5] Meshkati, F., Goldsmith, A.J., Poor, H.V., Schwartz, S.C.: A Game-Theoretic Approach to Energy-Efficient Modulation in CDMA Networks with Delay Constraints. In: Proc. of IEEE Radio and Wireless Symposium, pp. 11–14 (January 2007)

[6] Duan, X., Niu, Z., Zheng, J.: A Dynamic Utility-Based Radio Resource Management Scheme for Mobile Multimedia DSCDMA Systems. In: Proc. IEEE Global Telecom. Conf. 2002 (GLOBECOM 2002), Taipei, Taiwan (November 2002)

[7] Kastrinogiannis, T., Papavassiliou, S.: Probabilistic Short-term Delay and Throughput Requirements of Multimedia Services in High Throughput Wireless Networks. In: Proc. of IEEE Sarnoff Symposium on Advances in Wired and Wireless Communications (April 2007)

[8] Shenker, S.: Fundamental design issues for the future Internet. IEEE J. Selected Areas Commun. 13, 1176–1188 (1995)

Adaptive Priority Based Distributed Dynamic Channel Assignment for Multi-radio Wireless Mesh Networks

Tope R. Kareem[1,2], Karel Matthee[1], H. Anthony Chan[2], and Ntsibane Ntlatlapa[1]

[1] Meraka Institute, CSIR, Pretoria, South Africa
tkareem@csir.co.za,kmatthee@csir.co.za, nntlatlapa@csir.co.za
[2] Department of Electrical Engineering, University of Cape Town
h.a.chan@ieee.org

Abstract. This paper investigates the challenges involve in designing a dynamic channel assignment (DCA) scheme for wireless mesh networks, particularly for multi-radio systems.

It motivates the need for fast switching and process coordination modules to be incorporated in DCA algorithm for multi-radio systems. The design strategy is based on a reinterpretation of an adaptive priority mechanism as an iterative algorithm that recursively allocate a set of channels to radios in a fair and efficient manner in order to minimise interference and maximise throughputs. The algorithm, called Adaptive Priority Multi-Radio Channel Assignment (APMCA) is tested for overall performance to assess the effectiveness by determining its overall computational complexity.

The combined advantages of fast switching time and process coordination modules make the APMCA a useful candidate towards automating the channel assignment method in multi-radio wireless mesh network planning and design.

Keywords: Wireless Mesh Networks, Multi-radio, Channel Assignment.

1 Introduction

One of the strategies of improving system throughputs and network capacity in Wireless Mesh Networks (WMN) is by coordinated use of multiple radios. Multiple radios wireless mesh separates client access and wireless backhaul for the forwarding of mesh traffic. In this type of mesh, each node has a dedicated radio for backhaul connectivity operating at different frequency with performance similar to switched, wired connections. A downside of deploying a multi-radio system is the *herculean* task of a network administrator to statically configure all the available non-overlapped radio channels. Even if a network administrator painstakingly took up the challenge to assign radio channels statically to all radios in a community based wireless mesh network, we could not be sure of having a network plan that minimizes interference with other radios in the same network and other radios in the neighbouring networks. It is

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 321–332, 2008.
© Springer-Verlag Berlin Heidelberg 2008

therefore evident that a new intelligent method of assigning channels to radios in a multi-radio environment is required.

Previous investigation conducted by Kyasanur [1] classically divided channel assignment into three categories viz: static, dynamic and hybrid. While static channel assignment is used for applications that can tolerate large interface switching delay, dynamic channel assignment (DCA) is suitable for applications with limited available bandwidth and unpredictable variable bit rate traffic. A careful review of existing channel assignment (CA) algorithms for multi-radio (M-R) systems reveals two key design challenges. Firstly, there is the need for fast switching module for switching of radio channels among the multiple wireless radios installed on each node. Secondly, there is also the need for a process coordination module for network monitoring, supervision and control. According to the author of [2], these key challenges, if properly implemented would subsequently lower the number and the cost of mesh nodes needed to deploy any community-based wireless mesh network.

In another [3] selected review of literature on DCA a breadth-first search channel assignment (BFS-CA) algorithm was analysed. The algorithm takes as input, the interference estimates from the mesh routers and a multi-radio conflict graph (MCG). The interference estimate is used to select the default channel (i.e., the channel with the least interference) while MCG is used to model the non-default radios in the mesh network. Any radio assigned to a default channel is by implication a default radio. A multi-radio system unlike a single radio system considers all its radio independent, and therefore does not have a dedicated default radio for each node or a group of nodes. However, this technique of allocating a radio as default radio for every node in the network would further increase the process coordination requirements of the algorithm, thereby increasing its complexity.

In the same vein, the work of H. Skalli *et al*., as published in [4] proposed a similar algorithm called *MesTic*. The input parameter of this algorithm includes (as in [3]) , a traffic matrix in addition to the MCG, connectivity graph, the number of radio at every node and the number of non-overlapping channels. Both algorithms described in [3] and [4] use ranking technique to assign channels to radios. Although this technique is simple and easy to comprehend, a rank function requires a full description of its underlying parameters and their interdependency. Moreover, in this particular instance (i.e., channel assignment for multi-radio wireless mesh networks), there is need to specify in the algorithm whether a node rank or channel rank is referenced prior to the process of channel assignment.

In [5], a joint distributed channel assignment and routing algorithm is developed. The algorithm utilises neighbour discovery and routing protocol to allow each node to connect with its neighbour. Neighbour discovery protocol uses an ADVERTISE packet that contains the cost of reaching the gateway node. This cost in turn depends on residual bandwidth require to achieve load balancing in the network. Conversely, the aggregate load on each virtual link also depends on a given routing algorithm. It is therefore possible to infer that the interdependency of channel algorithm on specific class of routing algorithm (also known as path selection algorithm) will not promote interoperability between devices from different vendors.

Our proposed dynamic channel algorithm will not be tied to a specific routing algorithm to ensure baseline interoperability. Also, it will not differentiate the total number of radio interfaces on each node into fixed and switchable interfaces. In

addition, the number of available non-overlapped channels is expected to be far greater than the number of radio interfaces installed on each wireless node. Therefore each wireless node will need to be equipped with channel switching functionality in order to fully exploit the aggregate bandwidth available in the radio spectrum provisioned by the standard. Furthermore, since network links do not all have the same importance in carrying traffic, our algorithm should be able to identify links having a greater capability to carry traffic and therefore prioritised such links.

A solution like this, according to [1] requires fine-grained synchronization and thus will be difficult to implement without modifying the existing 802.11 MAC protocol. However, we relax the synchronization constraints by implementing two versions of the algorithm. The version with the fast switching module is implemented in distributive manner among all the mesh access point (MAP) and mesh point (MP) in the network as shown in Fig. 2, while the other version is centralized and has the process coordination module installed only on a dedicated management information base (MIB) server node. This process coordination module is responsible for keeping track and managing the interface switching, initiating the functional call for routing algorithm, monitoring the discontinuity of traffic flow between every communicating node pair, and setting the value of the *ReThreshold* attribute that defines the remaining length of frame to be transmitted before calling the routing algorithm.

Consistent with much of the literature radio assignment problem, this paper presents theoretical bounds on the number of radio channels, as well as some complexity analysis (NP-completeness) of the problem. It then proposes a multi-channel multiple radio wireless mesh network architecture. In this architecture, both the MAP and MP are running fast switching module version of dynamic channel assignment and a centralized dedicated server node runs the management protocol. Next, the proposed algorithm is discussed with explicit detail the fast switching and process coordination modules. An analysis to compute the order of overall complexity is presented. The computation allows us to evaluate the performance of the proposed scheme; and finally a concise summary and future work conclude the paper.

2 Problem Definition and Description

We consider the problem of assigning multiple channels to multiple radios so that each radio receives at most one channel. The wireless radios installed on each node have preferences (as stated in I) over the available channels, thus, the allocation mechanism does take the profile of the preferences as part of its inputs. An important assumption is that the number of available channels is more than the number of wireless radio installed on each node; and that the network traffic and conditions may vary over time.

Let $G(V, E, K)$ be a connected network graph where $V = (M_p, M)$ represent a set of mesh nodes differentiated to mesh access point and mesh point respectively; and $E(u^i, v^j)$ represent a set of links. Let K be the number of wireless radios installed on each node V, and N be the number of available non-overlapped channels, denoted by $\{1, 2, , , , , N\}$.

The DCA considered here is closely related to random assignment problem published by Akshay-Kumar *et al*., [6]. It is defined as probability distribution over static assignment, and the corresponding convex combination of permutation matrices is a stochastic matrix, whose (i, $j)^{th}$ entry represents the probability with which the wireless radios i receives channel *j*. The use of the word static in this context implies deterministic.

Then, given a dynamic channel assignment matrix *P*, we let P_i be the i^{th} row, which represents the assignment of radio *i* in this dynamic assignment. If we let *R* be the set of all possible dynamic assignments in a given network, we can therefore define the mechanism of assigning channels dynamically simply as the mapping from N^n to *R*.

A solution to the problem is obtained by selecting an assignment relation (as in Fig. 1) that maximises capacity and minimises interferences; while also satisfying some efficiency and fairness properties. Efficiency is measured in terms of network throughput and delay, while fairness is measured in terms of fairness ratio, which bounds the ratio of maximum and minimum throughputs values.

Formally, let us define the Kth wireless radio over two DCAs, *p* and *q,* and given that K is indifferent between *p* and *q* (fairness property), then

$$p \approx q \Leftrightarrow \sum_{k:k \geq ij} p_{ik} = \sum_{k:k \geq ij} q_{ik} \qquad (1)$$

$$\forall j \in N$$

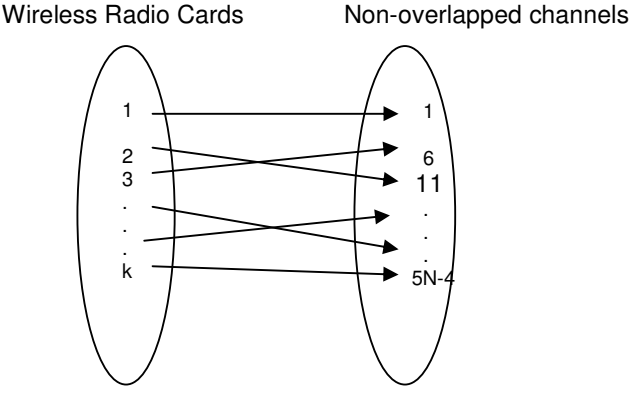


Fig. 1. Allocation of wireless radio cards to channels in a multi-radio multiple channels mesh network is modelled as Injective (and not Bijective) function since the number of radios is less than the number of independent channels

3 Architecture and System Design

The proposed multi-channel wireless mesh network architecture, shown in Fig.2, consists of dedicated infrastructure devices known as mesh point (MP) and mesh

access point (MAP). Mesh access point is a special type of mesh point which provides access point (AP) services in addition to mesh services. Users' devices (not shown in Fig.2) support mesh services and associate with mesh APs to gain access to the mesh network. These mesh nodes are equipped with two or more wireless radio cards and together, they form ad hoc network among themselves to relay traffic to and from end-user devices. In addition, the wireless radios are running fast switching applications (this is elaborated in Section IV) that allow them to support channel switching.

A dedicated centralized management information base (MIB) server is connected to the gateway. MIB server node runs the interface management protocol located within the process coordination module, and is responsible for keeping track and managing the interface switching.

Together, the devices are configured in a multipoint-to-multipoint architecture for internet connectivity. Internet connection to multipoint-to-multipoint mesh network does not come from a wired router but through the backhaul mesh via the gateway.

As indicated in Section I, the two versions of the proposed dynamic channel assignment algorithms are implemented in the network. The version with fast switching module is implemented in the MAP and MPs, while the version with process coordination module resides in the MIB server node.

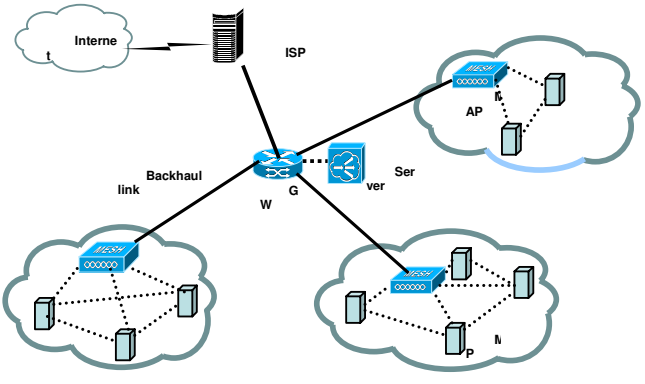


Fig. 2. A community-based multipoint-to-multipoint mesh network topology running fast switching and process coordination modules

4 Dynamic Channel Assignment

The proposed dynamic channel assignment algorithm called APMCA (Adaptive Priority Multi-radio channel assignment) is designed for a simple network structure where all the mesh nodes are equipped with equal number of radios, and a pre-defined number of available non-overlapped channels as shown in Fig.3.

The algorithm uses an iterative application of adaptive priority algorithm that terminates in (at most) N phases, where N is total number of non-overlapped channels available in the network. Adaptive priority implies that it is possible for the mesh nodes to reallocate the radio channels after each successful packet transmission from

source to destination nodes subject to the channel constraints as defined in the input sequence S_i.

Each radio interface receives S_i as an input sequence which is characterised by a list of 4 elements of non-negative numbers S_i = (*NodeName*, *Non-overlapped-Channels*, *NodeRadioLabel*, *AdjList*).

The "*NodeName"* is an identifier that denotes a uniquely assigned node name for each mesh nodes in the network. "*Non-overlappedChannels"* denotes the number of allowable co-located channels with centre frequencies of 5MHZ apart, the channels are 22MHZ wide, and the number of channels between successive channels is at least five apart. The *NodeRadioLabel* is an identifier that provides attributes to radios in each node. Lastly, *AdjList* is introduced as an identifier that defines *a* set of 2-tuples comprising the spatial channel re-use ratio and an estimate of co- channel interference in the network.

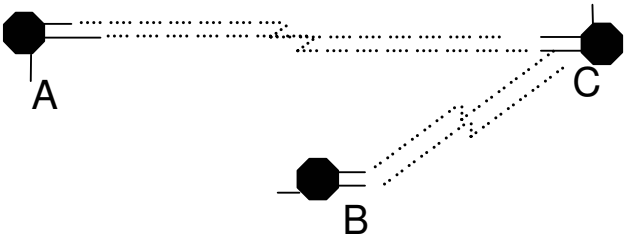


Fig. 3. Illustrates a network of three radios - four channel systems deployed in a wireless mesh network. Two of the three radios are dedicated backhaul links and the third radio is for configured for access network.

At start-up, every interface is randomly assigned a radio channel such that no two radios within the same communication range (as defined by channel reuse principle) are assigned the same channel, except for a pair of nodes communicating with each other through a common communication channel. This is done to eliminate selection biases that may degrade the network performance.

A pair of nodes that wish to communicate must first share a common communication channel that is used to set up a virtual link. If such common communication channel has already been established by default, then the algorithm proceeds to test the constraints as listed in *AdjList,* otherwise, a fast switching module is enabled on the source node. The mechanism of fast switching enables each wireless radio installed on the source node to randomly switch to channels available on the destination node until at least one communication channel is established. *AdjList* is a set of 2-tuples comprising the spatial channel re-use and interference estimation. The channel reuse factor depends strongly on the environmental characteristics, primarily, path loss and slow fading, while the estimation of interference depends mainly on the distance between the nodes.

A positive attempt towards the characterisation of spatial channel reuse in multi-radio WMN begins by using the concept of a simple classical triangular mesh (as given in [10]) of $L * L$ square area, and a frequency reuse distance D. Given that equal number of radios "K" is installed on each mesh node, then we have:

$$K = \frac{2L^2}{D^2\sqrt{3}} \tag{2}$$

Assuming that one MAP manages several MPs as stated in section II, and each MAP is fairly located at the centre point *R*, then the channel reuse ratio is calculated thus:

$$D/R \geq \zeta \tag{3}$$

where ζ is the parameter that defines the necessary and sufficient condition for good spatial reuse for the triangular mesh.

Similarly, interference estimation in a multi-radio multiple channel environments is also characterised by using a combination of heuristic and measurement-based technique. A modified version of the heuristics developed in [7] and [8] that is based on the distance between nodes is considered in this design, and a measurement based technique presented in [9] is extended to a multi-radio environment. We assume a worst case scenario in which each radio on each node is connected to another radio on another node thereby resulting in the emanation of multiple simultaneous active links from a single node. With this, the problem is reduced to that of estimating interference among multiple links in a wireless mesh network; and according to [5], this information is considered necessary for the design of an optimal channel assignment.

A node *A* that wishes to communicate with a node *B* must first sense the channel for the availability of a common communication channel. If it notices that either the channel is in use or there is no common channel available for the intended communication, it then backtracks (the default mechanism in 802.11 Protocol). A fast switching module rather than the backtracking algorithm is called as explained in sub section *A*.

In a situation where all available channels assigned to the radios on the same node are currently in use, a reuse distance is computed as discussed above. The overall purpose of these processes is to search for a free channel to use for communication between the nodes.

Algorithm APMCA (***A**daptive **P**riority **M**ulti-**R**adio **C**hannel **A**lgorithm)* To find an efficient and fair channel assignment P of multiple radios K to multiple channels N in a wireless mesh network G(V, E, K) that maximizes capacity and minimises interference. Let H_j be a target graph, T is define as the interference threshold, and V_k denotes each radio installed on each node in the network.

Step 0. [Initialise] H_j <– G (V, E, K); $P_i \equiv 0$; $P_i \in H_j$;
$|N| < K$ for all $_{i,j} \geq 1$, T = 0.65,
ζ = 1.16.

Step 1. [Iterate] testIfCommExist V_k =
dropWhile ($V_k \geq 2$) [V_k|V_k< – [1..j]]

Step 2. [Channel assignment] for pair of communicating nodes

```
                    u, v ∈ V ∃ Kij where Ki ∈ u
                    and Kj ∈ v;
                    pickRadio K rnd = K !! rnd;
                assignChannelToRadio Ki Ni result =
            M.insertWith (++) Ni [Ki ] result;
                case intersect Ki, Kj =
                 filter (\Cn − >any((==)Cn)Ki)Kj of
               {assign − > ( Cn <− [Ki, Kj] );
              nonAssign − >fastSwitchingSame f}
```

Step 3.[fast Switching]

```
                    fastSwitchingSame f
                    lookup 'Ki ' [('Ki-1', Ni), ..'Kn' Nj)];
            if intersect Ni Nj = [V| V< − Ni, V 'elem' Nj]
                then
                    swap Ki ; Kj;
                    else
            fastSwitchingNeighbour fn
            lookup 'Ki' map (*D) [('Ki-1', Ni), ..'Kn' Nj)];
            if intersect Ni Nj = [V| V < − Ni, V 'elem' Nj]
                    then
                interferenceEsti x y z  --function call;
               else
              reuseEsti k l r                --function call;
```

Step 4. [update process coordination server]

```
type State = (Integer, Bool)
update :: State − > State
meshAccessPoint :: [a]  − > (a  − >a)  − >[a]
meshAccessPoint (meshPoint, K) = K+1 ++ map (*n) [meshPoint];
 meshPoint :: a  − >[a]
 meshPoint (radioK : radioKs) = map (t+1) [radioK];
 update = [y | y < − [meshAccessPoint(K)t + 1] !! all;
  meshPoint(n),  filter (\y −> any
     ((==)meshPoint(K))meshPoint(t)meshPoint(t+1));
```

Step 5. [Interference estimation]

```
                    interferenceEsti x y z =
```

$$\frac{\left(\beta f^{x}yz + \Omega f^{y}xz + \Pi f^{z}yx\right)}{f_x + f_y + f_z};$$

```
                    -- where β, Ω, and Π are const.
                    values that are environmental and
                     hardware- dependent.
                     if interferenceEsti < T;
                        then
```

```
                                    processUpdate x y z;
                                    else
                                       reuseEsti k l r ;

Step 6. [Channel reuse estimation]
                          reuseEsti k l r  =
                          let reuseDistance = k / (0.931 * √l) ;
                           in reuseDistance / r ;
                           if reuseEsti ≤ 1.16;
                                then
                                   fastSwitchingNeigbour f;
                                 else
                                   P_i = P_i + 1;
```

4.1 Complexity Analysis of APMCA

Step 0 of the algorithm APMCA requires $O(m*n)$ operations to initialise each of K number of radios installed on V number of nodes.

Step 1, the iteration step, essentially requires $O(m)$ operations to determine if there are more radios not yet randomly assigned a channel.

Step 2 is executed exactly n (m-1) times. Each execution of step 2 requires that the APMCA search through the list of assigned radios to find a pair of communicating node whose radio share a common channel. This effort requires $O(n*(m-1))$ operations, where n (m-1) denote the number of available radios m on a receiving node n.

Step 3 involves four steps divided into two categories (Same node and Neighbourhood nodes). Searching process in both the "same node and Neighbourhood node" requires $\beta * O(\ln(m))$ operations (where β is a constant define differently for a case of "same node" and "neighbourhood node") taking into consideration that the data in the look up tables for both cases are already sorted. Furthermore, the process of swapping of radio K_i and K_j also requires $O(1)$ operations, and on the overall, the complexity of step 3 is bounded from above as $O(\log(m))$.

Step 4 primarily involves updating a dedicated server at every time t; and for every successful transmission from a radio K, this process requires $O(m)$ operations. For each of the notification sent to MAP, a report is sent to the server to notify the server of any changes in the state of the network. At each successive state, a 4-tuple constraint $S_{i,}$ is tested and this also requires $O(m)$ operations. Since none of the other substeps of step 4 requires more than $O(m)$ operations, the complexity of step 4 is therefore bounded by $O(m)$ using the theorem:

$$O(m) + O(m) = O(m) \tag{4}$$

as in [11].

Step 5 and Step 6 are functional calls. Step 5 comprises two loops whose running time is proportional to the square of the number of radios on a pair of communicating nodes. In addition, a computation of the ratio $\frac{W}{U}$ requires logarithmic operations, while the test of validity of ratio $\frac{W}{U}$ has a linear running time.

In summary, the complexity of step 5 is therefore bounded as $O(m^2)$. Similarly, *Step 6* has two linear operations for measurement of *l* and *r*. A computation of *reuseDistance* also requires a linear combination of quadratic and logarithmic running times. In addition to this, the last substep in step 6 requires a combination of linear and logarithm operations. We can therefore conclude that the complexity of step 6 is also bounded by $O(m^2 * \log(m))$.

4.2 Proof of Correctness

The first step is to show that all radios are randomly assigned to at most one channel.

The next step is to conduct a randomization test only for a pair of communicating nodes.

In order that to verify the above two steps, we start by denoting the number of radios installed on a pair of communicating nodes as K_1 and K_2. If we define the number of ways of assigning the non-overlapped channels *N* as *W*, then *W* is represented thus:

$$W = \frac{N}{K_1! K_2!} \tag{5}$$

The third step is to determine how many of these ways *W* of assigning the channels to radio satisfy both the local and global constraints. This number is denoted as "assignment" $P = \{P_1, P_2, P_3 \ldots P_n\}$.

The final step is to test the value of the interference estimated against the allowable threshold value.

The above four steps simply shows that a unique solution P_n exist for every pair of communicating radios in the multi-radio wireless mesh network.

4.3 Overall APMCA Complexity

The complexity analysis shown in subsection *A* which is based on the order-of-magnitude analysis and not on the coded implementation of the algorithm shows that the overall complexity is given thus:

$$O(m*n)+O(m)+O(\log(m)+O(m)+O(m^2)) \\ +O(m^2*\log(m)) \tag{6}$$

since K >> N, then we can conclude that $O(n) \in O(m)$ and subsequently, $O(m*n) \approx O(m)$.

Therefore, the entire complexity of algorithm APMCA computed from equation 6 is $O(m^2)$.

5 Conclusion and Future Work

This paper addresses the need for the addition of fast switching and process coordination modules to the design of channel assignment scheme for multi-radio wireless mesh networks.

Our proposed design is aimed at maximising the network capacity and minimizing the interference within the same node and among the nodes in the neighbourhood. The study commences with architectural and system design consisting of dedicated mesh routers differentiated into mesh point (MP) and mesh access point (MAP) equipped with two or more wireless cards, and a centralised management information base (MIB) server. These infrastructural devices (mesh routers and MIB), respectively host the two different versions of our proposed algorithm. The algorithm uses an iterative application of adaptive priority scheme that terminates in (at most) N phases, where N is the total number of non-overlapped channels available in the network. The input to the algorithm is a fully connected mesh network where the number of radios installed on each node out-numbered the available non-overlapped channels. Augmented with the fast switching capability and process coordination module, the algorithm allocates channels to every pair of communicating radios in an ordinally efficient and fair manner. We illustrate our algorithm in detail, prove its correctness and calculate the complexity. The order-of-magnitude analysis of its overall complexity reveal a $O(m^2)$ running time. Thus a more detailed analysis currently studied is expected to further prove its supremacy in terms of performance over the previous proposal and lead to better performance of multi-radio wireless mesh network.

References

1. Kyasanur, P.N.: Multi-Channel Wireless Networks: Capacity and Protocols, PhD Dissertation, Graduate College of the University of Illiois at Urbana-Champaign (2006)
2. Strix System, The business cases for wireless mesh networks, http://www.strixsystems.com/case-studies/WiFi-Mesh-business-case.asp
3. Ramachandran, K.N., et al.: Interference–Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In proc. of IEEE Infocom (2006), http://www.cs.ucsb.edu/~ebelding/txt/infocom06.pdf

4. Skalli, H., et al.: Traffic and Interference aware channel assignment for multi-radio Mesh Wireless Mesh Networks. In: Proc. of IEEE of the 13th annual ACM International conference on mobile computing and networking, Quebec, pp. 15–26 (2007)
5. Raniwala, A., Chiueh, T.: Architecture and Algorithms for IEEE 802.11 Based Multi-Channel Wireless Mesh Network. In: Proc. of IEEE INFOCOM 2005, vol. 3, pp. 223–234 (2005)
6. Katta, A.-K., et al.: A solution to the random assignment problem on full preference domain. Journal of Economic theory 131(1), 231–250 (2006)
7. De Couto, D., Aguayo, D., Bicket, J., Morris, R.: High –throughput path metric for multi-hop wireless routing. In: MOBICOM 2003 (2003)
8. Draves, R., Padhye, J., Zill, B.: Routing in multi-Radio, multi-hop wireless mesh network. In: MOBICOM (2004)
9. Kodiaalm, M., Nandagopal, T.: Characterising achievable rates in mult-hop wireless networks: The joint routing and scheduling problem. In: MOBICOM (2001)
10. Agha, K.A., et al.: Spatial Reuse. In: Wireless LAN Networks, `http://www.gang.inra.fr/~viennot/postscript/ifip2001.ps.gz`
11. Goodman, S.E., Hedetniemi, S.T.: Introduction to the design and analysis of Algorithm. McGraw-Hill, New York (1997)

Ranking and Sorting in Unreliable Single Hop Radio Network

Marcin Kik

Institute of Mathematics and Computer Science,
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Marcin.Kik@pwr.wroc.pl

Abstract. We propose simple and efficient sorting algorithm for unreliable single hop radio network. (In such network each listening station receives transmitted message with some probability $\mathsf{p} < 1$.) We also propose a method of periodic transmission of a sorted sequence that allows for efficient and energetically safe ranking in this sequence.

1 Introduction

We consider the problems of sorting and ranking in unreliable single hop radio network. Such network consists of n stations $s_0, \ldots, s_{n-1}$ communicating with each other by exchanging short radio messages. The stations are synchronized. Time is divided into *slots*. Within a single time slot a single message can be broadcast. During each time slot each station is either listening or sending or idle. If it is sending or listening then it dissipates a unit of energy. We assume that the stations are powered by batteries. Therefore we want to minimize *energetic cost* of the algorithm, i.e. maximum over all stations of non-idle time slots. Each station is in the range of any other station (i.e. a *single hop* network). If two or more stations send messages simultaneously, then a *collision* occurs. In this paper we consider only collision-less algorithms. If during time slot t only one station sends a message and any other station (say s_i) is listening, then s_i receives the message with probability p (*probability of successful reception*). The special case $p = 1$ means *reliable* network. The previously proposed sorting algorithms for this model ([10], [6], [3], [4], [5]) were designed for reliable network. If any transmission failed then the whole output would be devastated. Since radio transmissions are vulnerable to many unpredictable external interferences, we believe that practical algorithms should be robust to occasional losses of received messages. A simple general strategy of increasing the robustness of the algorithm is to make each transmission robust by repeating it many times. If the transmission from single sender to single receiver is repeated r times then the probability of failure is reduced from q to q^r, where $q = 1 - p$. However, the energetic cost of sending is increased r times. (The receiver may stop listening

* This work has been supported by the ICT Programme of the European Union under contract number FP7-215270 (FRONTS).

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 333–344, 2008.
© Springer-Verlag Berlin Heidelberg 2008

as soon as it receives the message.) The situation is still worse if there are m receivers, $m > 1$. We should ensure that **all** receivers have received the message with high probability.

Any sorting algorithm consists of $\Omega(n)$ transmissions, and any of those transmissions may have a large number of receivers. It seems that constructing an algorithm with reasonably high probability of success requires a lot of energy. The number of repetitions for each step should be rather overestimated since the failure of any robust step makes all the previous and remaining computations useless. We propose sorting based on simple merge-sort presented in [4]. Because of the asymmetry between sending and receiving energetic costs in that algorithm, the asymptotic expected energetic cost of our robust algorithm is as low as that of the sorting algorithms with asymptotically lower costs (e.g. [1], [10]) with retransmissions of each step, while the low constants and simplicity make it preferable in practical implementations.

By *ranking* we mean the problem of locating the position of some key x in a sorted sequence of keys (i.e. the number of keys in the sequence that are less than x). One of the many applications of efficient sorting and ranking algorithms may be the routing of packets. The routing algorithms for single hop network ([2], [7], [8], [3]) typically consist of some preprocessing reservation phase that allows for subsequent energetically efficient delivery of the packets. Such preprocessing may consist of sorting the addresses of the packets, and ranking by each station its own address and the next address in the sorted sequence (see e.g. [3]). Then the packets are delivered according to the sorted sequence and each station knows the interval of time slots in which it should listen. Note that even the approximation of such interval (its superset) can be useful. The ranking algorithms proposed in this paper find the exact rank by updating its lower and upper bounds (until they meet each other) while listening to the iterated transmissions of the sorted sequence. The quality of these bounds after the first iteration depend solely on the probability p and can be used for approximation of such interval. We also consider the case when the ranking station may start at arbitrary time slot while the sequence is periodically transmitted. Here we propose that the sorted sequence is transmitted in *recursive bisection ordering* (rbo), which is easily computable permutation. In the case of a reliable network we formally prove in Section 3.1 that the energy used by the ranking station is then $O(\lg n)$.

In our algorithms each message contains only a single key of the input sequence.

2 Preliminaries

We formulate the problem of sorting as follows: Each station s_i initially stores a key in its local variable $\mathsf{key}[s_i]$. The task of each station s_i is to compute the value $\mathsf{idx}[s_i]$ which is the index of $\mathsf{key}[s_i]$ in the sorted sequence of keys. (The indexes are numbered from 0 to $n-1$.)

In Section 3 we consider the problem of *ranking*: The sorted sequence is transmitted periodically (in some fixed ordering π). Each round requires n time slots.

During any time slot any station may start the computation of the rank (or its approximation) of some key in the transmitted sequence. (By the *rank* of the *key* in the sequence s we mean the number of elements of s that are less or equal to the *key*).

In this paper "lg" denotes "$\log_2$". For simplicity of description we assume that n (number of keys and stations) is a power of two (i.e. $\lg n$ is integer). By $Pr(\mathcal{E})$ we denote probability of the event $\mathcal{E}$. By $E[X]$ we denote expected value of random variable X. By $|S|$ we denote the size of the set S. Whenever we define a permutation π of $\{0,\ldots,n-1\}$, π^{-1} denotes the permutation reverse to π and we settle that $\pi(NIL) = \pi^{-1}(NIL) = NIL$, where NIL is a special constant distinct from all numbers.

3 Ranking

Let $n = 2^k$, where k is positive integer. The *generic ranking algorithm* is defined as follows: Let π_k be a permutation of the elements $\{0,\ldots,n-1\}$. Let $b_0,\ldots,b_{n-1}$ be a sorted sequence of keys. The sequence permuted by π_k is transmitted periodically, i.e. for $t \geq 0$, b_i such that $\pi_k(i) = t \bmod n$ is transmitted in time slot t. Let a be a station that wants to compute the rank of $\mathsf{key}[a]$ in the sorted sequence. a can start in arbitrary time slot. It knows permutation π_k and the numbering of time slots. Station a contains variables $\mathsf{minR}[a]$ and $\mathsf{maxR}[a]$ that are updated during successful receptions. Initially $\mathsf{minR}[a] = 0$ and $\mathsf{maxR}[a] = n$. In time slot t (i.e. when $key = b_{\pi_k^{-1}(t \bmod n)}$ is transmitted), a does:

> Let $t' = t \bmod n$. If $\mathsf{minR}[a] \leq \pi_k^{-1}(t') < \mathsf{maxR}[a]$ then a listens. If a received the *key*, then
>
> if $\mathsf{key}[a] < key$ then a sets $\mathsf{maxR}[a]$ to $\pi_k^{-1}(t')$, otherwise (i.e. if $key \leq \mathsf{key}[a]$) it sets $\mathsf{minR}[a]$ to $\pi_k^{-1}(t') + 1$.

Note the following invariant: The rank of $\mathsf{key}[a]$ is in the interval $[\mathsf{minR}[a], \mathsf{maxR}[a]]$. Thus as soon as $\mathsf{minR}[a] = \mathsf{maxR}[a]$ the exact rank of $\mathsf{key}[a]$ is computed. Station a participates in the algorithm as long as it needs or some limit imposed on time or its *listening* energy is exceeded.

Lemma 1 can be used for estimating the time needed for exact ranking with high probability.

Lemma 1. *Let c be a positive integer. After $c \cdot n$ time slots $\mathsf{minR}[a] = \mathsf{maxR}[a]$ with probability at least $1 - 2 \cdot (1-p)^c$.*

Proof. Let r be the exact rank of $\mathsf{key}[a]$. To have $\mathsf{minR}[a] = \mathsf{maxR}[a] = r$ the station needs successful reception of the keys b_{r-1} and b_r. The probability that during the c trials a fails to receive the key is $(1-p)^c$. Thus the probability that a fails to receive from both b_{r-1} and b_r is not greater than $2 \cdot (1-p)^c$. □

Lemma 2 estimates the size of the interval $[\mathsf{minR}[a], \mathsf{maxR}[a]]$ after n time slots.

Lemma 2. *The expected value of $\Delta = \mathsf{maxR}[a] - \mathsf{minR}[a]$ after n time slots is not greater than $2/p - 2$.*

Proof. Let r be the exact rank of $\mathsf{key}[a]$. In the n time slots all the keys b_i have been transmitted. We may think as follows: each transmission was successful with probability p, and whenever station a actually listened it simply observed this transmission. Let u be minimal integer such that $u = n$ or $r \leq u < n$ and the transmission of b_u was successful. It follows from the construction of the algorithm that a observes this transmission and ends up with $\mathsf{maxR}[a] = u$. Let $X_1 = u - r$. If u had not been limited by n, then $X = u - r + 1$ would have been random variable with geometric distribution: $Pr(X = m) = (1-p)^{m-1} \cdot p$ with the expected value $E[X] = 1/p$. Since $X_1 = \min\{X - 1, n - r\}$, we have $E[X_1] \leq 1/p - 1$. It follows (by symmetry) that, for $X_2 = r - \mathsf{minR}[a]$, $E[X_2] \leq 1/p - 1$. Thus, $E[\Delta] = E[X_1 + X_2] = 2/p - 2$. □

The choice of permutation π_k has great influence on the energy used by a. If π_k is identity, a starts listening in time slot 0 and rank of $\mathsf{key}[a]$ is n, then a is forced to listen in all n time slots. Much better option is to use *bisection ordering* (denoted by bo): first (on level 0) transmit the median x of the sequence, then (on level 1) transmit the two medians of the sub-sequences neighboring to x, and so on. For $n = 2^k$, we define precisely bo_k by selecting upper median whenever we have to choose. There is binary tree of depth $k+1$ corresponding to bisection ordering (see Figure 1). On Figure 1 each argument x is joined by vertical dotted line with its corresponding node labeled by $\mathsf{bo}_k(x)$. Note the dependence between binary representation of x and its position in the tree: The level of positive x is determined by the position of its rightmost one and the digits left to this one form the position of x within the level. $x = 0$ is the only argument placed on level k. For $x > 0$, let $\mathsf{irmo}(x) = \min\{j \geq 0 \mid \lfloor x/2^j \rfloor \bmod 2 = 1\}$ (index of rightmost one), and let $\mathsf{irmo}(0) = -1$. Let $\mathsf{lbo}_k(x) = k - 1 - \mathsf{irmo}(x)$ (level of x in bo_k). Now we can define bo_k as follows: $\mathsf{bo}_k(x) = 2^{\mathsf{lbo}_k(x)} - 1 + \lfloor x/2^{\mathsf{irmo}(x)+1} \rfloor$. (There are $2^{\mathsf{lbo}_k(x)} - 1$ nodes above the level $\mathsf{lbo}_k(x)$ and $\lfloor x/2^{\mathsf{irmo}(x)+1} \rfloor$ is the position of x on the level $\mathsf{lbo}_k(x)$.) The permutation reverse to bo_k can be computed as follows: Let $\mathsf{lev}(y) = \lfloor \lg(y+1) \rfloor$. Then $\mathsf{bo}_k^{-1}(y) = \left(y - \left(2^{\mathsf{lev}(y)} - 1\right)\right) \cdot 2^{k - \mathsf{lev}(y)} + \lfloor 2^{k - \mathsf{lev}(y) - 1} \rfloor$. (For $y = 2^k - 1$, the result is zero, and, for $y < 2^k - 1$, the first

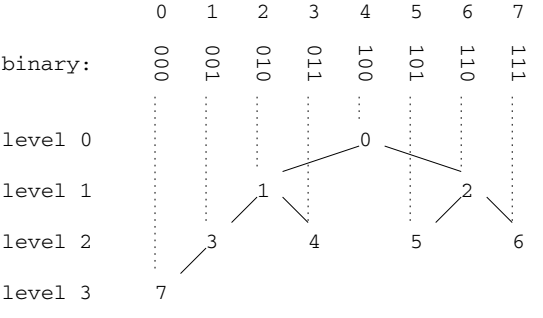


Fig. 1. The tree of bo_3

component of the sum is the position of y within the level $\mathsf{lev}(y)$ multiplied by $2^{k-\mathsf{lev}(y)}$ and the second component settles the rightmost one.)

In reliable networks the ordering of transmissions $\pi_k = \mathsf{bo}_k$ guaranties that if a starts in time slot t such that $t \bmod n = 0$, then a has to listen at most once on each level and therefore uses no more than $k+1$ units of energy. However, if we let a start its computation in arbitrary time slot, then a may be forced to listen in many time slots. For example, if a starts in time slot $n/2-1$ and the rank of $\mathsf{key}[a]$ is n, then a must listen in all $n/2$ time slots on level $k-1$. On the other hand, forcing a to wait until time slot t such that $t \bmod n = 0$ may cause serious delays. Therefore we propose slightly more "sophisticated" permutation that to a large extent eliminates this problem. The permutation *recursive bisection ordering* (rbo_k) is defined as follows: First we permute the elements according to bisection ordering and then we permute each level (except the first and the last one) according to recursive bisection ordering. The permutations rbo_k and rbo_k^{-1} can be computed by Algorithms 1 and 2, respectively. The permutation π_k can be imagined as a set of parallel vertical blades cutting of parts of horizontal interval containing the rank of $\mathsf{key}[a]$ as it falls downwards. To appreciate the difference between bo_k and rbo_k see Figure 2. Even if many highest blades of rbo are missing, the remaining ones perform (less exact) bisection.

function $\mathsf{rbo}_k(\mathsf{x})$
begin
  if $\mathsf{x} = 0$ **then return** $2^k - 1$;
  $\mathsf{y} \leftarrow \mathsf{bo}_k(\mathsf{x})$;
  if $\mathsf{y} = 0$ **then return** 0;
  $\mathsf{above} \leftarrow 2^{\mathsf{lev}(y)} - 1$;
  return $\mathsf{above} + \mathsf{rbo}_{\mathsf{lev}(y)}(\mathsf{y} - \mathsf{above})$;
end

Algorithm 1. Computation of $\mathsf{rbo}_k(\mathsf{x})$

function $\mathsf{rbo}_k^{-1}(\mathsf{y})$
begin
  if $\mathsf{y} = 2^k - 1$ **then return** 0;
  if $\mathsf{y} = 0$ **then return** $\mathsf{bo}_k^{-1}(0)$;
  $\mathsf{above} \leftarrow 2^{\mathsf{lev}(y)} - 1$;
  return $\mathsf{bo}_k^{-1}\left(\mathsf{above} + \mathsf{rbo}^{-1}_{\mathsf{lev}(y)}(\mathsf{y} - \mathsf{above})\right)$;
end

Algorithm 2. Computation of $\mathsf{rbo}_k^{-1}(\mathsf{y})$

Next we show that if the station a starts in time slot 0 and we use permutation bo or rbo, then the expected energy used by a during the first iteration is very low.

Lemma 3. *Let π_k be bo_k or rbo_k. Let the station a start in time slot 0. The expected energy used by a during the first $n = 2^k$ time slots is at most $1 + k \cdot (2/p - 1)$.*

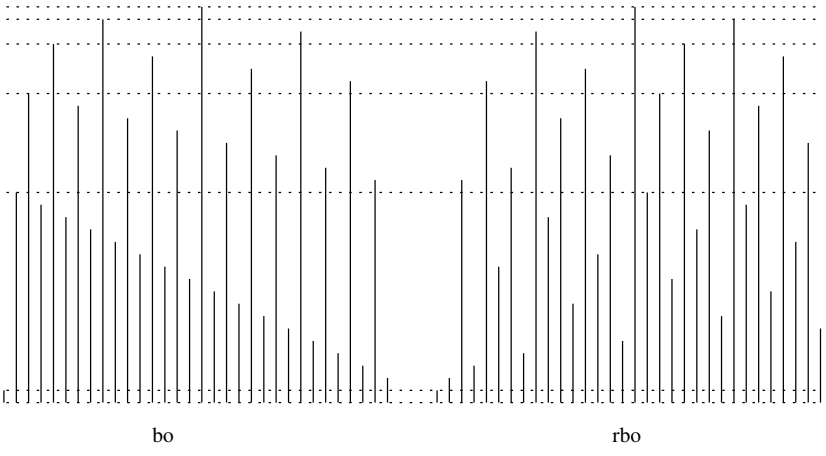


Fig. 2. Permutations bo_5 and rbo_5. (Dotted lines denote borders between levels of bo_5).

Proof. Let $B_l = \langle b_{l,0}, \ldots, b_{l,2^{l+1}-2} \rangle$ be a sequence of keys $\{b_j \mid \mathsf{lbo}_k(j) \leq l\}$ (i.e. the sub-tree of bisection tree from level 0 to l) sorted by j. Station a has to listen to $b_{0,0}$ (the root of bisection tree). Thus the energy used by a on level 0 is 1. For $l \geq 0$, let $r_l = |\{b_{l,i} \mid b_{l,i} \leq \mathsf{key}[a]\}|$ (i.e. the rank of a in the sequence B_l). Let d be maximal integer such that $d = -1$ or $0 \leq d < r_l$ and the transmission of $b_{l,d}$ was successful. Let u be minimal integer such that $u = 2^{l+1} - 1$ or $r_l \leq u < 2^{l+1} - 1$ and the transmission of $b_{l,u}$ was successful. As in the proof of Lemma 2, we can show that the expected value of $u - (d+1)$ is not greater than $2/p - 2$. It follows from the construction of the algorithm, that the station a will not listen to any keys that are before $b_{l,d}$ or after $b_{l,u}$ on the following levels of bisection tree. The sequence B_{l+1} consists of the keys from level $l+1$ on even positions and of the keys from B_l on odd positions. Thus a will have to listen to at most $X_l = u - d$ keys on level $l+1$ and $E[X_l] \leq 2/p - 1$. □

3.1 Ranking with rbo in a Reliable Network

In this subsection we assume that probability of successful reception is $p = 1$ (i.e. a reliable network), and the station a can start in arbitrary time slot t_0 (w.l.o.g. we assume that $0 \leq t_0 < n$) and continues until it learns its rank r_a. We also assume that the used permutation is $\pi_k = \mathsf{rbo}_k$, where $k = \lg n$ is positive integer.

Note that a listens until the last b_j with $j \in \{r_a - 1, r_a\}$ has been transmitted. This happens within n time slots. For given subset of indexes $S \subseteq \{0, \ldots, n-1\}$ and rank $r \in \{0, \ldots, n\}$, let $l(S, r) = \max\{i + 1 \mid i \in S \cup \{-1\} \wedge i < r\}$ and $u(S, r) = \min\{i \mid i \in S \cup \{n\} \wedge r \leq i\}$. For $t \geq t_0$, let S_t be a set of indexes of keys that have been transmitted during time slots $t_0, \ldots, t$. Then, just after t, $\mathsf{minR}[a] = l(S_t, r_a)$ and $\mathsf{maxR}[a] = u(S_t, r_a)$ and, for any $S' \subseteq S_t$, $l(S', r_a) \leq \mathsf{minR}[a] \leq \mathsf{maxR}[a] \leq u(S', r_a)$.

For a sequence of non-negative integers α we define subset of indexes $L(\alpha)$ as follows: $L(\langle\rangle) = \{0, \ldots, n-1\}$, and $L(\alpha \cdot \langle l \rangle)$ is the lth level of bisection tree formed from $L(\alpha)$ ("$\cdot$" denotes concatenation). Note that if $|L(\alpha)| = 2^{l'} \geq 1$ then, for $l < l'$, $|L(\alpha \cdot \langle l \rangle)| = 2^l$ (full levels), and for $l = l'$, $|L(\alpha \cdot \langle l \rangle)| = 1$ (the last level is singleton), and for $l > l'$, $|L(\alpha \cdot \langle l \rangle)| = 0$ (empty levels below the tree). In $\mathsf{rbo}_{\lg n}$ the sequence of subsets of indexes is: $L(\langle 0 \rangle), \ldots, L(\langle \lg n \rangle)$, and within each $L(\alpha)$ the sequence is: $L(\alpha \cdot \langle 0 \rangle), \ldots, L(\alpha \cdot \langle \lg |L(\alpha)| \rangle)$.

Lemma 4. *Let $|L(\alpha)| = 2^{l'} \geq 2$, and $1 \leq l < l'$. Let $t = \max\{\mathsf{rbo}_{\lg n}(x) \mid x \in L(\alpha \cdot \langle l \rangle)\}$. If, just after time slot t, $\mathsf{minR}[a] \geq l(L(\alpha \cdot \langle l \rangle), r_a)$ and $\mathsf{maxR}[a] \leq u(L(\alpha \cdot \langle l \rangle), r_a)$ then during each of the levels $L(\alpha \cdot \langle l+1 \rangle)$,...,$L(\alpha \cdot \langle l' \rangle)$ the station a listens at most twice.*

Proof. During transmission of the level $L(\alpha \cdot \langle l+1 \rangle)$ the station a listens only to the keys b_j with $\mathsf{minR}[a] \leq j \leq \mathsf{maxR}[a] - 1$. Since there are no such nodes in $L(\alpha \cdot \langle l \rangle)$, the only such nodes in $L(\alpha \cdot \langle l+1 \rangle)$ possibly are: the right child of $\mathsf{minR}[a] - 1$ (if $\mathsf{minR}[a] - 1 \in L(\alpha \cdot \langle l \rangle)$) and the left child of $\mathsf{maxR}[a]$ (if $\mathsf{maxR}[a] \in L(\alpha \cdot \langle l \rangle)$) in the tree $L(\alpha)$. After transmission of $L(\alpha \cdot \langle l+1 \rangle)$, we have $\mathsf{minR}[a] \geq l(L(\alpha \cdot \langle l+1 \rangle), r_a)$ and $\mathsf{maxR}[a] \leq u(L(\alpha \cdot \langle l+1 \rangle), r_a)$. Thus we can repeat the same reasoning for each following level in the tree $L(\alpha)$. □

Lemma 5. *Let $|L(\alpha)| = 2^{l'} \geq 16$, and $2 \leq l < l'$. Let $t = \max\{\mathsf{rbo}_{\lg n}(x) \mid x \in L(\alpha \cdot \langle l \rangle)\}$. If, just after time slot t, $|\{x \in L(\alpha \cdot \langle l \rangle) \mid \mathsf{minR}[a] - 1 < x < \mathsf{maxR}[a]\}| \leq 1$ then during the transmissions of $L(\alpha \cdot \langle l+1 \rangle)$ the station a listens at most four times.*

Proof. In the worst case a listens in $L(\alpha \cdot \langle l+1 \rangle)$ to some subset of: right child of $\mathsf{minR}[a] - 1$, both children of the single x between $\mathsf{minR}[a] - 1$ and $\mathsf{maxR}[a]$, and left child of $\mathsf{maxR}[a]$ in the tree $L(\alpha)$. □

Theorem 1. *If the assumptions formulated in the first paragraph of this subsection hold then the station a listens at most $4 \lg n$ times before it learns its rank.*

Proof. Let $U(\alpha, l) = \bigcup_{i=0}^{l} L(\alpha \cdot \langle i \rangle)$ (i.e. the uppermost $l+1$ levels of the tree $L(\alpha)$). Let $D(\alpha, l) = \bigcup_{i=l}^{\lg |L(\alpha)|} L(\alpha \cdot \langle i \rangle)$ (i.e. the lowest $\lg |L(\alpha)| - l + 1$ levels of the tree $L(\alpha)$).

For $t \geq 0$, let $\gamma(t)$ be the shortest sequence such that $x = \mathsf{rbo}^{-1}_{\lg n}(t \bmod n)$ is the root of the tree $L(\gamma(t))$ and let $\delta(t) = \min\{\delta \geq 0 \mid |L(\gamma(t+\delta))| \geq 2\}$. For $0 \leq i < \delta(t)$, $L(\gamma(t+i))$ is the last level (singleton) of some tree T_i. Hence, $L(\gamma(t+i+1))$ is a level of some tree T_{i+1} such that $|T_{i+1}| \geq 2 \cdot |T_i|$, or of the whole tree $L(\langle\rangle)$ if $(t+i+1) \bmod n = 0$ (in this case: $i+1 = \delta(t)$). T_i is predecessor of the level $L(\gamma(t+i+1))$ in T_{i+1}. $L(\gamma(t+\delta(t)))$ is a full level, thus its size is at least $2 \cdot |T_{\delta(t)-1}|$. Hence we have:

Claim. $|L(t+\delta(t))| \geq 2^{\delta(t)}$.

Let $\beta = \gamma(t_0 + \delta(t_0))$. If β is empty sequence, then a listens at most $\delta(t_0)$ times and then at most $\lg n + 1$ times starting from the root of the whole tree $L(\langle\rangle)$. By the Claim: $\delta(t_0) + \lg n + 1 \leq 2\lg n + 1$.

Otherwise, let $\beta = \langle l_1, \ldots, l_R\rangle$ and let $\beta_i = \langle l_1, \ldots, l_i\rangle$. Note that R (the length of β) is the level of recursion on which the bisection tree of $L(\beta)$ is formed and that $l_R \geq 1$, since $|L(\beta)| = 2^{l_R} \geq 2$.

First a listens $\delta(t_0)$ times. By the Claim: $\delta(t_0) \leq l_R$, since $|L(\beta)| = 2^{l_R}$. Then a listens to $L(\beta)$ starting from the root of $L(\beta)$. Thus it listens at most $l_R + 1$ times and, after that, $\mathsf{minR}[a] \geq l(L(\beta), r_a)$ and $\mathsf{maxR}[a] \leq u(L(\beta), r_a)$. Then rbo steps back one recursion level. Then it listens to (possibly empty) sequence of sets $L(\beta_{R-1} \cdot \langle l_R + 1\rangle)$, $\ldots$, $L(\beta_{R-1} \cdot \langle l_{R-1}\rangle)$. By Lemma 4, a listens at most twice in each of these sets. After that $\mathsf{minR}[a] \geq l(D(\beta_{R-1}, l_R), r_a)$ and $\mathsf{maxR}[a] \leq u(D(\beta_{R-1}, l_R), r_a)$. Then rbo steps back one recursion level. Such stepping back is repeated $R-1$ times, for i taking values $R-1,\ldots,1$. For each such i, initially $\mathsf{minR}[a] \geq l(D(\beta_i, l_{i+1}), r_a)$ and $\mathsf{maxR}[a] \leq u(D(\beta_i, l_{i+1}), r_a)$, and the following (possibly empty) sequence of sets is transmitted: $L(\beta_{i-1} \cdot \langle l_i + 1\rangle)$, $\ldots$, $L(\beta_{i-1} \cdot \langle l_{i-1}\rangle)$. Note that, since $L(\beta_{i+1}) = L(\beta_i \cdot \langle l_{i+1}\rangle)$ is a full (i.e. not last) level of $L(\beta_i)$, for each $x \in U(\beta_i, l_{i+1} - 1)$, the predecessor and the successor of x in $L(\beta_i)$ are in $D(\beta_i, l_{i+1})$ and, hence, $|\{x \in L(\beta_i) \mid l(D(\beta_i, l_{i+1}), r_a) - 1 < x < u(D(\beta_i, l_{i+1}), r_a))\}| \leq 1$. Thus, by Lemma 5, since $L(\beta_i) = L(\beta_{i-1} \cdot \langle l_i\rangle)$, the station a has to listen at most four times in $L(\beta_{i-1} \cdot \langle l_i + 1\rangle)$ and, by Lemma 4, at most twice in each of the remaining sets.

Consider the sequence of **all** the sets mentioned above. For each set $L(\beta_i \cdot \langle j\rangle)$ in the sequence (except $L(\beta_{R-1} \cdot \langle l_R\rangle) = L(\beta)$ – the first one) its index "j" is greater than the index of its predecessor. The greatest possible value of j is $\lg n$ (in the set $L(\langle\rangle \cdot \langle \lg n\rangle)$). Thus the number of the sets following $L(\beta)$ is at most $\lg n - l_R$ and in each of them a listens at most four times. Each set in which a has to listen more than twice (which is a full level) must be followed by at least one set (e.g. the last level) in which a has to listen at most two times. Thus the energy used by a while listening to the sequence of sets is at most $(4 \cdot \frac{1}{2} + 2 \cdot \frac{1}{2})(\lg n - l_R) + (l_R + 1) \leq 3\lg n - 2l_R + 1$.

This procedure is finished, for the last $i = 1$, just before time slot n that starts the next round. In this round a listens no more than $\lg n$ times (it does not need to listen in the last level again). Adding $\delta(t_0) \leq l_R$ initial slots, we have upper bound $4\lg n - l_R + 1 \leq 4\lg n$ on the energy used by a. □

The estimation $4\lg n$ of Theorem 1 seems to be very pessimistic. (In our tests a never had to listen more than $2\lg n$ times.) Nevertheless, it shows that the station a can safely start its ranking at any time slot. (This reduces the upper bound on ranking time from $2n - 1$ to n.) The simulations indicate that rbo is also energetically efficient on unreliable network (i.e. when $p < 1$).

4 Sorting

We assume that $\lg n$ is positive integer. Each station s_i contains variables: $\mathsf{idx}_0[s_i]$, $\ldots$, $\mathsf{idx}_{\lg n}[s_i]$, $\mathsf{minR}_0[s_i]$, $\ldots$, $\mathsf{minR}_{\lg n-1}[s_i]$, $\mathsf{maxR}_0[s_i]$, $\ldots$,

$\mathsf{maxR}_{\lg n-1}[s_i]$. The variables are initialized by the procedure init (see Algorithm 3). The ultimate goal for each s_i is to compute $\mathsf{idx}_{\lg n}[s_i]$, which is the

```
procedure init
Each s_i does (in parallel):
begin
    idx_0[s_i] ← 0;
    for k ← 1 to lg n do idx_k[s_i] ← NIL ;
    for k ← 0 to lg n − 1 do
        minR_k[s_i] ← 0;
        maxR_k[s_i] ← 2^k;
end
```

Algorithm 3. Procedure init

index of $\mathsf{key}[s_i]$ in the sorted sequence of keys. Our algorithm is designed to perform *stable* sorting (i.e. the initial ordering between equal keys is preserved). The basic building block of our algorithm is the procedure $\mathsf{rank}(k, l, d, \pi_k)$, where $d \in \{0, 1\}$ and $0 \leq l < n/2^{k+1}$, (see Algorithm 4) that tries to find the rank of each key from the stations $s_{l\cdot 2^{k+1}+d\cdot 2^k}, \ldots, s_{l\cdot 2^{k+1}+d\cdot 2^k+2^k-1}$ in the sorted sequence of keys from the stations $s_{l\cdot 2^{k+1}+(1-d)\cdot 2^k}, \ldots, s_{l\cdot 2^{k+1}+(1-d)\cdot 2^k+2^k-1}$. Once the station s_i knows its rank r in the neighboring sequence and its index idx in its own sorted sequence, it can compute its index $(r + \mathsf{idx})$ in the sequence merged from the two sequences. The permutation π_k is either rbo_k or bo_k (defined in Section 3). For any k, $0 \leq k < \lg n$, all procedures $\mathsf{rank}(k, l, d, \pi_k)$ are used to produce indexes for sorted sub-sequences of length 2^{k+1}. This is done by procedure $\mathsf{levelRanking}(k, \pi_k)$ (see Algorithm 5). We refer to k as a *level*.

Sorting algorithms can be built by composing sequences of levelRanking for various levels. For $n > 0$ and $0 < q, q' < 1$, let $c(q, q', n) = \lceil \log_{1/q}\left(\frac{2n \lg n}{q'}\right)\rceil = \lceil (1 + \lg n + \lg\lg n + \lg(1/q'))/\lg(1/q)\rceil$. We propose and analyze a simple procedure $\mathsf{sorting}_{q'}$ (see Algorithm 6) that successfully sorts with probability $1 - q'$ by repeating levelRanking $c(q, q', n)$ times on each level. The output consists of the final values of $\mathsf{idx}_{\lg n}$ in the stations.

Theorem 2. *For $0 < q' < 1$, the procedure $\mathsf{sorting}_{q'}$ (Algorithm 6) sorts any input sequence with probability greater or equal $1 - q'$.*

Proof. Let $q = 1 - p$ and $c = c(q, q', n)$. Let $\mathcal{Q}$ be the event that $\mathsf{sorting}_{q'}$ failed to sort (i.e. some indexes remained uncomputed). For $0 \leq k < \lg n$, let $\mathcal{Q}_k$ be the event that the first failure occurred at level k (i.e. some $\mathsf{idx}_{k+1}[s_i]$ remained uncomputed, while all values $\mathsf{idx}_{k'}[s]$, for $0 \leq k' \leq k$, for each station s, are computed.) Thus $Pr(\mathcal{Q}) = \sum_{k=0}^{\lg n-1} Pr(\mathcal{Q}_k)$ ($\mathcal{Q}$ is disjoint union of all events $\mathcal{Q}_k$). Let $\mathcal{F}_k$ be the event that repeating c times $\mathsf{levelRanking}(k, \mathsf{rbo}_k)$ fails to compute all indexes on level $k + 1$ under the condition that all indexes on levels up to k have been computed. Let $q = 1 - p$. By Lemma 1, the probability that some given index remains uncomputed is not greater than $2 \cdot q^c$. Thus $Pr(\mathcal{F}_k) \leq 2nq^c$, as we have to compute n indexes. Let $\mathcal{E}_k$ be the event that

```
procedure rank(k, l, d, _k)
for 0 ≤ i < 2^k:

  – let a_i denote s_{l·2^{k+1}+d·2^k+i}, and
  – let b_i denote s_{l·2^{k+1}+(1−d)·2^k+i}.

for time slot t ← 0 to 2^k − 1 do
    the (at most one) b_j with _k(idx_k[b_i]) = t broadcasts key = key[b_j];
    let x = _k^{−1}(t);
    each a_i with minR[a_i] ≤ x < maxR_k[a_i] does:
    begin
        a_i listens;
        if a_i received key then
            (* comparison for stable ranking *)
            if (d = 0 and key[a_i] ≤ key) or (d = 1 and key[a_i] < key ) then
                maxR_k[a_i] ← x;
            else
                minR_k[a_i] ← x + 1;
            (* cascading computation of indexes *)
            k' ← k;
            while k' < lg n and idx_{k'}[a_i] ≠ NIL and minR_{k'}[a_i] = maxR_{k'}[a_i] do
                idx_{k'+1}[a_i] ← idx_{k'}[a_i] + minR_{k'}[a_i];
                k' ← k' + 1;
    end
```

Algorithm 4. Procedure rank

```
procedure levelRanking(k, _k)
for l ← 0 to n/(2^{k+1}) − 1 do
    rank(k, l, 1, _k);
    rank(k, l, 0, _k);
```

Algorithm 5. Procedure levelRanking

all indexes on levels up to k has been properly computed in $\mathsf{sorting}_{q'}$. Then $Pr(\mathcal{Q}_k) = Pr(\mathcal{E}_k) \cdot Pr(\mathcal{F}_k) \leq Pr(\mathcal{F}_k)$ and, hence, $Pr(\mathcal{Q}) \leq 2nq^c \cdot \lg n$. It is easy to verify that $c \geq \min\{c \mid q^c \cdot 2n \lg n \leq q'\}$. This completes the proof. □

Theorem 3. *For any input, the expected energy used for listening by any single station in* $\mathsf{sorting}_{q'}$ *is at most* $\lg n \cdot (1 + (c-1)(2/p-2) + (2/p-1)(\lg n - 1)/2) + c(n-1)q'$, *where* $c = c(q, q', n)$. *The energy used for sending by any single station is* $c \lg n$. *Time of* $\mathsf{sorting}_{q'}$ *is* $cn \lg n$.

Proof. Let the input sequence be arbitrary and let s be any of the stations. Let X be random variable that is the energy used for listening by s. Let X_k be random variable that is the energy used by s in all c levelRankings on level k. Thus $X = \sum_{k=0}^{\lg n - 1} X_k$. Let Ω be the set of all elementary events (i.e. of all possible computations). Note that $E[X] = \sum_{\omega \in \Omega} X(\omega) \cdot Pr(\omega)$, where $X(\omega)$ is

procedure $\mathsf{sorting}_{q'}$
init;
Let $\mathsf{q} = 1 - \mathsf{p}$, where p is probability of successful reception;
for $\mathsf{k} \leftarrow 0$ **to** $\lg \mathsf{n} - 1$ **do**
  repeat $\mathsf{c}(\mathsf{q}, \mathsf{q}', \mathsf{n})$ times $\mathsf{levelRanking}(\mathsf{k}, \mathsf{rbo}_k)$;

Algorithm 6. Procedure sorting

the energy used by s in the computation ω and $Pr(\omega)$ is the probability of this computation. Let the events $\mathcal{Q}$, $\mathcal{E}_k$ be defined as in the proof of Theorem 2. Let $\mathcal{E} = \mathcal{E}_{\lg n}$. Note that Ω is a disjoint union of $\mathcal{Q}$ and $\mathcal{E}$ and, hence, $E[X] = S_{\mathcal{Q}} + S_{\mathcal{E}}$, where $S_{\mathcal{Q}} = \sum_{\omega \in \mathcal{Q}} X(\omega) \cdot Pr(\omega)$ and $S_{\mathcal{E}} = \sum_{\omega \in \mathcal{E}} X(\omega) \cdot Pr(\omega)$.

To estimate $S_{\mathcal{Q}}$ note that in the levelRanking on level k there are only 2^k time slots in which s is allowed to listen. Thus $X_k(\omega) \leq c \cdot 2^k$ and $X(\omega) \leq c \sum_{k=0}^{\lg n - 1} 2^k = c(n-1)$ and $S_{\mathcal{Q}} \leq c(n-1) \cdot \sum_{\omega \in \mathcal{Q}} Pr(\omega) \leq c(n-1)q'$ (by Theorem 2).

To estimate $S_{\mathcal{E}}$ note that

$$S_{\mathcal{E}} = \sum_{\omega \in \mathcal{E}} \sum_{k=0}^{\lg n - 1} X_k(\omega) \cdot Pr(\omega) = \sum_{k=0}^{\lg n - 1} \sum_{\omega \in \mathcal{E}} X_k(\omega) \cdot Pr(\omega)$$
$$\leq \sum_{k=0}^{\lg n - 1} \sum_{\omega \in \mathcal{E}_k} X_k(\omega) \cdot Pr(\omega) \leq \sum_{k=0}^{\lg n - 1} \sum_{\omega \in \mathcal{E}_k} \frac{X_k(\omega) \cdot Pr(\omega)}{Pr(\mathcal{E}_k)} = \sum_{k=0}^{\lg n} E[X_k | \mathcal{E}_k],$$

where $E[X_k|\mathcal{E}_k]$ is expected value of X_k under the condition $\mathcal{E}_k$ that all indexes up to level k have been computed. (The first inequality above follows from $\mathcal{E} \subseteq \mathcal{E}_k$, and the second one follows from $Pr(\mathcal{E}_k) \leq 1$.) Under the condition $\mathcal{E}_k$ the expected energy used for listening by s during the first levelRanking on level k is at most $1 + k(2/p - 1)$ (by Lemma 3). By Lemma 2, the expected value of $\Delta = \mathsf{maxR}_k[s] - \mathsf{minR}_k[s]$ after the first levelRanking on level k is $2/p - 2$. During each of the remaining $c - 1$ levelRankings on level k station s can listen to at most Δ stations, thus the expected listening energy for these levelRankings can be bounded by $(c-1)\cdot(2/p-2)$. We have $E[X_k|\mathcal{E}_k] \leq 1 + k(2/p-1) + (c-1)\cdot(2/p-2)$. Thus $S_{\mathcal{E}} \leq \sum_{k=0}^{\lg n - 1} (1 + k(2/p-1) + (c-1)\cdot(2/p-2)) = \lg n(1 + (c-1)(2/p-2)) + (2/p-1)\frac{\lg n(\lg n - 1)}{2}$.

The limits on time and sending energy follow from the fact that each station broadcasts only once in each levelRanking. □

Corollary 1. *Algorithm* $\mathsf{sorting}_{1/n}$ *sorts any input with probability at least* $1 - \frac{1}{n}$ *in time* $O(n \lg^2 n)$ *and, for each station* s*, the expected energy used by* s *is* $O(\lg^2 n)$.

Acknowledgments

Thanks to Maciej Gębala for helpful comments.

References

1. Ajtai, M., Komlós, J., Szemerédi, E.: Sorting in $c \log n$ parallel steps. Combinatorica 3, 1–19 (1983)
2. Datta, A., Zomaya, A.Y.: An Energy-Efficient Permutation Routing Protocol for Single-Hop Radio Networks. IEEE Trans. Parallel Distrib. Syst. 15, 331–338 (2004)
3. Gębala, M., Kik, M.: Counting-Sort and Routing in a Single Hop Radio Network. In: Kutyłowski, M., Cichoń, J., Kubiak, P. (eds.) ALGOSENSORS 2007. LNCS, vol. 4837, pp. 138–149. Springer, Heidelberg (2008)
4. Kik, M.: Merging and Merge-sort in a Single Hop Radio Network. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2006. LNCS, vol. 3831, pp. 341–349. Springer, Heidelberg (2006)
5. Kik, M.: Sorting Long Sequences in a Single Hop Radio Network. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 573–583. Springer, Heidelberg (2006)
6. Nakano, K.: An Optimal Randomized Ranking Algorithm on the k-channel Broadcast Communication Model. In: ICPP 2002, pp. 493–500 (2002)
7. Nakano, K., Olariu, S., Zomaya, A.Y.: Energy-Efficient Permutation Routing in Radio Networks. IEEE Transactions on Parallel and Distributed Systems 12, 544–557 (2001)
8. Nakano, K., Olariu, S., Zomaya, A.Y.: Energy-Efficient Routing in the Broadcast Communication Model. IEEE Trans. Parallel Distrib. Syst. 13, 1201–1210 (2002)
9. Singh, M., Prasanna, V.K.: Optimal Energy Balanced Algorithm for Selection in Single Hop Sensor Network. SNPA ICC (May 2003)
10. Singh, M., Prasanna, V.K.: Energy-Optimal and Energy-Balanced Sorting in a Single-Hop Sensor Network. PERCOM (March 2003)

Distributed Monitoring in Ad Hoc Networks: Conformance and Security Checking

Wissam Mallouli, Bachar Wehbi, and Ana Cavalli

Institut Telecom/Telecom SudParis, CNRS/SAMOVAR
{wissam.mallouli,bachar.wehbi,ana.cavalli}@it-sudparis.eu

Abstract. Ad hoc networks are exposed more than traditional networks to security threats due to their mobility and open architecture aspects. In addition, any dysfunction due to badly configured nodes can severely affect the network as all nodes participate in the routing task. For these reasons, it is important to check the validity of ad hoc protocols, to verify whether the running implementation is conform to its specification and to detect security flows in the network. In this paper, we propose a formal methodology to collect and analyze the network traffic trace. Observers running on a set of nodes collect local traces and send them later to a global observer that correlates them into a global trace thanks to an adapted time synchronization mechanism running in the network. The global trace is then analyzed to study the conformance and the security of the running routing protocol. This analysis is performed using dedicated algorithms that check the collected trace against a set of functional and security properties specified in an adapted formal language.

Keywords: Ad Hoc Networks, Monitoring, Trace Collection and Correlation, Conformance Testing, Security Analysis, Nomad Logic.

1 Introduction

Mobile ad hoc networks (MANET) are infrastructureless networks composed of a set of wireless mobile nodes. Nodes send packets directly to destinations that are in their coverage zone. When destinations are farther than the coverage range intermediate nodes cooperate to establish the communication path. This open and cooperative network aspect and the limited resources of mobile nodes make it difficult to define an efficient testing methodology to validate the conformance of existing routing protocols (like AODV [11], OLSR [6] or DYMO [5] etc.) and to guarantee the respect of predefined security properties.

Formal testing allows to insure the respect of the functional behavior and the security requirements of a system; it can be either active or passive. Active testing permits to validate a system implementation by applying a set of test cases and analyzing the system reaction. It implies that we have a global control on the network architecture which is difficult to perform in a dynamic topology such as ad hoc networks. Besides, the active testing becomes difficult to perform

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 345–356, 2008.
© Springer-Verlag Berlin Heidelberg 2008

when the network is built from components (nodes) that are running in their real environment and cannot be interrupted or disturbed. In this situation, there is a particular interest in using monitoring techniques that consist in testing passively during the run time the traffic flow in a deployed network. This testing consists in analyzing collected data according to some functional and security requirements described in a formal language.

In this paper, we use monitoring to collect distributed traces using local observers (called also probes) without interfering with the network under test. Two type of networks are considered. The first consists of a controlled area where a set of dedicated probes is installed to monitor the network. While the second is an open area network where the nodes perform themselves the trace collection task. In both cases, the local traces are sent to a global observer which is responsible for the traces correlation and analysis tasks. The correlation is performed based on an accurate time synchronization protocol [14] designed for ad hoc networks. This protocol follows the *receiver to receiver* mechanism that eliminates the major sources of synchronization inaccuracy. Whereas, the analysis consists of checking whether the trace is conform to a set of functional and security properties that we describe in a formal language adapted to distributed communicating systems. This checking is performed using a set of appropriate algorithms that we developed for this end. Once a property violation is detected, we identify the irregular node(s) behind it. Our mechanism allows to spot distant attacks that can only be discovered by the analysis of the global trace. More precisely, the main contributions of this paper are:

1. Definition of a precise method to collect distributed traces to cover the whole network. The collection methodology differs depending on the network nature (controlled or open areas).
2. Definition of a method for correlating local traces to obtain a global network trace. This correlation rely on an adapted time synchronization mechanism for ad hoc networks that permits to synchronize all the local observers.
3. Analysis of this global trace using specific algorithms to study the conformance and the security requirements of the considered routing protocol. The proposed algorithms allow to check a set of functional and security properties specified in Nomad formal language [7] on the collected trace.
4. Demonstration of the reliability of our approach by applying it on different ad hoc network scenarios running OLSR routing protocol to detect recurrent failures and attacks.

The remainder of this paper is organized as follows. In section 2, we discuss the related work tackling with monitoring in ad hoc networks. Section 3 presents the distributed collection of the ad hoc network traffic in a case of controlled network and an open area network. In section 4, we expose the approach to correlate the local traces in order to obtain the global network trace. Section 5 presents the methodology to analyze this global trace by comparing it to the functional and security requirements described in Nomad formal language. In section 6 we apply our methodology on OLSR routing protocol and the conclusion id given in section 7.

2 Related Work

Many papers [12,3,13,1] tried to tackle monitoring methodologies in ad hoc networks. In [13] the authors present DAMON, a distributed system for monitoring multi hop mobile networks. DAMON uses agents to collect the network traffic and sends collected measurements to data repositories. It was implemented in an AODV based ad hoc network. WiPal [1] is a merging tool dedicated to IEEE 802.11 traces manipulation which enables merging multiple wireless traces into a unique global one. Although DAMON and WiPal collect the network trace, they provide no process for its analysis.

The authors in [9] propose an intrusion detection scheme based on Extended Finite State Machines (EFSM) [8]. Indeed, they provide a formal model of the correct behavior of the routing protocol and detect specific deviations of the routing protocol implementation using a backward checking algorithm [2]. This work can only detect local attacks that violate the EFSM model of OLSR protocol (which is not the case of a big range of attacks).

The authors in [10] make use of a combination of deontic and temporal logic to specify the correct behavior of a node and to express complex security properties. They investigate different attacks targeting the link sensing mechanism of routing protocols and describe security policies to prevent them. Contrary to our methodology, this work considers only the local traffic trace of a given node. It does not allow to detect remote and distributed attacks. Moreover, it can only discover the existence of an incoherence in the collected traffic without determining the malicious node. In this paper we propose a different formal end-to-end methodology to collect and analyze global ad hoc network traffic.

3 Distributed Traffic Collection in Ad Hoc Networks

Network monitoring is an interesting approach that allows to collect the required information in order to analyze the behavior of the network. Monitoring in ad hoc networks can be *local* with respect to a node or *global* with respect to the network. In ad hoc networks, local monitoring is not sufficient to detect some types of errors and security anomalies [12,9]. For this reason we adopt in this paper the global monitoring approach based on a distributed monitoring.

Controlled Area Network: In this type of network, nodes move inside a defined limited area. Therefore, it is possible to place a set of wireless observers responsible for capturing transited packets. These observers are placed to cover the whole network area. They collect the communication traces and send them to the global observer in the network. The choice of this node (global observer) can be based on administrative preferences. The broadcast nature of the wireless medium combined with the interferences problems represent a classical problem in the monitoring of ad hoc networks. That's why we chose to install the observers in such a way they cover each zone portion twice or more. The advantage of this method is the collection of real network traffic (attackers cannot alter the collected traces).

Open Area Network: In the case of an open area network, the observers are the network nodes themselves. They perform a collaborative observation action. Each network node collects its local traffic trace and sends it to the global observer. We assume here that all the nodes have the collector program running on their systems. As the observers are the network nodes, it is possible for a node (attacker) to alter its collected trace. The traffic analyzer module on the global observer must take this property in consideration. This is the major difference with the limited area network where the collect is made by dedicated observers.

4 Traces Correlation Mechanism

The global observer receives the local traces collected by the local observers in order to analyze them. The first step toward performing this analysis is to correlate the traces and order them chronologically. We use a *receiver to receiver* network wide synchronization mechanism that we designed for wireless multi hop networks. Using this mechanism all the nodes in the network run with the same clock value allowing thus to perform the trace correlation. In the following we briefly describe the synchronization mechanism in section 4.1 and then describe the correlation procedure in section 4.2.

4.1 Synchronization Mechanism Overview

The objective of the time synchronization mechanism is to support each network node with the required timing information in order to build an adjustment function that transforms its local clock value to that of the reference node existing in the network. Using the adjustment functions they calculated, nodes, all over the network, run with similar clock values achieving therefore network wide synchronization. The mechanism is based on *receiver to receiver* synchronization which by definition eliminates the major sources of synchronization inaccuracy (send time and access time). The mechanism consists of two complementary parts; the sender nodes selection and the synchronization process. First, a hierarchy of sender nodes is constructed in order to guide the synchronization process in a multi hop environment. Sender nodes are responsible for transmitting reference messages. A reference message does not contain an explicit timestamp; instead, receivers use its arrival time to compare their clocks. Using information exchanged trough reference messages, each node constructs a table that contains for each received reference message the mapping between its local reception time and that of the reference node (or an already synchronized node). Then the node performs least squares linear regression to estimate the best fit line relating the node's clock to the reference node's clock. The estimated best fit line is an adjustment function that transforms the client's local clock value to that of the reference node. This adjustment function is given by equation 1 below:

$$T_{synch} = (1 + \widetilde{F}) \times T_{local} + \widetilde{Off} \tag{1}$$

Where $\widetilde{F}$ and $\widetilde{Off}$ are the estimated frequency error and offset parameters respectively. The synchronization process uses time information exchanged

through reference messages to achieve first an initial estimate of the node's adjustment function. Then, by observing the offset estimate variation on longer time period, it improves the frequency error estimation and therefore the time synchronization accuracy. Details about the synchronization mechanism can be found in [14].

4.2 Global Trace Construction

Using the synchronization mechanism, network nodes run in phase with the reference clock value. This network virtual clock will assist the global observer in correlating the different local traces received from the set of observers. In [14] we showed that in a multi hop network the precision P of the synchronization mechanism is in the order of few micro seconds (maximum of $5\mu sec$ for nodes at 5 hops away of the time reference) which is by far less than the time difference between two message transmissions (a minimum of $100\mu sec$) and the time difference between the transmission time of a message and its reception time at a neighbor node (higher than $20\mu sec$). According to this accurate precision the following properties are always satisfied:

- If two nodes, $N1$ and $N2$, in the same broadcast region, send two different messages $M1$ then $M2$ at local times $t1$ and $t2$; then $t1$ ¡ $t2$.
- If a node sends a message at local time $t1$, a receiver receives the message at local time $t2$ where $t1$ ¡ $t2$.
- If two messages $M1$ and $M2$ are collected at local times $t1$ and $t2$ where $|t1 - t2| < P$ then either M1 and M2 are the same message or $M1$ and $M2$ are independent (i.e. they are transmitted in two different broadcast zones).

5 Monitoring Methodology

5.1 Functional and Security Properties Formal Specification

We specify a set of properties that the network nodes have to respect using Nomad formal language which allows to express privileges on non atomic actions. It combines deontic and temporal logics and can describe conditional privileges and obligations with deadlines. It can also formally analyze how privileges on non atomic actions can be decomposed into more basic privileges on elementary actions. More details about Nomad syntax and semantics are presented in [7].

Definition 1. *Atomic action*
We define an atomic action as the emission or the reception of a message between two nodes using the following syntax:

$$Node_1 \ ?or!\ Msg(Par_1,Par_2,...,Par_n)\ Node_2$$

where Node_1 and Node_2 represent the source or the destination of the message. '?' and '!' define a reception and an emission of a message by Node_1. $\text{Msg}(\text{Par}_1,\text{Par}_2,...,\text{Par}_n)$ represents the message exchanged between Node_1 and Node_2 with its parameters. Node_1, Node_2, Msg, and Par_i can be replaced by the symbol $*$ to represent any node, any message or any parameter.

Definition 2. *Non-atomic action*
If α and β are actions, then $(\alpha;\beta)$, which means "α is followed immediately by β" and $(\alpha;*;\beta)$, which means " α is followed by β" are non-atomic actions.

Definition 3. *Formulae*
If α is an action then start(α) (action α is being started) and done(α) (action α is done) are formulae.

Some properties on actions and formulae:

- If A and B are formulae then $(A \wedge B)$ and $(A \vee B)$ are formulae.
- If A is a formula then $\neg A$,$\oplus A$ ("Next in the trace," A is true), $\ominus A$ ("previously in the trace, A is true") are formulae.
- If A is a formula then $O^{\leq d}A$ ("d units of time ago, A was true if d<0, or in the next d units of time, A will be true if d>0 ") is a formula.
- $(A|C)$ is a formula: 'In the context C the formula A is true'.

Definition 4. *Deontic modalities*
If A is a formula then modality $\mathcal{O}$ ("A" is mandatory), $\mathcal{F}$ ("A" is forbidden) and $\mathcal{P}$ ("A" is permitted) are formulae.

5.2 Trace Analysis Approach

To run the distributed monitoring process, the global observer needs two different input files: the traces files collected by the local observers and the properties file where are specified expected functional and security properties.

First, the global observer verifies through a syntax checking module that the desired behavior is well specified according to the Nomad format. This avoids syntax-related bugs in the test engine module.

Second, the collected traces files have to be analyzed using a pre-processing module that performs the following tasks: (i) filtering the traces files keeping only the relevant information for the protocol(s) under test. The basic idea is to keep in the traces only the messages and parameters corresponding to the specified properties to check. (ii) correlation of the traces files and the construction of a unique global trace file. (iii) parsing the global trace and creating a trace table

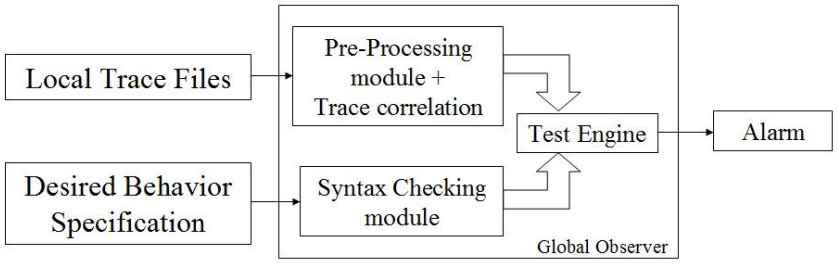


Fig. 1. Monitoring Architecture

which constitutes the target of the 'Test Engine' module queries. Each line of the trace table corresponds to an emission or a reception of a message in the network.

Finally, the trace analysis is performed using three algorithms according to the property type: permission, prohibition or obligation. These three algorithms are based on the same concept: each line in the trace table can correspond to (i.e. can be an instantiation of) one or many atomic actions described in one or many properties.

5.3 Properties Checking Algorithms

In this section we describe the general idea of the properties checking algorithms and provide in particular the overview of the algorithm verifying the prohibition properties on a network traffic trace.

Prohibitions Handler: The algorithm that allows checking prohibition properties begins first by parsing the trace table (build from the trace file) line by line to check if any context of any prohibition property is verified. For each line L, it verifies if L is an instantiation of an action A described in the context of the prohibition property Pr. If it is the case, it checks if the the chronological order of the actions described in this context is verified (using the procedure *Check_Context*), then it can deduce if the whole context is verified or not. If the context is verified, the algorithm has to ensure that the action described in the first part of the prohibition rule (the prohibited action) is not present in the trace. If it finds such action (using *Check_Prohibited_Activity* procedure), the verdict is FAIL. Otherwise, it concludes that the current rule is verified, the verdict in this case is: PASS. If the trace length is not long enough to ensure the verification, the output verdict is INCONCLUSIVE. The algorithm 1 presents the pseudo-code of the procedure used to check the prohibition properties on a trace and deduce the appropriate verdict. For each property Pr , we define 'Pr.action' as the prohibited action of the property and 'Pr.context' as the context of the property. 'Pr.action' (respectively 'Pr.context') is composed of one or many chronologically ordered actions 'Pr.act.action$_i$' (respectively 'Pr.context.action$_j$') where i (respectively j) is the number of atomic actions in the prohibited action (respectively context).

Permissions Handler: The permission to perform an action in a particular context does not mean that action must be systematically executed when this context is verified. In the case of checking permission properties, we first look in the traces file (the trace table) if the permitted activity exists; then, we ensure that the context was true to conclude that the property is respected (verdict PASS), otherwise the verdict is FAIL. If the trace is not long enough to check the context, the verdict is INCONCLUSIVE.

Obligations Handler: For obligation properties the approach is very similar to that used for testing prohibition properties. We start first by checking whether

Algorithm 1. Prohibition Properties Handler

```
Require: PPS[Pr] : Prohibition Properties Set + Tr[l] : the trace table.
 1: for each property Pr of PPS do
 2:    Context(Pr) = 'not verified'
 3: end for
 4: for each line l of Tr do
 5:    for each property Pr of PPS do
 6:       if (Context(Pr)='verified') then
 7:          verdict[Pr] := INCONCLUSIVE
 8:          if (Prohibition deadline Reached) then
 9:             verdict[Pr] := PASS
10:             Context(Pr)='not verified'
11:          else
12:             if (l=instantiation(Pr.act.action_i)) then
13:                verdict [Pr] := Check_Prohibited_Action (Pr.action)
14:                if (verdict [Pr] := 'FAIL') then
15:                   Memorize error and position in the trace
16:                   Context(Pr)='not verified'
17:                else
18:                   Memorize verified parts of the prohibited activity /* (in this case
                      verdict [Pr] := 'INCONCLUSIVE') */
19:                end if
20:             end if
21:          end if
22:       end if
23:       if (l=instantiation(Pr.context.action_i)) then
24:          Context(Pr) = Check_Context(Pr.context)
25:          if (Context(Pr) = 'verified') then
26:             Calculate prohibition deadline
27:          else
28:             if (Context(Pr) = 'not yet verified') then
29:                Memorize verified parts of the context
                   /* (Context (Pr) = 'not yet verified' if some actions of the context are
                   verified and are in the right chronological order. But the whole context
                   is not yet verified. We have to check next messages in the trace, to
                   deduce if the tested system is in the right context or not.) */
30:             else
31:                Erase memorized parts of the context if exist
                   /* (This is case when the context is no more verified) */
32:             end if
33:          end if
34:       end if
35:    end for
36: end for
```

the context of the property is verified. Then, we check if the action specified in the first part of the property (mandatory action) is present in trace. If it is the case, the verdict is PASS otherwise it is FAIL. If the trace is not long enough, the verdict is INCONCLUSIVE.

5.4 Irregular Node Determination

Once a property violation is detected, the monitor has to analyze the source of the violation in order to deduce the irregular node. The methodology of this determination is the following:

- Identification of the corresponding trace section: a violation is in general due to some messages in the global trace that does not respect a given property.
- Identification of the nodes implicated in a detected violation: in the case of a message reception related violation, the node claiming the reception, the assumed sender and its neighbors are implicated. In the case of an emission related violation, the assumed sender node and its neighbors are implicated.
- Identification of the implicated trace part: going backward in the trace from the position of the message causing the property violation to extract the messages related to the nodes implicated in the violation. The number of extracted messages depends on the studied protocol. In wireless networks, messages can be lost because of the interference and collisions problem. For this reason, ad hoc protocols like OLSR and AODV wait a certain number of periods before announcing a link break. In our study, we go backward in the trace for a certain period that guarantees the protocol convergence. For example, OLSR waits 3 periods of 2 seconds each before announcing a link break with a neighbor from which he has not received Hello messages. To guarantee that OLSR has converged (i.e. the link break is advertised) we go backward one more period; this means we extract the messages exchanged in the last 8 seconds.
- Construction of coherent nodes sets: the extracted trace part is analyzed to detect coherent and non coherent nodes within those implicated in the violation. We compare each pair of implicated nodes to detect if they are coherent or not. The set with the highest number of nodes is considered as the regular set whereas the remaining set (or sets) contain the irregular nodes. We assume that the number of irregular nodes in the network is lower than the number of regular nodes in all the broadcast regions.

6 Case Study: OLSR

We tested our methodology on OLSR ad hoc routing protocol in an open area network. We started first by extracting from the RFC some OLSR properties that we described in Nomad formal language. Then we changed in NS2 the behavior of OLSR in order to model typical attacks against OLSR like Hello message poisoning, link spoofing and black hole attack. We added in NS2 a special module that allows each node to collect its local network trace. This module gives the attacker the possibility to alter its local trace. A standalone module is also developed to correlate the collected local traces and analyze the obtained global trace using the algorithms presented in the previous sections.

We run a simulation with 100 mobile nodes located in a topology of 1500x1500 for 1200 seconds. Among these nodes, 5 are attackers and 2 of them can alter

their local trace to simulate collaborative attack. In total 20 different attacks were launched. The simulation provided us the local traces that the standalone module correlated and analyzed. The global trace was around 5 million of lines. The analysis of the global trace gave 21 fail and 2 inconclusive verdicts. The inconclusive verdicts are due to incomplete execution trace due to multiple link breaks. The 21 fail verdicts correspond to the attacks and one false negative due to nodes mobility. In the next subsections, we emphasize on 2 of these attacks:

6.1 Hello Messages Poisoning

One of the first properties to check is the correct logical order of HELLO messages exchange. That is a node cannot announce a symmetrical link to any neighbor without having previously received a HELLO message claiming an asymmetric link from that node. The connectivity establishment process must respect the following properties:

- $Pr1 : \mathcal{F}\ (start\ (n\ ?\ Hello(n : Asym)I) \text{ — } O^{\leq 2sec} \neg\ done(n\ !\ Hello()*))$
- $Pr2 : \mathcal{F}\ (start\ (n?Hello(n : Sym)I) \text{ — } O^{\leq 2sec}\ (\neg\ done(n!Hello(I : Asym)*) \wedge \neg\ done(n!Hello(I : Sym)*)))$
- $Pr3 : \mathcal{F}\ (start\ (n?Hello(n : MPR)I) \text{ —} O^{\leq 2sec} \neg\ done(n!Hello(I : Sym)*))$

In figure 2, node I sends a Hello message claiming a symmetrical link to node A after receiving an empty Hello from it. In addition to this protocol violation I may insert a fake entry in its trace claiming the reception of an asymmetrical Hello message from A. In both cases, our methodology detected the violation:

1. I has not changed its local trace: In this case I violates the property $Pr2$. We can conclude that I is the malicious node.
2. I changed its local trace by claiming the reception of an asymmetrical Hello message from A. In this case the trace violates the property $Pr4$ which indicates that a message must have been emitted in order for a node to receive it.

- $Pr4 : \mathcal{O}\ (\ominus\ done\ (Node_1\ !\ M(p)\ Node_2) \text{ —} start\ (Node_2\ ?\ M(p)Node_1))$

6.2 Link Spoofing with Distant Node

In figure 2, we illustrate an example of a link spoofing attack on OLSR. The intruder I can insert Hello messages claiming a non existing symmetrical link to

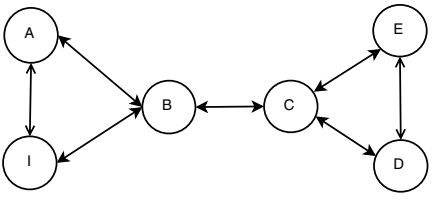


Fig. 2. A distant Link Spoofing Attack on OLSR

C. Consequently, the intruder might be selected as a MPR by A and the traffic from A to C will be disrupted to the intruder. If we analyze the global traffic in this part of the network, we notice one of these two cases:

1. Node I has not changed its local trace: Node I can not claim a symmetrical link to C according to the protocol specification violating thus property $Pr2$. We can conclude that I is the malicious node.
2. Node I has changed its local trace to claim the reception of a Hello message M from C specifying a symmetric link. Here, the trace violates the property $Pr5$ which indicates that if a node C receives a message from node N, all the symmetric neighbors of N ($V_S(N)$) must have received the same message. Therefore, we are in a message reception related violation; node I claiming the reception, the assumed sender C and its neighbors B, D and E are implicated. We split these nodes into two sets $\{I\}$ and $\{B, D, E\}$, the first claims the reception of the Hello message from node C where this message does not appear in the traces of the nodes in the second set. We can conclude that I is the irregular node. We note again that we are assuming that the number of irregular nodes is lower than that of regular ones in any neighborhood.

– $Pr5 : \forall\ B \in V_S(N),\ \mathcal{O}\ (done\ (B?M(p)N) \text{ — } done\ (C?M(p)N))$

We highlight here that this property expresses a distributed network behavior that allows to detect distant attacks. This detection can only be made through checking the global trace.

7 Conclusions and Future Work

This paper proposes a distributed monitoring approach to detect functional and security flows in ad hoc networks. It considers two types of networks : an open area network and a controlled area network. Dedicated observers collect the local network traffic in a controlled area network whereas this collection is performed by the nodes themselves in an open area network. In both cases, the local traces are sent to a global observer. This latter is responsible for the local traces correlation and their analysis. The correlation is performed based on an accurate synchronization mechanism designed for ad hoc networks.

Our analysis rely on two main features : (1) functional and security properties specified using an instantiation of Nomad model, and (2) a correlated trace of the network traffic. Based on dedicated algorithms, we prove that our methodology allows to detect a large range of flows and errors.

As future work, we are investigating several approaches to improve the passive testing algorithms in order to perform online monitoring, possibly by including vulnerability cause graphs [4] of the implementation under test. We are also studying the different reactions that the network has to perform following a property violation detection.

References

1. http://wipal.lip6.fr/index.html
2. Alcalde, B., Cavalli, A.R., Chen, D., Khuu, D., Lee, D.: Network protocol system passive testing for fault management: A backward checking approach. In: Núñez, M., Maamar, Z., Pelayo, F.L., Pousttchi, K., Rubio, F. (eds.) FORTE 2004. LNCS, vol. 3236, pp. 150–166. Springer, Heidelberg (2004)
3. Badonnel, R., State, R., Festor, O.: Monitoring end-to-end connectivity in mobile ad-hoc networks. In: ICN (2), pp. 83–90 (2005)
4. Byers, D., Ardi, S., Shahmehri, N., Duma, C.: Modeling software vulnerabilities with vulnerability cause graphs. In: ICSM, pp. 411–422 (2006)
5. Chakers, I., Perkins, C.: Dynamic manet on-demand (dymo) routing. IETF Internet-Draft draft-ietf-manet-dymo-06 (work in progress) (October 2006)
6. Clausen, T., J., P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized link state routing protocol (OLSR). RFC 3626, Network Working Group (October 2003)
7. Cuppens, F., Cuppens-Boulahia, N., Sans, T.: Nomad: A security model with non atomic actions and deadlines. In: CSFW, pp. 186–196 (2005)
8. Lee, D., Yannakakis, M.: Principles and methods of testing finite state machines - A survey. In: Proceedings of the IEEE, vol. 84, pp. 1090–1126 (1996)
9. Orset, J.-M., Alcalde, B., Cavalli, A.R.: An EFSM-based intrusion detection system for ad hoc networks. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 400–413. Springer, Heidelberg (2005)
10. Orset, J.-M., Cavalli, A.R.: A security model for olsr manet protocol. In: MDM, p. 122 (2006)
11. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing. RFC 3561 (July 2003)
12. Ploskonka, J.A., Hurson, A.R.: Self-monitoring security in ad hoc routing. In: Kunz, T., Ravi, S.S. (eds.) ADHOC-NOW 2006. LNCS, vol. 4104, pp. 238–251. Springer, Heidelberg (2006)
13. Ramachandran, K., Belding-Royer, E.M., Almeroth, K.C.: DAMON: A Distributed Architecture for Monitoring Multi-hop Mobile Networks. In: Proceedings of the 1st IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON) (October 2004)
14. Wehbi, B., Laouiti, A., Cavalli, A.: Accurate and efficient time synchronization mechanism for wireless multi hop networks. Technical report, TELECOM SudParis (2008)

Improved Distributed Dynamic Power Control for Wireless Mesh Networks*

Thomas Olwal[1,2], Felix Aron[2], Barend J. van Wyk[2], Yskandar Hamam[2], Ntsibane Ntlatlapa[1], and Marcel Odhiambo[3]

[1] Meraka Institute at the CSIR
P.O. Box 395 Pretoria 0001, South Africa
thomas.olwal@gmail.com, nntlatlapa@csir.co.za
[2] The French South African Technical Institute in Electronics at the Tshwane University of Technology
jakajode@gmail.com, vanwykb@gmail.com, hamama@tut.ac.za
[3] The University of South Africa
ohangmo@unisa.ac.za

Abstract. One of the main objectives of transmission power control (TPC) in wireless mesh networks (WMNs) for rural area applications is to guarantee successful packet transmission and reception (SPT-R) with low power consumption. However, the SPT-R depends on co-channel multiple access interferences (MAI) including the effects from hidden terminals. In this paper we investigate how MAI can be minimized through a MAC-dependent transmission scheduling probability (TSP) model. In what follows, we show how a distributed scheduling probability model improves the dynamic power control algorithm. The resulting optimal power control is derived from a network centric objective function. The analytical results show that transmit power solutions converge to a unique fixed point. The simulation results show that a high average feasibility rate, given a coexistence pattern, can be achieved. There is significant average transmission power savings compared to conventional methods.

Keywords: Energy-constrained mesh nodes, MAC-DDPC algorithm, Transmission scheduling probability, WMNs.

1 Introduction

Wireless Mesh Networks (WMNs) deployed in rural areas suffer from battery power limitations. However, due to architectural complexities and high capacity requirements, conventional power control solutions proposed for cellular, Ad Hoc and sensor networks may not be alternatives for WMNs [1]. In this work we consider the problem of power control for energy-constrained distributed mesh nodes (MNs) for rural community applications [2]. We focus on a distributed transmission power control (DTPC) policy in which power is adjusted in response to cross-layer feedback

* This work is supported by the Meraka Institute at the Council of Scientific and Industrial Research (CSIR), Pretoria, South Africa.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 357–368, 2008.
© Springer-Verlag Berlin Heidelberg 2008

information [5]. The DTPC policy allows MNs to setup and maintain stochastic wireless links with minimum power while satisfying constraints on the quality of service (QoS). The benefits of power minimization are not only increased battery life but also the mitigation of effective multiple access interference (MAI). Consequently, the overall network capacity can also be increased by allowing higher frequency reuse [2].

Traditional distributed QoS-based approaches for power control has been researched for an uplink power control problem in cellular systems [3], [6], [15]. However, most of the approaches are deemed greedy in which transmission power is adapted by an individual node with sole objective of maintaining QoS target metrics during a communication session [3], [15]. Though suitable for delay sensitive applications, such approaches may lead to high energy expenditure. The work in [6] presents power control policies that address various node-centric and network centric objectives adapting power in either a greedy or an energy efficient manner. However, the work assumes a special case where all sender nodes communicate to a centralised base station in a CDMA system. In practice, some nodes may become active or inactive during the course of a frame transmission. Thus, a distributed power control model for such dynamic behaviour in an Ad Hoc fashion would be necessary.

2 Related Work

Recent research focussed on the application of autonomous power control in infrastructure-less Ad Hoc networks [12], [7], [13]. Most of these schemes use maximum transmit power for RTS-CTS and the minimum required transmit power for DATA-ACK transmissions in order to save energy. The work in [8] presents a power control MAC protocol that allows nodes to vary transmission power level on a per-packet basis. Simulation results in [8] show that schemes in [12], [7] can degrade network throughput [2] and result in higher energy consumption than in the case of no power control. Furthermore, conventional CSMA/CA systems demonstrate low network capacity and scalability properties. Such performances are undesirable for large-scale mesh network deployments [1]. In [9], the authors present a joint scheduling and power control strategy supporting multicasting traffic. The process of power control entails the elimination of weak connections while maximizing the number of successful simultaneous transmissions and still achieving minimum total transmit power. However, the contribution does not guarantee power control solutions for hidden terminal problems.

Our paper presents a power optimization problem similar to the work by Sorooshyari and Gajic [6]. However, the authors assumed single hop communications in CDMA cellular systems. In the context of CSMA/CA protocol, we propose a distributed transmission scheduling probability (TSP) based power control model. Through bidirectional information exchange among nodes, we show that that cross-layer power control model yields several advantages. First, the QoS at the receivers can still be maintained at low transmission power consumptions when both the channel and multiple transmission activity (MTA) of the network are known to the power control system. While the physical (PHY) layer encodes the signalling to overcome the channel impairments, the MAC protocol provides scheduling disciplines for the MTA in a

shared channel. Second, the transmission power control based on a clear channel assessment (CCA) reduces the probability of traffic retransmissions. Retransmissions result in additional power consumptions and cause excessive network delays. Finally, the distributed dynamic power control (DDPC) algorithm with the knowledge of the network topology may improve performance metrics on routing decisions for multiple hop communications.

The paper is organised as follows: Section 2 presented the related work, while section 3 analyses the cross-layer probability model. Section 4 formulates the problem. In section 5, an adaptive transmission power control algorithm is developed. Section 6 presents and analyses the simulation results. Section 7 concludes the paper.

3 Cross-Layer Probability Model

Basic Formulations and Assumptions: Consider an N stationary mesh nodes (MNs) network randomly distributed in a space S. Let us assume that each MN is equipped with omni directional antenna with carrier sensing range (CSR) at least twice larger than the transmission range (TR) [8]. Thus, the resulting wireless network can be modelled as a graph $G = (V, E)$ where V represents a set of nodes in the network and $E \subseteq V \times V$ the edge set which gives the available communications: $(i, r) \in E$ if node i can send messages directly in one hop to node r and vice versa. Let $V_r \subseteq V$ and $V_i \subseteq V$ be the two subsets of nodes whose signal powers can be perceived by nodes r and i respectively. We have $|V_r| = N_r$ and $|V_i| = N_i$ nodes, respectively in the sets V_r and V_i. In practice, the wireless links (channels) between nodes i and r or among any other nodes are typically subjected to large-scale path loss, shadowing and possibly small scale multi path fading dynamics [11]. This implies that the time-variant channel gain function can be denoted as $g_i^r(k)$ for any $i \in V_i$ and any $r \in V_r$. If we consider that a sender node i chooses a transmission power level $l(k)$ from a finite set $\mathbf{L}_i = (1, 2, 3, \ldots l \ldots, \ell_i)$, containing ℓ_i power levels then, the actual transmission power value corresponding to the lth power level is given by $p_i(l, k)$. Consider that the transmission power vector $\mathbf{p}$ is constrained as

$$\mathbf{p}^{\min} \le \mathbf{p} \le \mathbf{p}^{\max} \qquad \forall i \in V_i \ , \tag{1}$$

where $\mathbf{p} = [p_1(l,k) \quad p_2(l,k) \quad \ldots \quad p_N(l,k)]^T$.

Thus, at a receiving node r, the received power due to the transmission from the sender node i is given by

$$p_i^r(k) = g_i^r(k)\, p_i(l,k) \tag{2}$$

Scheduling Probability Model: Let us consider the spread-spectrum channel signalling system for the MNs [5]. Such signalling methods provide anti-jamming capabilities, robustness to multi path effects and potential for multi user access through CDMA techniques. In spread-spectrum systems supported by the IEEE 802.11

standard, DDPC methods are affected by MAI powers at the receiver node [6]. The MAI powers due to other users' concurrent transmissions degrade the quality of transmission and reception. This implies a scheme to schedule multiple transmissions such that significant mesh network capacity is guaranteed. Let's consider that in a distributed MAC protocol and in the context of power control, the sender node desires to minimize the number of retransmissions of its packets. To achieve this, the node must perform CCA in order to guarantee successful transmission with low power consumption. Furthermore, the need for bidirectional channel signalling information can significantly reduce collisions caused by MAI during transmission attempts. Thus, the transmission power for each packet from node i must overcome the MAI level at node r [8]. However, due to channel dynamics and heterogeneity of the wireless devices, MAI levels may change during the transmission of the packet and a model to generalize such change is desirable. The instantaneous interference plus receiver noise (II+N) at node r as defined by

$$q_{-i}\left(k,\mathbf{p}_{-i}\right) = \sum_{j\in V_r,\ j\neq i} x_j\left(k\right).g_j^r\left(k\right)\,p_j\left(l,k\right)+\eta_r \ . \tag{3}$$

Here, η_r denotes the thermal noise power at the receiver node $r\in V_r$, while $x_j\left(k\right)$ is a binomially-distributed random variable that dictates the number of nodes in the set V_r and V_i that are transmitting concurrently and whose CSRs the receiver happens to fall. Let $x_j\left(k\right)$ be a binomially-distributed random variable with probability of occurrence ρ_j for all $j\in V_r\cup V_i$. Thus, the binomially-distributed random variable $x_j\left(k\right)$ may be defined as

$$x_j\left(k\right)=\begin{cases}1 \ if \ \ j \ transmits \ at \ time \ k\\ 0 \qquad\qquad\qquad\qquad otherwise\end{cases}. \tag{4}$$

The information on the number of MTA must be known by a scheduling and a power control system. If the number of MTA at node r is $\left|V_r\right|=N_r$, then there are exactly 2^{N_r-1} possible combinations of MTA in the set V_r excluding the transmitting node itself at any given time. Sets of such combinations of MTA can be denoted as $\left\{\phi_{in}^r\right\}_{n=1,\ldots,2^{N_r-1}}$ [13], [10]. Correspondingly, we can define a random variable $\Phi_i^r\left(k\right)$ which indicates the occurrence of a specific combination $\phi_{in}^r\left(k\right)$ of independent interferers, interfering with node i's transmission at a certain time k. Thus, the probability that $\Phi_i^r\left(k\right)$ assumes the value of $\phi_{in}^r\left(k\right)$ for the nth combination of independent interferers can be defined as

$$\Pr\left\{\Phi_i^r\left(k\right)=\phi_{in}^r\right\}=\prod_{m\in\bar{\phi}_{in}^r}\left(1-\rho_m\right)\prod_{l\in\phi_{in}^r}\rho_l \ . \tag{5}$$

Here, $\bar{\phi}_{in}^r$ of the first product term denotes the compliment of ϕ_{in}^r in the second product term. That is, the first term of the product function is the probability describing nodes which are *not* transmitting with sender node i at time k. On the other hand, the second term refers to the probability of those *actively* transmitting with node i.

Considering the definition in eq (5), assuming unicast traffic and dropping the time index k for simplicity reasons, the probability υ_i^r that a channel-assessment packet transmitted with power $p_i(l)$ by the node i is successfully received at the node r conditioned on certain MAI levels is given as:

$$\begin{aligned}\upsilon_i^r &= \Pr\{\textit{successful packet reception at } r\} \\ &= \sum_{n=1}^{2^{N_r-1}} \Pr\{\textit{succ. pac. recept.}|\Phi_i^r = \phi_{in}^r\}\Pr\{\Phi_i^r = \phi_{in}^r\} \\ &= \sum_n f(\phi_{in}^r)\Pr\{\Phi_i^r = \phi_{in}^r\}.\end{aligned} \tag{6}$$

Here, $f(\phi_{in}^r)$ denotes the probability of successful packet reception by node r due to transmission of node i, conditioned on a certain MAI level. The functional form of $f(\phi_{in}^r)$ depends on the specific choice of the PHY-layer aspects such as wireless channel model, modulation and demodulation schemes, channel coding and the receiver designs. If we assume that the forward and backward transmissions are independent then, the joint probability of successful reception of packets at the nodes i and r can be given as follows:

$$\begin{aligned}\upsilon_i &= \Pr\{\textit{forward success}, \textit{ backward success}\} \\ &= \sum_n^{2^{Nr-1}} \sum_l^{2^{N_i-1}} f(\phi_{in}^r) f(\phi_{rl}^i) \Pr\{\Phi_i^r = \phi_{in}^r\}\Pr\{\Phi_r^i = \phi_{rl}^i\}.\end{aligned} \tag{7}$$

The MAC protocol in place exploits the PHY-layer signalling information in eq (7) and the interaction among other nodes in the topology to determine adaptive scheduling rules for actual application packet transmissions. This can be done in a way that minimises the number of unsuccessful transmissions. Such MAC-dependent functional may take the form $\rho_i = \xi_i(\upsilon_i)$. In general, this functional is a non-linear model and related analysis is complex. In linear representation, $\zeta_i(\upsilon_i)$ can be assumed to have an nth derivative throughout the interval $[0,1]$ such that the Maclaurin series expansion is given as

$$\rho_i = \xi_i(\upsilon_i) = \xi_i(0) + \upsilon_i \xi_i^{'}(0) + \ldots + \frac{\upsilon_i^{n-1}}{(n-1)!}\xi_i^{(n-1)}(0) + \frac{\upsilon_i^n}{n!}\xi_i^{(n)}(\varepsilon), \tag{8}$$

where $0 \le \varepsilon \le \upsilon_i$. The first order approximation of eq (8) is given by

$$\rho_i \approx m\upsilon_i, \text{ where } m = \xi^{'}(0), \text{ at } \upsilon_i = 0. \tag{9}$$

Here, m is a time-varying proportionality design factor for the linear model in eq (9). This proportionality factor relates the PHY-layer successful packet reception probability (PRP) υ_i with the MAC-dependent TSP, ρ_i at any given time. For design

purposes m, can be chosen to be $m << 1$ since in Maclaurin expansion series, the conditional successful PRP $\upsilon_i = 0$ when the number of MTA becomes very large.

4 Problem Formulation

If we consider that each sender node i, desires that it's SINR QoS degradation and the aggregate network MAI to be minimal, then a corresponding convex cost function can be given as in [6]

$$J_i(k) = \omega_{i1}\,\varepsilon_i^2(k+1) + \omega_{i2}\,q_i^2(k+1). \tag{10}$$

In eq (10), the first term describes the action taken by an individual node in order to achieve its own target quality of service (QoS). That is, how the received SINR $\gamma_i(k)$ deviates from the SINR threshold $\bar{\gamma}_i$. On the other hand the second term in eq (10) describes a network-centric cost function i.e., how the action of the transmitting node impacts on the other network users. As explained in [6], these terms can be defined as follows: $\varepsilon_i(k+1) = \bar{\gamma}_i - \gamma_i(k+1)$, where

$$\gamma_i(k+1) = \frac{p_i(l,k+1)g_i^r(k+1)}{q_{-i}(k+1)+\eta}, \quad \text{and from eq (10)} \tag{11}$$

$$q_i(k+1) = q_{-i}(k+1) + p_i(k+1)g_i^r(k+1). \tag{12}$$

The expression in eq (12) represents the predicted aggregate interference powers that impact significantly on any receiving node in the network. The reliability of $\rho_i(k)$ depends on the simultaneous transmissions within an interference range of each link as shown in eq (5). However, the value $\rho_i(k)$ dictates the activity state of the random variable $x_j(k+1)$ in the next power update step in a manner that network MAI levels are minimised. Thus, the iterative power control system is given as

$$p_i(k+1) = p_i(k) + \alpha_i(k)q_{-i}(k), \text{ subject to: } \quad p_i^{\min} \le p_i(k+1) \le p_i^{\max}. \tag{13}$$

In this formulation a unique fixed point p^* can be achieved if the adaptive control gain $\alpha_i(k)$ can be optimum for all $i \in V$ in the network. This optimum point can be derived from: $\alpha_i^*(k) = \arg_{l \in V} \min J_l(k)$. The outline of the derivation follows: If we substitute the value of $p_i(k+1)$ in eq (11) and eq (12) with the expression in eq (13) and evaluate the first partial derivative of eq (10) with respect to $\alpha_i(k)$, and set the result to zero, we get

$$\alpha_i^*(k) = \frac{\left(\bar{\gamma}_i - \gamma_i(k) - \frac{\omega_i}{q_{-i}(k)}\left\{q_{-i}^3(k+1) + g_{ii}(k)p_i(k)q_{-i}^2(k+1)\right\}\right)}{g_{ii}(k) + \omega_i}. \tag{14}$$

Here, $\omega_i = \omega_{i2}/\omega_{i1} \geq 0$ is a non-negative power control strategic-weight. The strategic-weight ω_i and the MAC-dependent $\rho_i(k)$ are locally assigned to each node depending on the channel states and traffic applications [6], [10]. Using matrix notations and considering the MTA of the network we have

$$\mathbf{p}(k+1) = (\mathbf{I} + \mathbf{A})\mathbf{p}(k) + \mathbf{b}\,, \tag{15a}$$

$$\text{Subject to: } \mathbf{p}^{\min} \leq \mathbf{p}(k+1) \cong \Gamma(\mathbf{p}) \leq \mathbf{p}^{\max}\,. \tag{15b}$$

Here,

$$\mathbf{A} = \begin{bmatrix} \alpha_1^* g_{11}(k) & \alpha_1^* g_{12}(k) & \alpha_1^* g_{13}(k) & \dots & \alpha_1^* g_{1N}(k) \\ \alpha_2^* g_{21}(k) & \alpha_2^* g_{22}(k) & \alpha_2^* g_{23}(k) & \dots & \alpha_1^* g_{2N}(k) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_N^* g_{N1}(k) & \alpha_N^* g_{N2}(k) & \alpha_N^* g_{N3}(k) & \dots & \alpha_N^* g_{NN}(k) \end{bmatrix}$$

$$g_{ir}(k) = g_i^r(k) \quad \forall i, r \in V\,,$$

and $\mathbf{b} = \begin{bmatrix} \alpha_1^*\eta & \dots & \dots & \alpha_N^*\eta \end{bmatrix}^T$.

Theorem 1. If the optimal gain vector $\boldsymbol{\alpha}^*$ is unique then it implies that power update function $\Gamma(\mathbf{p})$ has a unique fixed point at the optimal power vector $\mathbf{p}^*$.

Proof by contradiction: Suppose $\boldsymbol{\alpha}^a$ and $\boldsymbol{\alpha}^b$ are two distinct fixed points at a and b for all $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ at the same time. Thus, from eq (15), the following properties can be defined:

- $\|\mathbf{I} + \mathbf{A}\| \leq \|\mathbf{I}\| + \|\mathbf{A}\|$, *(Triangle Inequality)*
- $\left(\|\mathbf{I}\| + \|\boldsymbol{\alpha}\|^T \|\mathbf{G}\|\right) > \|\mathbf{0}\| \quad 0 < g_{ij}(k) \leq 1$,
- $\|\mathbf{b}\| > \|\mathbf{0}\| \quad$ *where* $\eta > 0$, *(non zero)*
- $\Rightarrow \alpha_i \neq 0 \quad \forall i = 1, 2, \dots N$. *(non zero)*
- $f(\boldsymbol{\alpha}^a) \geq f(\boldsymbol{\alpha}^b)$ if $\boldsymbol{\alpha}^a \geq \boldsymbol{\alpha}^b$, *(Monotonicity)*
- $\delta f(\boldsymbol{\alpha}) > f(\delta\boldsymbol{\alpha}) \quad \forall \delta > 1$, *(Scalability)*

Let us assume that there exists j such that $\alpha_j^a < \alpha_j^b$ for all j. Correspondingly there exists $\delta > 1$ such that $\delta\boldsymbol{\alpha}^a \geq \boldsymbol{\alpha}^b$. Thus, there exists for some j, $\delta\boldsymbol{\alpha}^a = \boldsymbol{\alpha}^b$. The monotonicity and scalability implies:

$$f_j(\boldsymbol{\alpha}^b) = \alpha_j^b \leq f_j(\delta\boldsymbol{\alpha}^a). \tag{16}$$

$$f_j(\delta\boldsymbol{\alpha}^a) = \delta\alpha_j^a < \delta f_j(\boldsymbol{\alpha}^a). \tag{17}$$

The result in equations (16) and (17) implies that a and b are two distinct points. Thus, there can be no more than one solution of $\boldsymbol{\alpha}^*$ at the same time. Furthermore if the wireless channels hold their states in the duration of the power control, then $\boldsymbol{\alpha}^*$ can be unique with exact solution as shown in eq (14). From theorem 1, having shown that $\boldsymbol{\alpha}^*$ is unique then the *proof* that $\Gamma(\mathbf{p})$ has a unique fixed point at $\mathbf{p}^*$ can be found in [3]. However, uniqueness of $\mathbf{p}^*$ does not necessarily imply feasibility of the power vector $\Gamma(\mathbf{p})$ in a contention based and a distributive WMN environment [9]. In such situations, the TSP aware dynamic power control algorithm becomes necessary. That is, each sender aware of the TSP may decide whether to transmit at a certain time using a controlled power in a manner that the aggregate MAI component of the objective function in eq (10) is minimised. The remaining sender nodes can then attain feasible power solutions via the execution of transmission power iterations i.e., $\mathbf{p}(0) \geq \mathbf{p}(1) \ldots$ if $\mathbf{p}(0) > 0$. Hence, the feasibility implies monotonicity [14].

Lemma 1. If $\mathbf{p}$ is a feasible power vector for all nodes, then $\Gamma(\mathbf{p})$ is a monotonically decreasing sequence of feasible power vectors that is lower bounded by the minimum power and $\Gamma(\mathbf{p})$ converges to a unique fixed point $\mathbf{p}^*$ [14]. Conversely starting from $\mathbf{p}(0)=0$, then $\Gamma(\mathbf{p})$ is a monotonically increasing sequence of power vectors that is upper bounded by a unique fixed point $\mathbf{p}^*$.

The *proof* is developed in [3] and extended in [9].

5 Adaptive Power Control Algorithm

This study presents a scalable CCA model according to the given MAC protocol at any time. Based on the bidirectional and reliable feedback information, the DDPC algorithm is outlined as follows:

1) Each node, say node i, measures its thermal noise η_i.
2) Each node, say node i, draws an independent uniform random variable to select an initial channel assessment power level. If an integer parameter Q represents the total number of power levels to which a transmitter can be adjusted in practice then,

$$\mathbf{p}_{uniform}(0) = \left\{ \frac{1}{Q} p_i^{\max}, \frac{2}{Q} p_i^{\max}, \ldots, p_i^{\max} \right\}. \tag{18}$$

3) Each node, say node i, measures its direct channel gain to any receiver; say node r, i.e., $g_i^r(k) = g(e,k)$ as given in [4], where $e \in E$ is the link between node i and node r.
4) Each link, say link $e \in E$, evaluates the MAI predictive procedure proposed in [4].

5) Each link, say link $e \in E$, computes its time-varying signalling information on the transmission scheduling probability (TSP), i.e., $\rho_i(k)$ as in eq (9).
6) The joint CCA and the adaptive power control algorithm can be given as

$$\text{If } \rho_i(k) = \begin{cases} 0 \quad then \quad p_i(k+1) = p_i^{\min} \\ 1 \quad then \quad p_i(k+1) = p_i^{\max} \\ otherwise \quad then \quad p_i^{\min} < p_i(k+1) < p_i^{\max} \end{cases} . \tag{19}$$

The advantage of the algorithm is that it provides a correction mechanism to the problem of greedy algorithms. That is, any node i experiencing $\rho_i(k)=0$ will go on power-save mode while causing no interference to actively transmitting node $j \in V_r \cup V_i$ in state $x_j(k+1)$. Conversely, with $\rho_i(k)=1$, the sender node i can transmit with up to maximum power taking advantage of the favourable link condition. However, due to the inherent interference caused to the network, the network may become disconnected and the sender i gets discouraged in the long run. Node i then executes the optimal power iteration procedure discussed in this paper.

6 Simulation Results

For simulations, we used MATLAB™ version 7.1. We placed collections of 5 to 50 nodes randomly within a 1000 x 1000 m^2 area, i.e., a size big enough to deploy a multi-hop network. Performance metrics were evaluated by Monte Carlo simulations for 50 independent runs for each random network configuration (instance). It was assumed that every node has a maximum transmission power (P*max*) of 500 mW and a minimum transmission power (P*min*) of 0 mW. The propagation path loss model exponent and a white Gaussian noise (AWGN) were also assumed to be 4 and 0.001mW respectively.

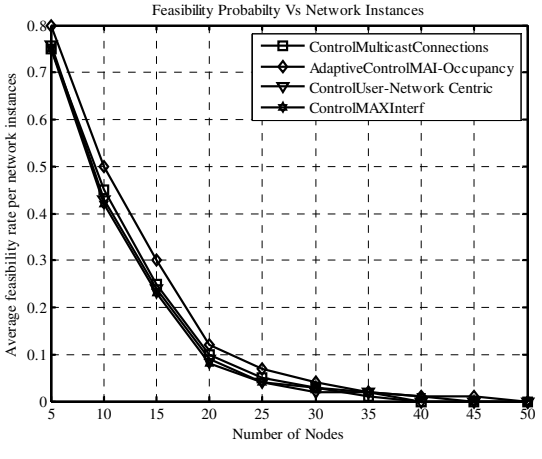


Fig. 1. Feasibility probability versus number of senders

Figure 1 shows an average feasibility rate per network scenario versus number of admitted sender nodes. The average feasibility rate (feasibility probability) indicates how many senders can be active simultaneously in a specific area without causing MAI significantly i.e., a case when the power vector $\Gamma(\mathbf{p})$ converges to a unique fixed solution $\mathbf{p}^*$. Infeasibility implies that no successful transmission can be obtained and the transmission power vector $\Gamma(\mathbf{p})$ does not convergence to $\mathbf{p}^*$ in the long run. As shown in Fig. 1 the feasibility probability drops sharply as the number of simultaneous active senders increases. However, the TSP based DDPC algorithm (*Adaptive Control MAI-Occupancy*) can accommodate slightly more nodes than some recently proposed algorithms [9], [6]. This is significant in improving the WMN capacity.

Figure 2 shows the simulation result for a non-zero TSP ($0<\rho_i \leq 1$) incorporated in a greedy and energy-efficient DDPC method. In Fig. 2, sender 4 at the beginning of simulation adjusts its transmission power to a value minimum enough to achieve the target SINR threshold in the steady state. At later time, say after 38 seconds, sender 4 chooses to opt-out of the network participation in response to unfavourable channel conditions. Sender 1 chooses to stay active in the network throughout the power control convergence and continue to achieve the target QoS. The rest of the users remain inactive throughout the power control convergence and the transmission of a packet.

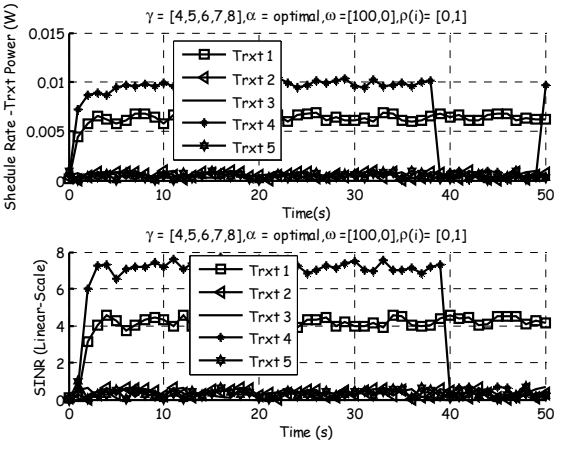


Fig. 2. Scheduled Joint greedy and energy- efficient method

In Fig. 3 a comparative performance of the average transmission power after convergence is shown. The simulation result reveals that the average transmission powers drops exponentially as the number of allowable senders increases. However, the proposed MAC-DDPC algorithm indicates much more power savings than some conventional methods [9] [6].

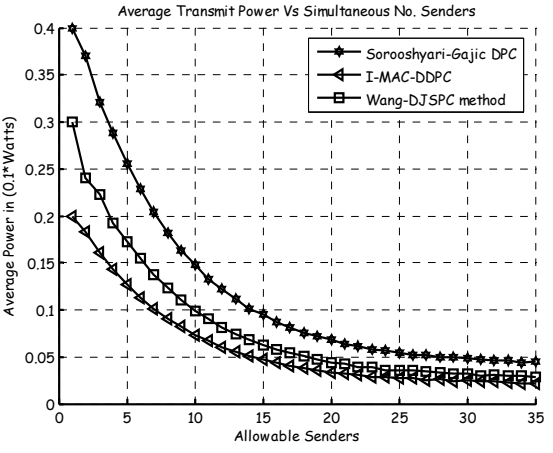


Fig. 3. Average Transmission Powers after steady state

7 Conclusion

In this paper it was shown that if TSP information is known to the power control system, improved performance of the DDPC algorithm is observed. As revealed in Fig. 1, MTA can be achieved with TSP model. More average transmission power savings than in cases of some conventional methods were noted in Fig. 3. Thus, the information exchange between the PHY and the MAC-layers can be exploited to improve the conventional power control methods. As future work, we intend to investigate the effect of MAC-DDPC on throughput performance [1].

References

1. Akylidiz, I.F., Wang, X., Wang, W.: Wireless Mesh Networks: a survey. J. Computer Networks. 47, 445–487 (2005)
2. Bambos, N.: Toward power-sensitive network architectures in wireless communications: Concepts, Issues, and Design Aspects. IEEE Personal Communications, 50–59 (1998)
3. Yates, R.D.: A framework for uplink power control in cellular radio systems. IEEE Jnl. Select. Areas in Commun. 13(7), 1341–1347 (1995)
4. Leung, K.K.: Power control by interference prediction for broadband wireless packet networks. IEEE Trans. Wireless Commun. 1, 256–265 (2002)
5. Lin, S., Zhang, J., Zhou, G., Gu, L., He, T.: Stankovic.: ATPC: Adaptive transmission power control for wireless sensor networks. In: 4th ACM Conf. Embed. Networks Sensor (2006)
6. Sorooshyari, S., Gajic, Z.Z.: Autonomous dynamic power control for wireless networks: user-centric and network-centric consideration. IEEE Trans. Wireless Commun. 7(3), 1004–1015 (2008)
7. Agarwal, S., Katz, R.H., Krishnamurthy, S.V., Dao, S.K.: Distributed power control in ad hoc wireless networks. In: Proc. PIMRC 2001 (2001)

8. Jung, E.S., Vaidya, N.H.: A power control MAC protocol for ad hoc networks. J. Wireless Networks 11, 55–66 (2005)
9. Wang, K., Chiasserini, C.F., Proakis, J.G., Rao, R.R.: Joint scheduling and power control supporting multicasting in wireless ad hoc networks. J. Ad Hoc Networks 4, 532–546 (2006)
10. Carvalho, M.M., Garcia-Luna-Aceves, J.J.: A scalable Model for Channel Access Protocols in Multi hop Ad Hoc Networks. In: Proc. ACM MobiCom 2004, September 26-October 1, 2004, Philadelphia, USA (2004)
11. Stuber, G.L.: Principles of Mobile Communication. Kluwer Academic Publishers, Dordrecht (2000)
12. Qu, Q., Milstein, B., Vaman, D.R.: Distributed Power and Scheduling management for mobile ad hoc networks with delay constraints. In: Milcom 2006 (2006)
13. Capone, A., Carello, G.: Scheduling optimization in wireless MESH networks with power control and rate adaptation. In: Proc. Sensor and Ad Hoc Communications and Networks 2006. SECON 2006 3rd Annual IEEE Communications Society, September 28, 2006, vol. 1, pp. 138–147 (2006)
14. Gauss, S.I.: Linear programming methods and applications, 5th edn., Minneola, New York (2003)
15. Chen, C.-C., Lee, D.-S.: A joint design of distributed QoS scheduling and power control for wireless networks. In: Proc. IEEE Infocom, Barcelona, Catalunya, Spain, April 23-29, 2006, pp. 1–12 (2006)

Identifying the Boundary of a Wireless Sensor Network with a Mobile Sink

Majid I. Khan[1], Wilfried N. Gansterer[1,2], and Günter Haring[1]

[1] Department of Distributed and Multimedia System, University of Vienna, Austria
majid@ani.univie.ac.at, guenter.haring@univie.ac.at
[2] Research Lab Computational Technologies and Applications, University of Vienna, Austria
wilfried.gansterer@univie.ac.at

Abstract. This paper summarizes an effort to evaluate the usability of a mobile sink for identifying the boundary of a wireless sensor network. In order to achieve the desired task we transform the problem of boundary identification into one of edge node identification. The algorithm designed is based on a mobile sink equipped with a directional antenna, which identifies the edge nodes and connects them to complete the boundary of the sensor field. The proposed scheme has following distinct features. Firstly, it is independent of the sensor node deployment, and therefore can be used for fields having very low node density. Secondly, it does not require sensor field flooding which helps preserving the nodes' energy. Thirdly, it works with low cost sensor nodes, i.e., it does not impose any special requirements on the hardware of individual sensor nodes (no GPS, no special antennas, etc.), which makes it cost effective.

Keywords: Boundary identification, mobile sink, directional antenna.

1 Introduction

Environmental/habitat monitoring, war field surveillance and monitoring volcanic eruptions are some application examples for wireless sensor networks which usually require ad-hoc deployment of the sensor nodes. Such deployments make it impossible to preprogram nodes with information like routing tables, boundary of the field, neighbor density, etc. This paper addresses the problem of boundary identification of a wireless sensor network. Nowak et al. state two fundamental limitations in the boundary identification process, *spatial density* of the nodes that can seriously affect the accuracy of the boundary estimation scheme, and *energy constraints* of the nodes which can limit the complexity of the boundary identification algorithm [1]. On the other hand, it has been observed that the state of the art [2, 3] imposes strong assumption regarding node placement, spatial density and communication model of the nodes which are very hard to assure during random deployment of the sensor nodes. Moreover, in large scale sensor networks budget constraints is another important factor to be considered during the development of a boundary identification scheme. Thus, schemes having assumptions, like each node being equipped with a GPS for position

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 369–379, 2008.
© Springer-Verlag Berlin Heidelberg 2008

estimation [7], or each node being equipped with a directional antenna [4], are not appropriate for the type of sensor field under consideration.

Our major focus in this paper is to reduce the overall deployment cost of the network by using very cheap sensor nodes as well as to increase the lifetime of the sensor network by avoiding message flooding.

It has been recognized that the use of mobile sink in wireless sensor networks is growing at a very fast rate because of its advantages in terms of increased lifetime of the network [5], cost effective sensor field localization [14], etc. In this paper, we show how to use a mobile sink equipped with directional antenna as a tool for the boundary identification of a sensor network. In order to achieve the desired task we transform the problem of boundary identification into one of the edge node identification, where the sink identifies an edge node, moves to it and then determines the next edge node. The process continues until the sink completely identifies the boundary of the sensor field. One very common objection against application of a mobile sink is that for some types of terrain it is difficult for a sink to move around. However, advancements in robotics have resulted in the production of machines which can move in difficult terrains. For example, a DARPA funded research project named *BigDog* develops a machine that is mobile even in harsh terrain, can move at 4 miles per hour and can climb slopes up to 35 degrees [15].

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 outlines the basic setup, Section 4 presents the boundary identification scheme, Section 5 is discussion and analysis, and Section 6 concludes the paper.

2 Related Work

This section presents few state of the art methodologies for the boundary identification of a wireless sensor network.

Wang et al. [3] divides the existing methods in the area of boundary identification of wireless sensor networks into three classes, depending on the techniques used. *Geometric methods* are based on the assumption that every node knows its position coordinates. *Statistical methods* utilize the probability distribution of the deployed sensor nodes and identify the boundary nodes on the basis of average neighbor density. *Topological methods* make use of sensor field flooding for boundary identification. In [3] they have also proposed a flooding-based algorithm that determines the edge nodes in a sensor field. It is based on the observation that holes in the sensor field create irregularities in the hop count distances which helps identifying *cuts* in the sensor field. These cuts are then utilized to determine the boundary of the sensor field.

Kröller et al. [2] presented an algorithm that is based on a distributed flower structure for edge node detection; it also identifies the natural geometric clusters in the sensor field.

Zhang et al. [4] presented a neighbor embracing polygon (NEP) based algorithm where each node only requires the direction information of the neighboring nodes to create a convex hull of its neighbors. If the node which created the convex hull is located outside the convex hull boundary then it is an edge node and vice versa.

Fekete et al. [6] worked on identifying the edge nodes based on the fact that nodes located close to the center of the sensor field have higher centrality than nodes located near the boundary, provided that nodes follow a suitable random distribution.

Prerequisite in Zeinalipour-Yazti et al.'s [7] algorithm is that each node knows its own position coordinates along with the neighboring nodes. Then the node having minimum y coordinates in the sensor field is determined and marked as starting perimeter node which then selects the neighboring perimeter node by measuring the polar angles of all the neighboring nodes on its x-axis. The line obtained by connecting identified edge nodes is the boundary of the field.

Discussion shows that all existing techniques impose one or more of the following conditions on the sensor field: sufficient node density, special hardware requirements, or intensive communication requirements. The boundary detection scheme proposed in this paper operates with low-cost sensor nodes and significantly reduces the communication requirements amongst the nodes.

3 Preliminaries

This section summarizes the basic assumptions underlying the paper.

It is assumed that we have to monitor a highly polluted site, for example, containing toxic or radioactive materials. However the terrain of the area is assumed to be suitable for sink mobility. Deployment of the nodes is performed by dropping them from an airplane or cannon fire which leads to uniform random distribution of the nodes as shown in Figure 1. Deployed nodes are static and inexpensive having omnidirectional antennas with the same fixed transmission range which is very small compared to the size of the sensor field. Each node is equipped with limited power supply that cannot be recharged or replaced, and thus nodes are programmed to operate at 1% duty cycle. Moreover, it is assumed that the nodes have no knowledge of their position coordinates in the field.

Each node in the sensor field acquires a "valid" or "invalid" status. The validity of a sensor node is determined by the number of its neighboring nodes. A node x is called *neighbor* of node y if x lies within the transmission range of y. It is assumed that the sensor field contains only one cluster of valid sensor nodes and other nodes located outside this cluster are invalid nodes (see Figure 1).

The sink is a special node which is mobile and equipped with an unlimited energy resource, a GPS and a compass that are used to determine its position and direction of mobility. Moreover, the sink is also equipped with a sectored directional antenna having fixed transmission range equal to that of the sensor node. It can be used to determine the angle of arrival (AOA) of a message from a sensor node [8] and to roughly estimate the distance between a node and the sink using RSSI [9] based distance measurement.

The sink knows the *area of interest* (AOI). The AOI is a rectangular region which contains all the deployed sensor nodes. Assumption regarding AOI does not affect the generality of our algorithm as it is only used to locate the sensor field by the sink (discussed in Section 4.1). In the following, some terminology frequently used in this paper is defined.

The *boundary of a sensor field* is a subset of valid sensor nodes with the property that the line obtain by connecting each node in this subset with its neighboring edge node "encloses" all the other valid sensor nodes as shown in Figure 1.

Edge Nodes are valid sensor nodes that are connected to obtain a *boundary line* which enclose all the other valid sensor nodes as shown in Figure 1.

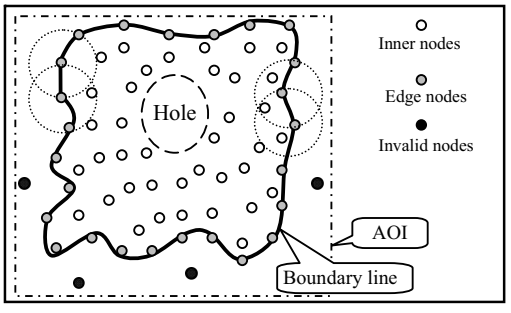


Fig. 1. Network model

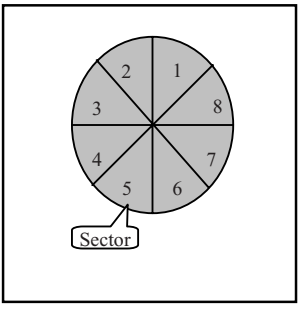


Fig. 2. Directional antenna

A *directional antenna model* presented in [13] is considered in this paper. The antenna system is composed of N beams such that their intersection is zero and their union covers the entire 360 degree plane as shown in Figure 2. The width of each beam is equal to 360/N and the area covered by one beam is called a sector. We consider a large value for N therefore the size of a sector is very small.

Edge node position estimation refers to the estimation of the position coordinates of an edge node by the sink. For this purpose the sink uses its directional antenna to measure the angle of arrival (AOA) [10] and the received signal strength (RSSI) [11] from a sensor node. Then, based on its own position (calculated using GPS), the sink estimates the position coordinates of the sensor node (discussed in Section 4.12 and 4.2).

4 MObile Sink Based BOundary Detection (MoSBoD)

This section presents the new algorithm *MoSBoD* for boundary detection using a mobile sink. The two main phases are (*i*) bootstrapping (for sensor node validation and identification of a starting edge node) and (*ii*) edge node identification and boundary traversal using the mobile sink.

4.1 Bootstrapping Phase

The bootstrapping phase is an initialization phase of the MoSBoD algorithm. During this phase sensor nodes prepare themselves for the arrival of the sink by calculating their validity status. Simultaneously, the sink locates a valid edge node in the sensor field and marks this node as the starting edge node.

4.1.1 Bootstrapping of the Sensor Nodes

In the bootstrapping phase sensor nodes are divided into two groups of valid and invalid nodes. As input for this phase, each node is preprogrammed with a time value *t1* and neighbor density *(validity_cnt)* required for calculating the validity status of a node. After the deployment of a sensor field each node performs the following operations: activate message reception mode and broadcast a message containing the own ID (on expiration of time *t1*). On receipt of messages from neighboring nodes create a list of neighbors containing their ID's and set their validity status to *false*. Then,

(i) If the neighbor count becomes equal to *validity_cnt*, then set own validity equals *true* and broadcast a message containing the ID and the validity status.
(ii) On receipt of a validity message update the sender nodes' validity status.

4.1.2 Identification of Starting Edge Node

During this phase the sink calculates the mobility direction to reach the boundary of the AOI and afterward locates the starting edge node. In order to achieve this task the sink carries out the following operations: Determine its current location and alignment with respect to the boundary of the AOI using GPS and compass; calculate mobility direction and move to reach the closest boundary point at the AOI.

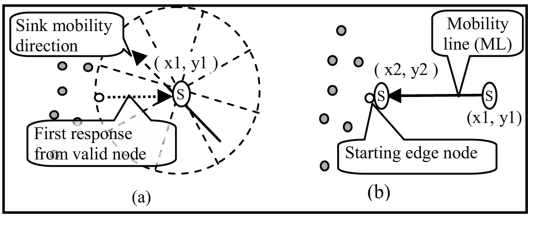


Fig. 3. Identification of the starting edge node

Upon reaching the boundary of the AOI, the sink calculates the center of the AOI (utilizing AOI coordinates), switch on its antenna, starts transmitting a *hello* message and begins to move towards the center of the AOI. The sink continues until a response *(ID, validity)* from a valid sensor node is received. On receipt of a response message, the sink marks the responding node as a starting edge node and saves own current coordinates as *(x1, y1)* (see Figure 3(a)). Also, by utilizing the AOA of the received response and the *edge node position estimation* procedure explained in Section 3, the sink calculates and moves to the position of the starting edge node *(x2, y2)*. Exceptional situations, such as when multiple valid nodes respond to the sink, are also handled in the pseudo code of Module-1. Moreover, it should be noted that unlike [7] where the starting node is the node with minimum *y* coordinates (calculated using GPS) and identified by sensor field flooding, Module-1 does not impose any such requirements.

Module-1 engender following outputs: coordinates of the location when the sink receives first response from a valid sensor node *(x1, y1)*; coordinates of the starting edge node *(x2, y2)*, and starting edge node ID.

Module-1: Locating the AOI and the starting edge node

INPUT: AOI coordinates, AOI = false, s_node = null

1: Calculate and move to the nearest boundary of AOI using AOI coordinates and own position (calculated using GPS)
// Sink moves inside the AOI in search of valid sensor nodes
2: **while** *s_node == null*
3: Move towards center of AOI, broadcasting *hello* message
4: **if** *single node responds* **AND** *nodeValidity == true* **then**
5: *s_node*=respondingNodeID & *(x1,y1)*=current position
6: **else if** *multiple nodes responds* **then**
7: **if** *responding nodes are at same shortest distance from the sink* **then**
8: starting node = node with minimum ID
9: **else** *s_node* = node at shortest distance from the sink
// (Calculated using RSSI based distance estimation)
10: **end if**
11: **end if**
12: **if** (*s_node !== null*)
13: Utilize received response from *s_node* to perform edge node position estimation, calculate *(x2, y2)* and move to the calculated location of the *s_node*
14: **end if**
15: **end while**

4.2 Edge Node Identification and Boundary Traversal

This section presents Module-2 that enables the sink to identify the neighboring edge nodes of a current node (the node where the sink is currently positioned). The line then obtained by connecting all the identified edge nodes with their corresponding neighboring nodes is the desired boundary of the sensor field.

Module-2 is based on the use of mobility and a directional antenna by the sink. Prerequisites for the execution of this algorithm are, the sink is positioned at an edge node *i*, it knows the position coordinates of the current node *i* and identified neighbor edge node *i-1* of the current node.

The sink initiates execution of Module-2 by calculating the reference line which is defined as the line obtained by joining the position coordinates of the current node *i* and it's identified neighboring edge node *i-1*. Then the sink numbers the sectors starting from the one located beside the reference line towards the mobility direction of the sink. We specify that the sink traverses the boundary of the sensor field in counterclockwise direction. In this case, the sink will mark the sector located in counterclockwise direction of the reference line as sector 1 as shown in Figure 4(b).

Once the sectors are numbered, the sink broadcasts a *hello* message to the neighboring nodes of the current node. The node whose response is received in the lowest sector number is assigned the status of next edge node *i+1* by sending an *edge node confirmation message*. For example, in Figure 4(b) response from node *i+1* is received in Sector 4 while the responses from all the other nodes are received in sectors having ID greater than 4. Therefore, node *i+1* is assigned the status of next edge node. Moreover, position coordinates of node *i+1* are estimated using *edge node position estimation* and the sink moves to its position.

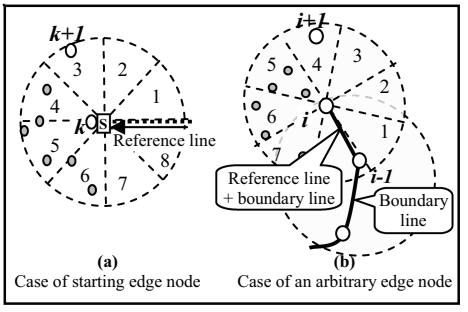


Fig. 4. Neighboring edge node identification

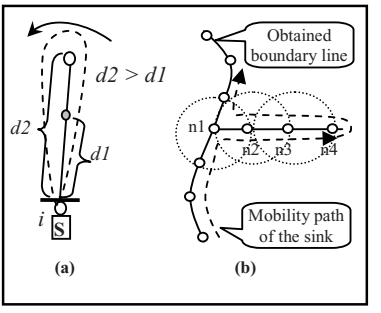


Fig. 5. Special cases in Module-2

On reaching node *i+1,* the sink again executes Module-2 to identify the neighboring edge node of node *i+1*. Thus, by moving from one edge node to the next, the sink eventually returns to the starting edge node after completing the boundary trace.

The discussion so far leaves one open question: How the sink determines the reference line when it is positioned at the starting edge node? It is known that the starting edge node is the first node to be identified as an edge node and at this point of time the sink has no information about the neighboring edge nodes of the starting node. Thus, the starting edge node is a special case for the reference line identification process. In this case we utilized the coordinates of the current node (*x2, y2)* and the coordinates *(x1, y1)* (obtained from Module-1) to define the reference line. Since it is assumed that the sink traverses the boundary of the field in counterclockwise direction, the sink assigns numbers to the sectors in ascending order starting from the one located towards counterclockwise direction of the reference line, as shown in Figure 4(a). The rest of the procedure for the identification of the next edge node is the same as already discussed.

During the neighboring edge node identification some exceptions can arise which are handled in the pseudo code of Module-2. For example, if the sector with the lowest number (Sector 4 in Figure 4(b)) receives responses from two or more nodes then it is assumed that the two nodes are located on a line, because it is assumed that the size of a sector is very small. In this case, the node located at farthest position from the current node is selected as next edge node as shown in Figure 5(a). There may also be the case where a group of nodes are connected to the main sensor field via a single link, like node n_1 which connects n_2, n_3 and n_4 with the rest of the field as shown in Figure 5(b). Since we do not impose any restriction on the number of times the sink can visit an edge node during the boundary identification process, such cases can also be handled successfully by our algorithm.

Module-2: Edge node identification and boundary traversal

Input: The sink is positioned at the starting edge node, *i*. Coordinates of the edge nodes *i* and *i-1* are (x2, y2) and (x1, y1) respectively.

1: *current node = i;*
2: **do**
3: Number sectors according to the reference line
// Identification of the next edge node

```
4:    Transmit a hello message to the neighbors of the current node;
5:    if only one node nb responds then    i+1 = nb;
6:    else if  multiple responding nodes are located at same farthest distance from sink then
7:          i+1 = node having minimum ID
8:    else  i+1 = the farthest node;  // determined using RSSI
9:    end if
10:   Apply edge node position estimation for node i+1;
11:   Store position coordinates of node i+1 and move to it
          // edge node confirmation message
12:   Send a message to node i+1 demanding its list of neighbors;
13:   Set  i-1 = i;        i = i+1;
14: until i != current node
```

Example. Figure 6 shows the implementation of the MoSBoD algorithm on a sample sensor field. Once the boundary is fully identified, the sink continues its mobility along the boundary line monitoring possible edge node failures for boundary reconstructing. Since the later trips along the boundary line are based on the stored position coordinates of the edge nodes, they will require less time than the first trip.

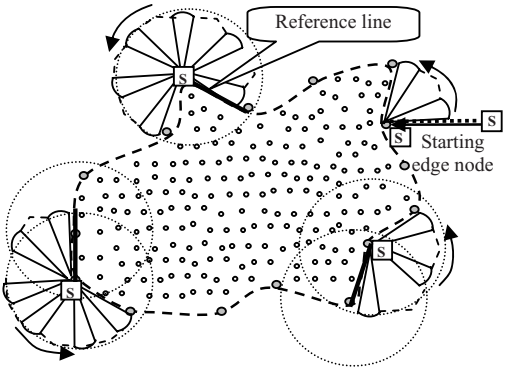


Fig. 6. Edge node identification

5 Discussion and Analysis

This section presents an evaluation of the MoSBoD algorithm based on the OMNeT++ simulation tool. We analyze the effects of neighbor node density and of the size of the sensor field on the energy consumption of the MoSBoD algorithm. Also, a theoretical analysis of the completion time of the MoSBoD algorithm is given. For completeness we have compared the results obtained with a boundary identification scheme presented in [3]. We selected this particular scheme for comparison because it has similar assumptions regarding sensor node hardware, deployment strategy etc. which we are using in this paper. The graphs show 95% confidence intervals for the quantities of interest.

The basic simulation setup comprises an area of $800\text{x}500\,m^2$ where the sensor nodes are uniformly, but randomly, deployed. The communication range for the nodes

is set to 80 m. We applied the MoSBoD algorithm to sensor fields with random, U shaped, circular and rectangular boundary shapes with varying neighbor node densities of 4, 6, 14 and 18. The boundary obtained with our algorithm remains the same irrespective of the changes in neighbor node density. It reflects the fact that, in contrast to [2], [3] or [12] which require a neighbor density of at least 7 to produce acceptable boundaries [3], our MoSBoD algorithm is based on sink mobility which and does not impose any such conditions.

5.1 Energy Consumption

With respect to energy consumption, we investigated two hypotheses:

Hypothesis 1: In the MoSBoD algorithm, the number of messages sent out per sensor node is constant and the number of messages received depends linearly on its neighbor node density.

Hypothesis 2: Scaling up/down the area of the sensor field with constant node density has no effect on the number of messages exchanged by the sensor nodes in the MoSBoD algorithm.

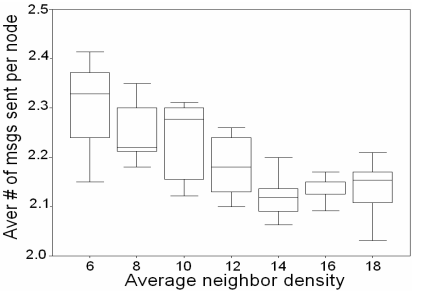


Fig. 7. Messages sent per node vs average neighbor density

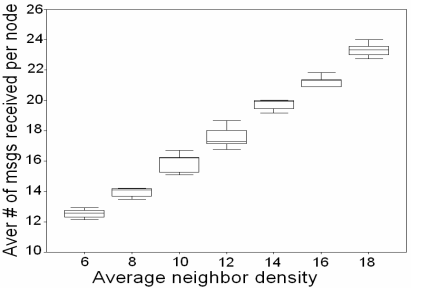


Fig. 8. Messages received per node vs average neighbor density

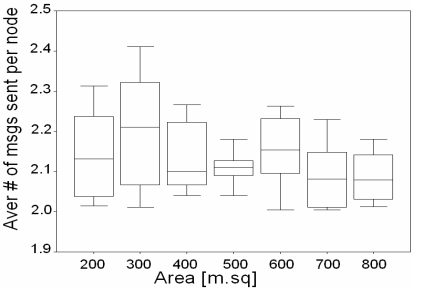


Fig. 9. Messages sent per node vs area of sensor field

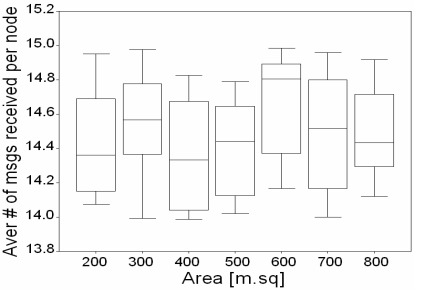


Fig. 10. Messages received per node vs area of sensor field

For a simulation based validation of *Hypothesis 1* we set up a simulation environment where the deployment area for the sensor field was fixed. Then nodes were deployed with different densities to this area and the MoSBoD algorithm was

executed for boundary identification. Figure 7 shows that the total number of messages sent out by a node during the boundary identification procedure is basically constant (slightly larger than 2) irrespective of the neighbor density, while Figure 8 shows that the number of messages received by a node is a linear function of its neighbor density. In contrast, the scheme discussed in [3] requires each node to broadcast at least *three* messages. This implies that the MoSBoD algorithm consumes at least 33% less energy from the nodes as compared to [3], both in terms of message transmission and reception which increases the lifetime of the sensor field.

On investigating *Hypothesis 2*, we observe that for average neighbor node density equal 7 the number of messages exchanged by the nodes is practically not affected by a change in the area of the sensor field, as shown in Figures 9 and 10. This is due to the fact that in the MoSBoD algorithm node to node communication takes place only to determine the validity status of a node, which is a localized phenomenon and does not depend on the size of the field. This shows the highly scalable nature of our algorithm where the size of the field does not affect the behavior of an individual node (for constant neighbor density).

5.2 Completion Time

As explained in Section 4, the MoSBoD algorithm exploits sink mobility to identify edge nodes and then connects them to obtain the boundary of a sensor field. Since the mobility speed of the sink is very slow compared to the speed with which messages can be exchanged between the nodes, the first impression may be that the completion time of the MoSBoD algorithm is extremely high as compared message flooding based schemes, for example, as discussed in [3]. It is the topic of ongoing work to analyze this aspect quantitatively. A potential strategy for reducing the completion time of the MoSBoD algorithm is a selective increase of the duty cycles of sensor nodes. It is required, though, to carefully balance the expected reduction in latency with the resulting increase in energy consumption. A detailed investigation and analysis of this strategy is provided in a forthcoming paper.

Area size of the sensor field. Since the MoSBoD algorithm is based on sink mobility for the identification of the edge nodes, an increase in the area of the sensor field leads to a linear increase in the completion time of the algorithm.

6 Conclusion and Future Work

In this paper we introduced a new scheme for boundary identification of a sensor field. Our approach utilizes a mobile sink for edge node identification which reduces the communication requirements amongst the nodes. The proposed MoSBoD algorithm has a definite edge over currently available algorithms in terms of energy consumption and in terms of neighbor node density requirements for correct boundary identification. Moreover, it does not impose any restrictions on the deployment of the sensor nodes.

We are currently extending this work along two fronts. First, we pursue ideas for improving the quality of the identified boundary as well as for reducing the completion time of the algorithm. Second, we are carrying out a detailed analysis of energy

consumption and completion time of the improved MoSBoD algorithm and quantitatively compare it with the state of the art methodologies.

References

1. Nowak, R., Mitra, U.: Boundary Estimation in Sensor Networks: Theory and Methods. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634. Springer, Heidelberg (2003)
2. Kröller, A., Fekete, S.P., Pfisterer, D., Fischer,: Deterministic boundary recognition and topology extraction for large sensor networks. In: SODA 2006, pp. 1000–1009. ACM Press, New York (2006)
3. Wang, Y., Gao, J., Mitchell, J.S.: Boundary recognition in sensor networks by topological methods. In: MobiCom, pp. 122–133. ACM Press, New York (2006)
4. Zhang, C., Zhang, Y., Fang, Y.: Localized coverage boundary detection for wireless sensor networks. In: Proceedings of the 3rd international Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Waterloo, Ontario, Canada, August 07 - 09, 2006. QShine 2006, vol. 191, p. 12. ACM Press, New York (2006)
5. Luo, J., Hubaux, J.P.: Joint mobility and routing for lifetime elongation in wireless sensor networks. IEEE INFOCOM (March 2005)
6. Fekete, S.P., Kaufmann, M., Kröller, A., Lehmann, K.: A new approach for boundary recognition in geometric sensor networks. In: 17th Canadian Conference on Computational Geometry (CCCG 2005), Windsor (2005)
7. Zeinalipour-Yazti, D., Andreou, P., Chrysanthis, P.K., Samaras, G.: SenseSwarm: a perimeter-based data acquisition framework for mobile sensor networks. In: Proceedings of the 4th Workshop on Data Management For Sensor Networks: in Conjunction with 33rd VLDB (September 24-24, 2007). DMSN 2007, pp. 13–18. ACM, New York (2007)
8. Nasipuri, A., Li, K.: A directionality based location discovery scheme for wireless sensor networks. In: WSNA, pp. 105–111. ACM Press, New York (2002)
9. Kuo, S., Tseng, Y., Wu, F., Lin, C.: A Probabilistic Signal-Strength-Based Evaluation Methodology for Sensor Network Deployment. In: Proceedings of the 19th international Conference on Advanced information Networking and Applications, March 25 - 30, 2005. AINA, vol. 1, pp. 319–324. IEEE Computer Society, Washington (2005)
10. Elnahrawy, E., Austin-Francisco, J., Martin, R.P.: Adding Angle of Arrival Modality to Basic RSS Location Management Techniques. In: Proceedings of IEEE International Symposium on Wireless Pervasive Computing (ISWPC 2007) (February 2007)
11. Ash, J., Potter, L.: Sensor Network Localization via Received Signal Strength Measurements with Directional Antennas. In: Proceedings of the Forty-Second Annual Allerton Conference on Communication, Control, and Computing, Champaign-Urbana, IL, September 2004, pp. 1861–1870 (2004)
12. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.A.: Models and solutions for radio irregularity in wireless sensor networks. ACM Trans. Sen. Netw. 2(2), 221–262 (2006)
13. Choudhury, R., Vaidya, N.: Capture-Aware Protocols for Wireless Multihop Networks Using Multi-Beam Directional Antennas. Technical report, UIUC (March 2005)
14. Ssu, K.-F., Ou, C.-H., Jiau, H.C.: Localization with mobile anchor points in wireless sensor networks. IEEE Transactions on Vehicular Technology 54(3), 1187–1197 (2005)
15. Raibert, M., Buehler, M.G., Playter, R.R.: Boston Dynamics. In: "BigDog", SPIE Defense & Security Symposium, Orlando, Florida, USA, April 9-13 (2007)

Analysis of IEEE 802.11e Line Topology Scenarios in the Presence of Hidden Nodes*

Katarzyna Kosek, Marek Natkaniec, and Andrzej R. Pach

AGH University of Science and Technology, Krakow, Poland
{kosek,natkanie,pach}@kt.agh.edu.pl
http://www.kt.agh.edu.pl/

Abstract. In this paper an innovative simulation study of five IEEE 802.11e network configurations is presented. The conducted analysis is crucial for understanding how a theoretically simple and, most of all, popular line topology network can be degraded by the presence of hidden and exposed nodes. The discussion of the obtained results helps to understand how and why the behavior of IEEE 802.11e based line topologies changes when the number of nodes increases. Furthermore, the usefulness of the four-way handshake mechanism is argued. Finally, the need for a better MAC protocol is stressed and a number of novel conclusions about the IEEE 802.11e nature is provided.

Keywords: ad-hoc, hidden and exposed nodes, IEEE 802.11e.

1 Introduction

Wireless networking technology is quickly evolving and its importance grows constantly. The most interesting technology, not only from the perspective of a researcher but also from the perspective of an average user, seem to be ad-hoc networking. These networks without infrastructure do not need complicated admin-istration and may greatly facilitate Internet access. Unluckily, all wireless networks were created to deal with data exchanges and not multimedia services. Therefore, the need for QoS assurance for delay sensitive and/or bandwidth consuming services remains an interesting and unresolved issue. Constantly changing and unpredictable channel conditions, hidden and exposed node problems, varying network load, changeable device performance, different transmission and sensing ranges, and mobility of ad-hoc networks make it an even more difficult task. In this article the authors focus on the hidden and exposed node problems which they find the most interesting.

Five different configurations of ad-hoc line topologies are simulated. The purpose of analyzing line topologies is simple. A good example of such a case in a real environment is a simple mesh network in which ad-hoc nodes communicate with a gateway (GW) every time they access the Internet services. At the same time, most of these nodes are out of range of GW and need to send their data through other nodes.

* Disclaimer This work has been realized under the Polish Ministry of Science and Higher Education project no. N51739133.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 380–390, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Another example are long distance multi-hop links using the same radio channel which could be used in rural areas where access to infrastructure is highly limited.

Due to the fact that all kinds of topologies require QoS, the authors found it crucial to check if IEEE 802.11e [1] can assure QoS in such environments. This paper presents novel results regarding line topologies. To the authors' best knowledge similar analysis has not been performed. Related work can be found in [4] in which, however, the authors did not take into account different line topologies and did not analyze how the length of a line impacts the network performance. Additionally, they did not notice the undesirable inversion in prioritizing traffic and, furthermore, the values of EDCA access parameters used were not compatible with the IEEE 802.11e standard.

The analysis presented in this article helps to draw innovative conclusions about IEEE 802.11e behavior. Among many consequences of the hidden and exposed nodes presence, the most important seem the unavoidable unfairness in granting medium access and distortion of the throughput levels of different priority streams. The paper also argues the usefulness of the four-way handshake method in minimizing their degrading impact on IEEE 802.11e performance. Additionally, the gathered results are compared with the results obtained for two different star topology networks presented in [5].

The remainder of this paper is organized as follows. Section 2 describes the simulation scenarios. Section 3 gives explanation of the obtained results and presents scrupulous conclusions. More general conclusions can be found in Section 4.

2 Simulated Scenarios

The simulation analysis was performed with the use of an improved version of the TKN EDCA enhancement [3] to the ns2 simulator. The adjustments made mostly affect the RTS/CTS mechanism which was not supported properly by the original version of the TKN EDCA patch. Additionally, the handling of duplicate drops was fixed. Important simulation parameters are given in Table 1 and Table 2.

Table 1. EDCA parameter set

Priority	AC	$CW_{min}[AC]$	$CW_{max}[AC]$	*AIFSN[AC]*	*TXOP*
P0	Vo	7	15	2	0
P1	Vi	15	31	2	0
P2	BE	31	1023	3	0
P3	BK	31	1023	7	0

Table 2. General simulation parameters [2]

SIFS	10 µs	***DIFS***	50 µs
PIFS	30 µs	***Slot Time***	20 µs
Tx Range	250 m	***Tx Power***	0.282 W
Frame Size	1000 B	***Traffic Type***	CBR/UDP
CS Range	263 m	***Node Distance***	200 m

The simulation study was performed with the assumptions that all nodes send CBR traffic[1] with a varying sending rate (from 10 kb/s to 10 Mb/s) and the IEEE 802.11b standard [2] is used as the physical layer type. The nodes form line topologies in which each node can only detect transmissions of its nearest neighbors (c.f., Fig. 11). The number of nodes changes from 3 (numbered from left to right N0-N2) to 7 (N0-N6)[2]. Additionally, for every analyzed network setup, four different EDCA configurations are simulated. In each configuration a different EDCA class is used for the flows generated by the network-forming nodes. The propagation model used is the two-ray ground reflection model.

In order to combat the hidden node problem the RTS/CTS mechanism is used. Additionally, for the sake of clarity of the presented figures, if two nodes obtain similar throughput it is presented as a single mean value (e.g., N0/N2 for nodes N0 and N2 in Fig. 2). For the same reason, in Fig. 3-Fig. 5 only the curves representing Vo and BE priority are presented because their performance is very similar to that of Vi and BK, respectively (c.f., Fig. 3). Moreover, in all presented figures the error of each simulation point for a 95 % confidence intervals does not exceed ± 2 %.

3 Simulation Results

In this section the results obtained for the three- to seven-node line scenarios will be described. Firstly, the overall performance of particular networks will be analyzed by comparing the obtained throughput by nodes for four different priorities. Secondly, the six-node line will be described in detail by means of frame dropping probability, retransmission drops and duplicate drops. Finally, a comparison with two star topology networks [5] will also be given.

3.1 Three-Node Line

With the RTS/CTS exchange disabled, hidden nodes with Vo and Vi priorities obtain smaller throughput than BE and BK in general (Fig. 1a). Furthermore, when the overall traffic load exceeds 225 KB/s, the throughput of Vi and Vo streams drops to zero. For the unhidden node, the order of the throughput levels is in line with the IEEE 802.11e guidelines.

With RTS/CTS enabled, the throughput of hidden nodes slightly increases and drops for the unhidden node (Fig. 1b). However, the strong unfairness between the hidden and unhidden nodes is not eliminated. Additionally, when hidden nodes are transmitting Vo traffic the unfairness is strongest as they obtain the lowest throughput which is practically equal to zero for traffic load exceeding 500 KB/s.

The above observations lead to a conclusion that with RTS/CTS both enabled and disabled, the three-node line topology network will not work properly. In such a network, for hidden nodes, low priority traffic will be always prioritized over high priority traffic and, additionally, the unhidden node will be strongly prioritized over the hidden ones.

[1] The authors performed the analysis also with different traffic types however the achieved results were very similar to those obtained for CBR traffic.

[2] The maximal number of nodes was set to 7 because in the real world it is very hard to find longer line topologies (e.g., long distance links).

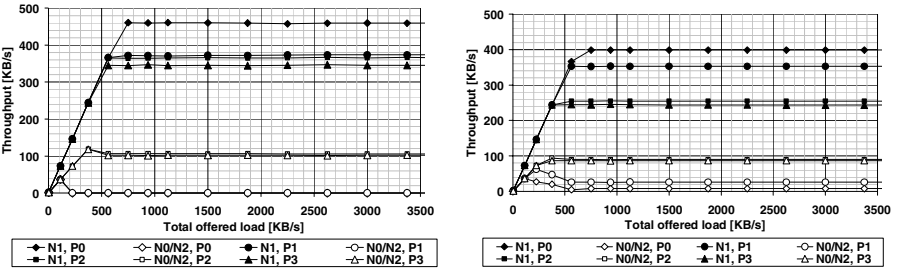


Fig. 1. Three-node line. Throughput for RTS/CTS (a) disabled (b) enabled.

3.2 Four-Node Line

For a four-node line topology the throughput curves change (Fig. 2a) in comparison to the three-node line. With the RTS/CTS exchange disabled the throughput order may be divided into two main sets. Under lighter load (below 2 MB/s), N0/N3 transmitting BE and BK obtain higher throughput than N2/N4 transmitting Vi and Vo. Under heavier load, this order changes so the unfairness between these pairs of nodes is even stronger. Furthermore, under network load exceeding 150 KB/s, for all nodes, BE and BK priority streams are favored over Vi and Vo.

With RTS/CTS enabled, N1/N2 are prioritized over N0/N3 (Fig. 2b). Moreover, for all nodes low priority streams obtain higher throughput than high priority streams. In comparison to the three-node topology, the overall throughput drops and, therefore, the network performance of four-node line topology is worse for both enabled and disabled RTS/CTS.

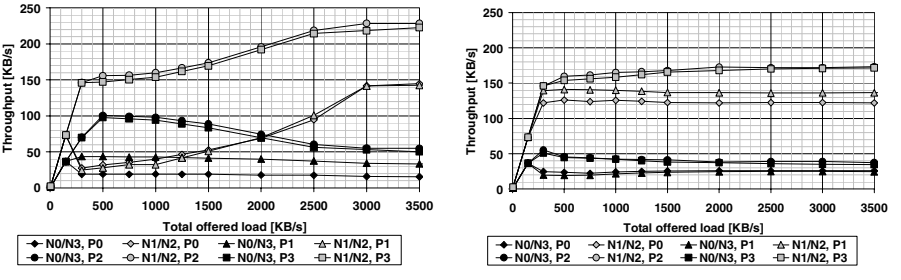


Fig. 2. Four-node line. Throughput for RTS/CTS (a) disabled (b) enabled.

3.3 Five-Node Line

For the five-node line, with the RTS/CTS exchange disabled, for Vo priority traffic nodes N1/N3 obtain the highest throughput, N0/N4 smaller and N2 the smallest (which is totally unacceptable for network load over 375 KB/s). Moreover, the observed unfairness between certain nodes increases as the total offered load grows. For BE priority the order of N1/N3 and N0/N4 is reversed (Fig. 3a).

With the RTS/CTS exchange enabled, the throughput level order may be divided into two sets (Fig. 3b). The throughput levels under non-saturation conditions for N1/N3 and N0/N4 for Vo are the lowest but they grow with the increase of the offered load. Similarly, also the throughput of N0/N1/N3/N4 sending BE grows linearly. At the same time, a decrease in the throughput value of N2 can be observed for both BE and Vo. Finally, under network load of over 2.5 MB/s the throughput values are stable and N2 obtains smallest throughput regardless of the traffic priority it transmits. In all analyzed cases, BE is prioritized over Vo but the strongest unfairness is present for nodes N0 and N4.

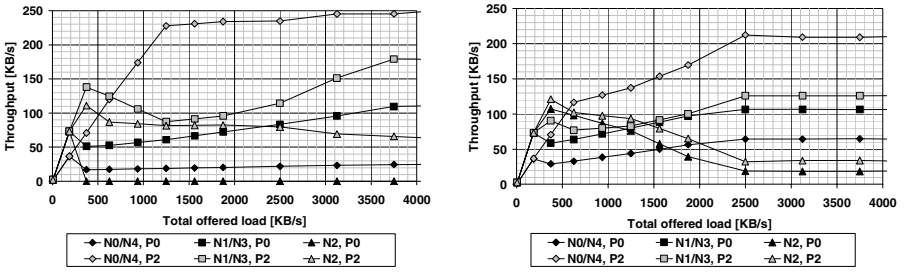


Fig. 3. Five-node line. Throughput for RTS/CTS (a) disabled (b) enabled.

3.4 Six-Node Line

In case of the six-node line topology, the obtained results resemble the results for the four-node line in the case of Vo/Vi priority transmission. Which means that the nodes being in the middle of the line have the smallest throughput, the ones next to them win the competition for medium access most often and, finally, all other nodes receive average priority in medium access. However, in this configuration the unfairness between high priority traffic and low priority traffic is stronger because practically under every network load for RTS/CTS both enabled and disabled, all nodes sending Vo obtain smaller throughput than the corresponding ones sending BE (Fig. 4). The strongest unfairness is observed for the side nodes — N0/N5 (similarly to the four-node line's N0/N4) but it also increases for the nodes N1/N3.

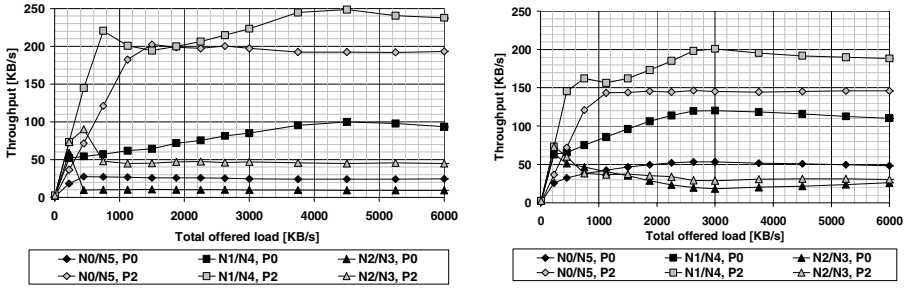


Fig. 4. Six-node line. Throughput for RTS/CTS (a) disabled (b) enabled.

3.5 Seven-Node Line

The observations for the seven-node line are similar to the ones made for the six-node line. Once again, regardless of the RTS/CTS exchange, nodes sending high priority traffic streams obtain smaller throughput than the corresponding ones sending low priority streams. Additionally, N1/N5 win the competition for medium access most often, and the one in the middle (N3) less often. The main difference is that the strongest unfairness can be observed for nodes N1/N5 and not the side-nodes.

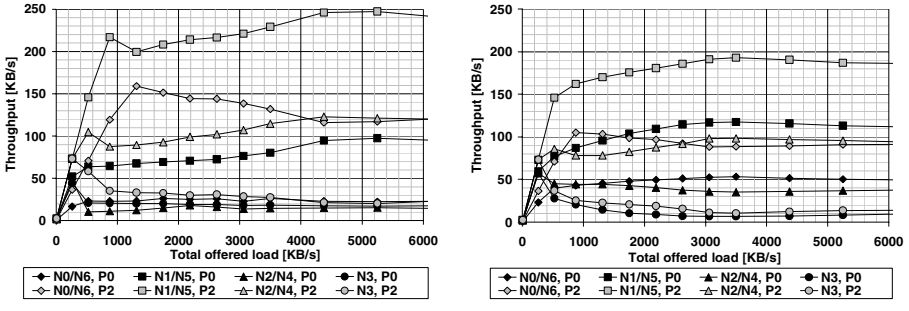


Fig. 5. Seven-node line. Throughput for RTS/CTS (a) disabled (b) enabled.

3.6 Overall throughput

The overall saturation throughput levels obtained for the analyzed scenarios are presented in Fig. 6. As can be seen, for the high priority traffic the saturation throughput is highest for the shortest line. For the low priority traffic the situation changes because the saturation throughput grows meaningfully as the number of nodes increases. Additionally, in all cases the throughput of high priority traffic is smaller than for low priority traffic which differs from IEEE 802.11e assumptions.

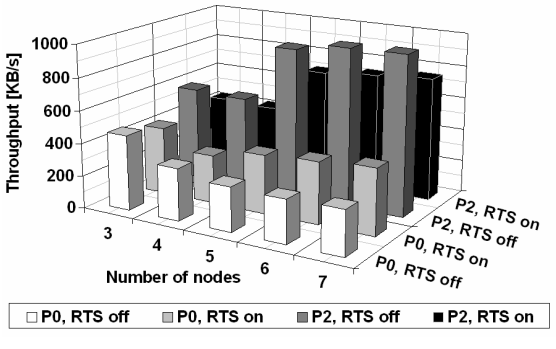


Fig. 6. Overall saturation throughput

3.7 Detailed Conclusions

The performance shown in Fig. 1-Fig. 5 can be explained by the rate of the total frame loss for each of the analyzed flows in every simulation scenario. Additionally it can be also justified by the general character of each of the analyzed networks. For example in the case of the three-node line topology, two hidden and zero exposed nodes appear. In the five-node line there are five hidden nodes and three exposed ones, however, the hiddenness and exposedness of particular nodes differs.

Due to the lack of space only the performance of the six-node network will be explained in great detail with the help of the results presented in Fig. 7-Fig. 10. This network has been chosen as the most general one. The conclusions regarding this network will also be valid for the remaining ones.

The frame dropping probability is computed on the basis of the number of dropped frames in interface queues between the LLC and the MAC layers. In particular, it is the number of dropped frames to the number of generated frames.

$$P_{drop} = \frac{n_{dropped}}{n_{generated}}. \tag{1}$$

For RTS/CTS disabled (Fig. 7a), N2/N3 almost always have the same frame dropping probability. However, there is a great difference for N1/N4 and N0/N5 between Vo and BE. This behavior means that for BE the dropping probability is, in general, smaller than for Vo which differs from IEEE 802.11e assumptions. The dropping probability for N0/N5 is non zero for BE for the total offered load exceeding 1 MB/s, for Vo – exceeding 0.45 MB/s. In both cases, for all remaining nodes it starts earlier. Such a behavior could lead to an assumption that N0 and N5 should achieve highest throughput for both Vo and BE. When we look, however, at Fig. 4a and Fig. 4b, we see that these are N1 and N4 which outperform all other nodes. This is a result of the fact of the exposedness of N1 and N4 which leads to high number of duplicate drops (c.f., Fig. 10) described later in this section. In general, for BE non zero frame dropping probability starts later than for Vo which contradicts IEEE 802.11e. It is a result of less frequent attempts in obtaining medium access by BE. Higher priority means smaller values of EDCA access parameters and a higher possibility for competing for medium access. Such a behavior causes a higher probability of collisions for hidden nodes and, consequently, it leads to a higher number of retransmissions which cause quicker filling of the four MAC priority queues. As a result, the quicker a certain queue is filled the higher the probability that it will be overloaded and more interface queue drops will be observed. Obviously, the more duplicate frames are sent the higher probability that they will collide in the wireless medium instead of the good frames. Therefore, duplicate drops and retransmission drops should be analyzed together in order to understand this complicated behavior. One other thing which matters in analyzing the frame dropping probability curves is the type of their slopes. The steepness of slopes show how high is the speed of filling the MAC priority queues. Curve slopes are gentlest for N0/N5 and steepest for N2/N3. This is a result of the strength of the exposedness and hiddenness of particular nodes. N0/N5 are only hidden and N2/N3 are the most exposed and the most hidden nodes.

With RTS/CTS enabled (Fig. 7b), N2/N3 have a slightly different frame dropping probability for Vo and BE. Also a smaller difference for N1/N4 and N0/N5 between Vo and BE can be noticed. The dropping probability for N0/N5 for BE is non zero for the total offered load exceeding 0.75 MB/s, and for Vo – 0.22 MB/s. However, in general, frame dropping probability increases for BE and decreases for Vo (it increases slightly only under light network load for N0/N5 and N1/N4) in comparison with RTS/CTS disabled. The performance of Vo flows can be explained by the small values of EDCA access parameters which lead to more frequent medium access attempts. Obviously, this time DATA transmissions can be successful more often than with RTS/CTS disabled due to the small lengths of the RTS and CTS signaling frames in comparison to DATA frames. The performance of BE can be explained by the increased signaling overhead which causes that DATA frames to wait in the MAC queues for the successful RTS/CTS exchange. The increased overhead for Vo is not as meaningful because of the incomparable gain from successful transmissions of DATA frames. Due to the fact that the frame dropping probability curve's slopes are very similar to the previous ones, the explanation is the same and the strength of the exposedness is the main reason to blame.

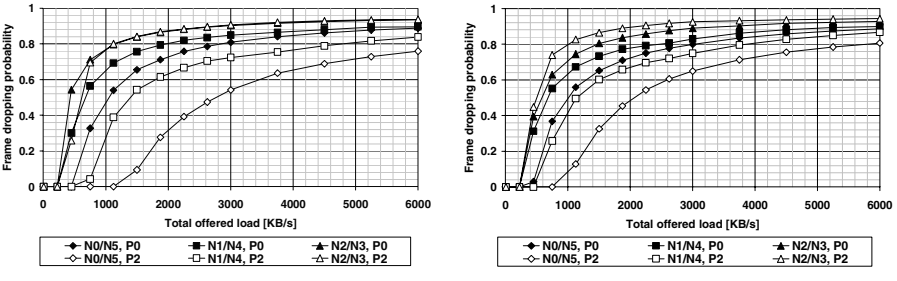


Fig. 7. Six-node line. Frame dropping probability for RTS/CTS (a) disabled (b) enabled.

For the sake of further conclusions, it is important to stress that the strongest hiddenness and exposedness can be observed for N2/N3, weaker for N1/N4, and weakest for N0/N5. However, nodes N0 and N5 hear only N1 and N4, respectively, and nodes N1/N4 hear twice as many nodes each.

The number of retransmission drops (c.f., Fig. 9a-b) is a result of the transgression of the long retry limit (equal to 7, for RTS/CTS enabled) or short retry limit (equal to 4, for RTS/CTS disabled). When the number of retransmissions is compared with the number of collisions (c.f., Fig. 8a-b) for particular nodes it is easily noticeable that with the RTS/CTS exchange disabled the number of retransmissions is in line with the number of collisions for all of the nodes. With RTS/CTS enabled the situation changes drastically. The order of curves representing collisions is completely reverse to those representing retransmissions. This is caused by the fact that in this situation the RTS frames collide instead of the DATA frames. It is also evident that, in comparison to all other nodes, the number of retransmissions decreased most meaningfully for N0/N5 and less meaningfully for N2/N3. Such a behavior leads to a

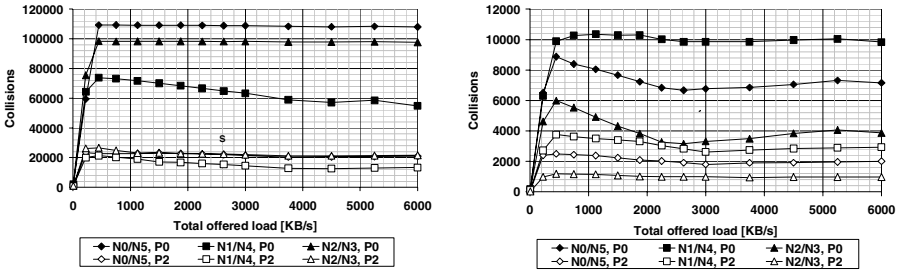


Fig. 8. Six-node line. DATA collision drops for RTS/CTS (a) disabled RTS/CTS (b) enabled.

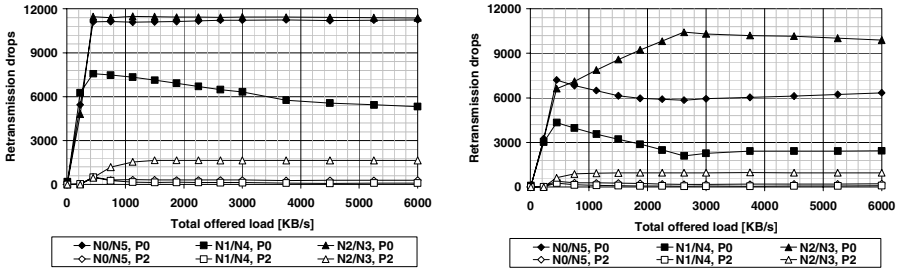


Fig. 9. Six-node line. Retransmission drops for RTS/CTS (a) disabled (b) enabled.

conclusions that with RTC/CTS enabled the hiddenness of nodes is weakly and the exposedness is strongly evident.

Duplicate drops are a result of collisions of either DATA and ACK frames (in the case of RTS/CTS disabled) or RTS and ACK frames (in the case of RTS/CTS enabled) caused mainly by the exposedness of nodes. The exact reason is that the duration of ACK frames together with SIFS is shorter than AIFS. Consequently, every exposed node can start its transmission of a DATA or RTS frame to a destination node before this destination node receives an ACK from its other neighbor. Collisions on ACK frames cause the node which does not receive the ACK to send its DATA frame once again. As a result, the node which previously sent an ACK frame (which collided) receives the same DATA frame. After the node checks that it already has this frame, it will drop it.

As can be seen in Fig. 10, with RTS/CTS disabled, a meaningful number of duplicate drops can be noticed only for N1/N4. This is because N0/N5 are not exposed at all and N2/N3 are most strongly exposed and hidden. Therefore, the frame transmissions triggered by N2/N3 are in many cases either strongly delayed, collide or are simply impossible. With RTS/CTS enabled, the number of duplicate drops decreases by half for N1/N4 and increases for N2/N3. Similarly as in the case of retransmission drops, this is because introducing RTS/CTS reduces the number of collisions of DATA frames of N2/N3, decreases their hiddenness and emphasizes the exposed nature of N2/N3.

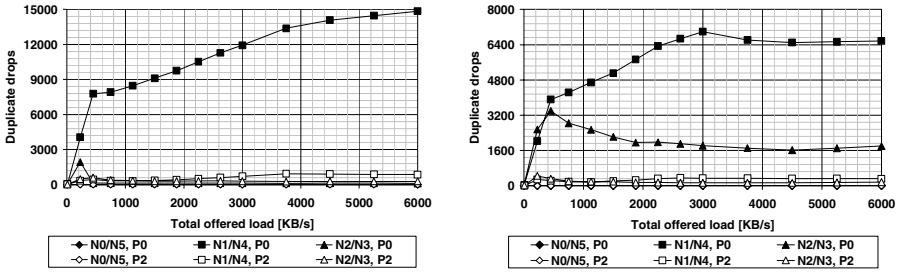


Fig. 10. Six-node line. Duplicate drops for RTS/CTS (a) disabled RTS/CTS (b) enabled.

3.8 Comparison with Star Topology Networks

When the behavior of line topology networks is compared with the behavior of star topology networks (presented in [5]) several important joined conclusions appear. Fist of all, in both cases the strong unfairness in granting medium access between particular nodes is present. Second of all, the order of throughput levels of different priority streams is reverse to the desirable ones (i.e., those expected by IEEE 802.11e). Finally, the employment of the RTS/CTS exchange does not bring meaningful changes because it does not eliminate the aforementioned problems.

The behavior of three- and four-node line topologies is most similar to the behavior of four- and five-node star topology networks. In these configurations nodes being in the middle of the network are favored over the edge nodes. In all other cases the performance of line topologies changes. This is because the importance of the exposed nature of certain nodes (especially the middle ones) grows. Additionally, also the strength of the hiddenness of the middle nodes grows as the line length increases. These two factors cause the medium access of the middle nodes to be strongly hindered. The transmissions triggered by the middle nodes either collide, are strongly delayed or even blocked.

4 General Conclusions

This paper presents a novel simulation study of five different line topology networks based on IEEE 802.11e. The impact of hiddenness and exposedness of particular nodes is commented in details. Moreover, the paper argues the usefulness of the employment of the RTS/CTS mechanism in such networks. In both cases, with RTS/CTS enabled or disabled, nodes sending high priority traffic obtain lower throughput levels than the corresponding ones with low priority streams. Furthermore, high unfairness in medium access between different line-forming nodes is stressed. The general prioritization patterns of nodes are presented in Fig. 11.

As can be easily noticed, the higher the number of nodes the more the middle ones are harmed in terms of throughput. Consequently, it can be noticed that it seems impossible for the side nodes to obtain meaningful dominance over other nodes when there are more than four nodes in a line. It can be also expected that similar behavior will occur when the line will be lengthened for RTS/CTS both enabled and disabled.

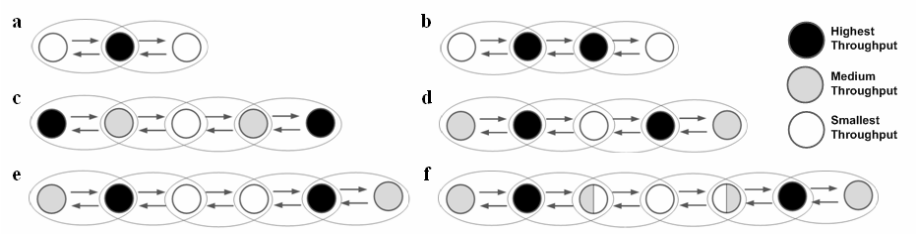


Fig. 11. Prioritization order in (a) three- (b) four- (c) five- (low priority traffic) (d) five- (high priority traffic) (e) six-, and (f) seven-node line

Additionally, the presented line topology networks are compared with previously analyzed star topology networks. Several joined conclusions are revealed and the main differences are highlighted. The cause of the differences is also explained.

Even though the presented analysis is rather thorough, there is a need for further simulations. The behavior of line topology networks should be checked when the most harmed nodes, in terms of access prioritization, will generate the high priority traffic, while the prioritized ones will generate low priority traffic. Such analysis should be done in order to check if simple changes of EDCA access parameters is a good direction in solving hidden/exposed node problems within IEEE 802.11e based networks. Additionally, other topology networks (more spontaneous than star and line) should be taken into account. Future work will also comprise an analysis of new scenarios to provide even more general conclusions. The overall aim of the planned analysis is to show which threats are most dangerous, and which EDCA factors are most important in building a new mechanism eliminating the degrading impact of hidden/exposed nodes on IEEE 802.11e.

References

1. IEEE 802.11e: Medium Access Control (MAC) Quality of Service Enhancements. IEEE Inc., New York(November 2005)
2. IEEE 802.11b: Higher-speed PHY extension in the 2.4 GHz band (1999)
3. TKN EDCA 802.11e extension (2006), `http://www.tkn.tu-berlin.de/research/802.11e_ns2`
4. Bai, X., Mao, Y.M.: The Impact of Hidden Nodes on MAC Layer Performance of Multi-hop Wireless Networks Using IEEE802.11e Protocol. In: International Conference on Wireless Communications, Networking and Mobile Computing 2007, WiCom 2007, pp. 1479–1483 (September 2007)
5. Kosek, K., Natkaniec, M., Vollero, L., Pach, A.R.: An Analysis of Star Topology IEEE 802.11e Networks in the Presence of Hidden Nodes. In: Proc. The International Conference on Information Networking 2008, ICOIN 2008, Korea (January 2008)

Interference and Congestion Aware Reservations in Wireless Multi-hop Networks

Stéphane Rousseau, Laure Lebrun, Hervé Aïache, and Vania Conan

Thales Communications, 146 Boulevard de Valmy, 92204 Colombes, France
{stephane.rousseau,laure.lebrun,herve.aiache,
vania.conan}@fr.thalesgroup.com

Keywords: wireless multi-hop, resource reservation, interference, congestion.

1 Introduction

Multi-hop wireless networks are dynamically forming networks of radio equipped nodes. Most of the early work has been motivated by scenarios where nodes are mobile, leading to both theoretical results for the capacity of the network [1] and to practical proposals for routing protocols [2]. With the recent deployment of *community* wireless networks, more specific attention has been given to multi-hop wireless networks composed of both mobile mesh clients and more static mesh routers which form the backbone of the wireless mesh and provide the clients with access to the Internet [3].

In the present paper we consider multi-hop wireless networks that require resources to be reserved across the network; such mechanisms may be needed in both the mobile and static cases. Firstly multimedia applications require different quality of service guarantees. For example a voice application would require low delay and jitter, and video streaming would need bandwidth guarantees. The second use case applies more specifically to the wireless mesh backbone. In this context, different Service Level Agreements (SLAs) can be provided to users in a wireless mesh community network. The network provider would thus need means to ensure that the expected Quality of Service [15] (in terms of average or peak bandwidth) is actually delivered to its subscribers.

Resource reservation in wireless multi-hop networks is a challenging issue especially because it involves mechanisms from different layers, especially the MAC and network layers. The present work focuses on the problem of resource reservation across the network. We consider that the MAC layer is capable of reserving resources on the links of this ad-hoc network. This means to be able to rely on TDMA-based MAC layers, such as in [10], [11], QoS aware scheduling [12] [13] or service differentiation as in 802.11e [14].

We furthermore consider that the multi-hop network runs a proactive or link state protocol that provides knowledge of the network and resource state to all nodes. Reservation may then be performed by the source, and the route is then pinned by each router along the path. The aim of the reservation scheme is thus to maximize the usage of the network (its capacity) under the constraint that each link and node can only provide a fixed limited amount of resource.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 391–400, 2008.
© Springer-Verlag Berlin Heidelberg 2008

First the reservation scheme must be congestion aware. Because of the conformation of the network and traffic demands, traffic is not routed evenly in the network, some nodes are more popular than others and then become congested. The scheme must thus take into account resource utilization to balance the reservations across the network. In doing so, one also wishes to maintain average utilization of all nodes as low as possible.

But since we are looking at wireless nodes, the reservation scheme must also be interference aware. Interferences are known to be the major limiting factor of wireless networks. The issue is that simultaneous transmissions of neighboring nodes may interfere with one another [16]. This has an impact on many different aspects: on network capacity [1] [4], transport throughput [5] and routing protocol design [6]. Interferences caused by simultaneous transmissions are also impacting resource reservation. The problem of finding a reservation path that does not degrade existing reservations is called the *Path with Remaining Capacity* problem. It was shown to be NP-complete [7] [8]. To make reservations in the network, it is thus necessary to retort to heuristics. [9] compares three heuristics, called H_1, H_{inc} and HN_1: H_1 weighs the links using an estimate of the remaining capacity to route around congested links (using a weighted Dijsktra algorithm). H_{inc} is similar to H_1 but tries incrementally less and less stringent weights. HN_1 replaces the weights of H_1 to integrate the remaining capacity of the adjacent nodes to compute the routes. They show that HN_1 performs best of the three.

The contributions of the paper are threefold:

First we introduce a model for the computation of the remaining capacity of each node in the wireless multi-hop network. The model takes into account radio interferences between neighboring nodes. It also captures capacity reductions implied by multiple-rate and robust coding schemes that are implemented at the physical layer to insure communication on pour links. This information is used by the heuristics, but is also valuable in its own right to provide evaluation grounds for other routing, transport or reservation proposals.

Second we propose and evaluate a new heuristics for resource reservation across the network. The heuristics combines the estimation of link congestion, knowledge of link quality and impact of local interferences. We show that it achieves load balancing of the reservations across the network and outperforms previous metric-based mechanisms (ETX and HN_1).

Third we show in simulation that the proposed algorithm provides a versatile mechanism that applies in contrasted scenarios. In the case of all nodes working on the same bandwidth, the scheme manages to reduce interferences, and in the case of a wireless mesh operating on non-interfering bands; it is shown to minimize congestion.

The remainder of the paper is organized as follows. First we present the interference model that we use throughout the paper. Second we present the congestion and interference aware heuristics to solve the resource reservation problem. Third we present simulation results that compare its performances with state-of-the-art solution. The paper finishes with a section on related work and a conclusion.

2 Remaining Capacity Model

As outlined in the introduction, it is critical that the MAC layer be capable of reserving resources on the links [17]. For the sake of clarity, in the remainder of the paper we will consider that the MAC layer uses time division multiplexing (TDMA), so the resources that are reserved at MAC layer are time slots. TDMA is the basic mechanism used for example in 802.11 physical layer, where 802.16 may also use Frequency division multiplexing (FDMA).

Furthermore in the model we assume that the MAC layer scheduling is omnipotent and achieves maximum concurrent transmissions among all nodes. Let us underline that the problem still remains difficult even for this bestcase simplification [18]. Finally, we consider a transport protocol that ensures fairness in terms of data rate for all of the flows in the network.

2.1 Interference Model

During a time-slot, each node can transmit packets to one or more nodes located in its sending area. In this model, if a node receives two packets at the same time and on the same frequency, interferences can appear. We have to distinguish two cases. The first one is the case in which both SNRs are equal. In this case the two packets are destroyed. In the second case one of the SNRs is considerably greater than the other one. In this case, only the packet with the smallest SNR is destroyed.

This problem of interference is a key problem in wireless networks. The crudest interference model is the Boolean link model, which considers a fixed radius of interference. In the paper, we consider a more realistic interference model taking into account the distance between the source and the destination. More the destination is far away from the source, more the link quality is altered and sensitive to noise and interferers influence.

The quality of the transmission depends on the quality of the link. This quality depends on the SNR (Signal-to-Noise Ratio). Several estimation formulas have been proposed with the following general shape (for two nodes x and y at distance d(x,y)):

$$\forall x,y \text{ nodes}, \mathrm{SNR}(x,y) = P/d(x,y)^{\wedge}\beta. \quad (1)$$

Where P is the transmission power and β, the path loss exponent, ranging from 2 for the free space propagation model to 5. In our study it is chosen equal to 3, corresponding to a peri-urban propagation model.

As one can see, the SNR decreases fast with distance. If the link has a good quality then high transmission rate, low redundancy mechanisms are used. Thus a packet of size one takes exactly one bandwidth resource. However, it is possible that the link quality is degraded (due to fading,). The physical layer will then try to compensate for this loss of quality by lowering the transmission rate and adding redundancy to send a packet of size one. Then the needed bandwidth resource using this link can be 2, 3 or 4 times more for a packet of size one. Considering for example the Physical layer of IEEE 802.11 standard, 4 modulation schemes are introduced from BPSK to 64-QAM with coding rates going from ½ to ¾.

2.2 Remaining Capacity Estimate

In this section, we show how to estimate the remaining capacity for each node. We consider a capacity C for each node. We assume a packet size is equal to one. Thus, at each time-slot, a node can transmit at most C packets or receive C packets. Moreover, this capacity is shared between the sending and receiving processes. Then, the sum of the transmitted packets and received packets cannot be greater than C. We present below the remaining capacity model that takes into account the interference assumptions given in [8].

3 Interference Aware Heuristics

3.1 Heuristic Based on ETX

ETX is a metric that evaluates the link quality. The heuristic based on ETX consists in assigning for each link a weight equal to the ETX metric. Then, to compute a new path for a request, we use the Dijkstra algorithm that returns the shortest path in terms of link quality.

This heuristic takes into account the link quality but not the load balancing. The load balancing can be taken into account considering the remaining capacity for each node. A heuristic has been proposed and we present it below.

3.2 Heuristic Based on the Remaining Capacity

The heuristic based on the remaining capacity consists in avoiding area of saturated nodes when the path is chosen. This heuristic has been proposed in a context where all 1-Hop links have the same quality. Thus, the heuristic consists in computing for each node a weight that indicates the remaining capacity of the node and the weight of its neighbours. We recall here the formulas given in [9] to compute this weigh:

$$\text{Weight}(x)=1/Cx+\Sigma 1/Cy \tag{2}$$

Where Cx is the remaining capacity of the node and Cy the remaining capacity of its 1-HOP neighbours.

After having assigned a weigh for each node, a path is computed using the Dijkstra algorithm to find the shortest path in terms of remaining capacity. Once a new request is accepted, a new weight is computed for each node with the new remaining capacity.

3.3 New Heuristic

This heuristic combines the heuristic based on ETX and the one based on the remaining capacity. The goal of this heuristic is to return the better path for a request conserving a good load balancing in the network for next requests.

The first step of this heuristic consists in assigning a weight for each link. The formula that we proposed is the following one:

$$\text{Weight}(x)=\alpha * (ETX^{\beta}) * (1/Cx+\Sigma 1/Cy) \tag{3}$$

Where α and β are parameters.

In this article we don't discuss on those parameters and we put α=1000 and β=3. Thus, ETX metric is more considered than the remaining capacity when the path is computed. However, if two paths have the same cumulated weight in terms of ETX, the heuristic returns the one with the most remaining capacity.

Then, we use the Dijkstra that returns the shortest path in terms of link quality and remaining capacity.

4 Simulation and Analysis

4.1 Model Description

We consider wireless network composed of 100 nodes. These nodes are located in an 1200m*800m area. All nodes are randomly placed. We assume the noise is insignificant. Thus, we assume the SNR depends only on the distance between the source node and the destination node. Then, the SNR between two nodes is compute with the following formula proposed in section 2.

We distinguish four degrees of link state quality. The most the SNR is great the most the quality of the link is good. When the quality is not good enough, we assume that sending a packet of size 1 can cost in real more than 1 unit of resource radio. According to this model, a packet of size 1 can take either 1, or 2, or 3, or 4 times more resource radio to ensure its good reception.

If the SNR is very bad, then no packet can be transmitted using this link.

In this model, we assume each node has a capacity of 8000kb/s. This capacity can be used either for transmit packets or receive some. For example, a node that receives 200kb/s and transmits 100 kb/s has a remaining capacity of 7700kb/s.

At the beginning of the simulation, the network is empty. Then, 2000 requests are generated and treated one after the other. All requests have the same characteristics:

- An origin node that is randomly chosen.
- A destination node that is randomly chosen.
- A capacity (equal to 1 kb/s)

When a node treats a request, all requests accepted and the reserved resources radio are known but it is not the case of the next requests.

All decisions are done within the origin node, then after one step all nodes in the network update their information about link state and remaining capacity.

The goal of these simulations is to compare three heuristics for the reservation of resource radio that takes into account interferences. In order to be the most general as possible, we first compare those heuristics when all nodes have a different frequency (MESH network) then when all nodes have the same frequency (Ad-Hoc network model.

In order to compare the three heuristics, we focus on the load balancing of the network and the distribution of remaining capacity of each node. Recall that we are in an on-line context. It means that all past reservations are known but none of the next requests can be anticipated. Then, a way to ensure a good use of the network is to maximize the minimum remaining capacity considering all nodes.

4.2 Evaluation of Performances in Terms of Load Balancing

To compare the three heuristics, we propose to focus on the load balancing in the network when all requests have been treated. At the end of these simulations all requests have been accepted. Thus, for the same set of requests, we compare the amount of remaining capacity of each node. Two kinds of curve are presented here.

- In the first one the x coordinate represents the location of nodes and the z coordinate represents the remaining capacity for each node. With this kind of curve, we underline the difference between the most loaded node and the less loaded one. Then we focus on the repartition of the load among all nodes.
- In the second kind of curve the x coordinate and the y coordinate indicate the location of nodes in the whole area. We color sub-areas of the network according to the load of the nodes located in it. When an area is composed of loaded nodes, this area is black and when the area is composed of unloaded nodes, it is white. Using this representation, we show where most loaded nodes are located.

Considering a Mesh Network
We consider all nodes have a different frequency. Then, no interferences appear between two nodes in this network.

Results of the heuristic based on ETX metric
Results are shown on Figures 1 and 2. If we consider Figure 1, the most loaded node has about 500 kb/s remaining capacity and the less loaded one has about 5000 kb/s remaining capacity. If we consider Figure 2, the heuristic based on ETX chooses the shortest path for all requests. If a node is saturated (i.e. no remaining capacity on this node) then the heuristic gives another shortest path to connect the origin node and the destination node. Thus, most of accepted requests go from the origin node to the destination node via the center of the network. When the center of the network is saturated, then next accepted connections go round the center. Thus, most of nodes with few remaining capacity are located at the center of the network. Outlying nodes have more remaining capacity.

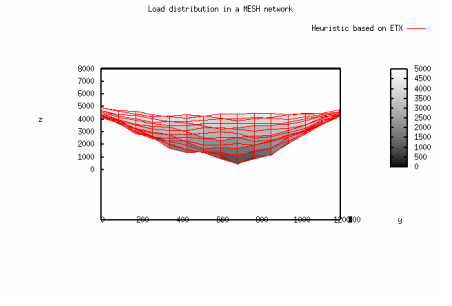


Fig. 1.

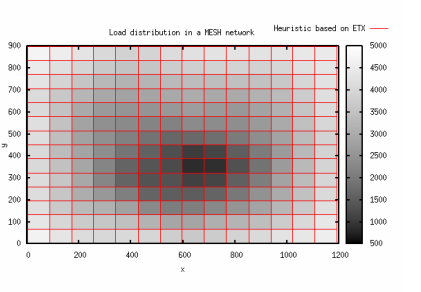


Fig. 2.

Results of the heuristic based on remaining capacity

Results are shown in Figure 3 and 4. The most loaded node has 0 kb/s remaining capacity. Thus, using the heuristic based on remaining capacity, some of the nodes in the network are saturated. It means, those nodes cannot accept requests anymore. The less loaded node has about 5000 remaining capacity. Here, we can underline the gap between the most loaded node and the less loaded one. The distribution of the load within the network is not equal from a node to another. Let us focus on the location of loaded nodes in the network. Figure 4 shows the distribution of load in the network according to the geographic location. Unlike the heuristic based on ETX where the center of the network tends to be saturated, with the heuristic based on the remaining capacity, saturated nodes are located around the center. This result is due to the heuristic that tries to avoid area with a lot saturated nodes. Thus, the first accepted requests are routed round the center in order to balance the load. But, because this heuristic does not consider link quality, some of computed paths generate a lot of interferences. Then, all nodes around the center are saturated and none of the next requests can be connected the origin node and the destination node via the center. A kind of ring is drawn around the center and the load of the network is concentrated on it.

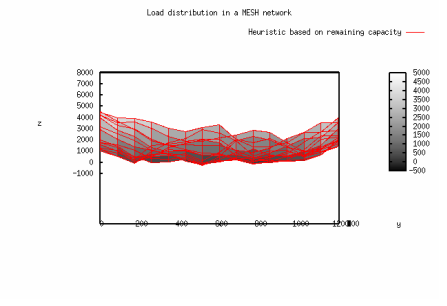


Fig. 3.

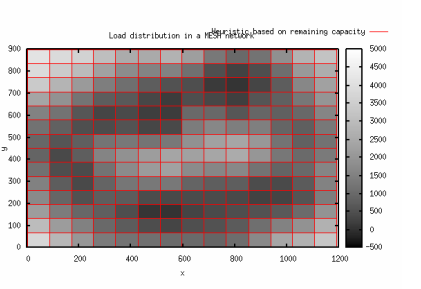


Fig. 4.

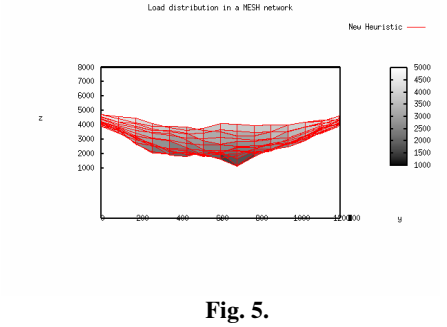


Fig. 5.

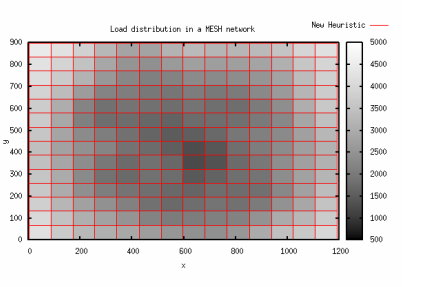


Fig. 6.

Results of the proposed heuristic

Now, we present the results that we obtained with the heuristic we propose in this article. This heuristic is based not only on the link state quality but also on the remaining capacity of each node. In Figure 5, the curve represents the gap between the most

loaded node and the less one. The most loaded node has not been saturated yet; its remaining capacity is about 1000 kb/s. The less loaded node remaining capacity is about 5000 kb/s. Thus, the distribution of load in the network is quite fair among nodes. In Figure 6, we present the repartition of the load in the area. A small subset of loaded nodes is concentrated at the center of the network. Most of the load is well equitably distributed around the center of the area.

Conclusion on the MESH network

In a MESH network, it is possible to organize nodes and assign different frequency in order to limit the number of interferences. In the simulation model, we assume that all nodes have a different frequency. Then, when a node transmits a packet, no interference can appear with the other nodes transmissions. However, we still are in a wireless network, so each node has to share resource radio with all 1-Hop neighbors for communications to and from them. This is the main constraint in this model of MESH network. Considering only this constraint, we compare three heuristic. The heuristic based only on ETX, has good performances and the one based only on the remaining capacity is not so good. Combining the two criteria for the choice of the path for each request, the performances are better than the two others. Indeed, the link state quality seems to be an important criteria but it is necessary to take into account of the remaining capacity in the path choice.

Considering an Ad-hoc Network

In an Ad-Hoc network, assigning different frequency for each node is not so easy. Indeed, nodes are mobile and the assignment has to be dynamically done. Obviously, it is possible to add a signaling protocol in order to do it.

Here, we consider all nodes with the same frequency. However, we assume all nodes can be considered as static (i.e. nodes are not mobile). This kind of Ad-Hoc network is often used when a network has to be deployed rapidly but once the network is deployed, nodes don't move anymore.

In this context, it is very important to take into account the interference model described in Section 2. Now, we compare the three heuristics when all nodes have the same frequency. In Figure 7, 9 and 11, we present the gap between the most loaded node and the less one. Proposed heuristic is better than the heuristic based on ETX.

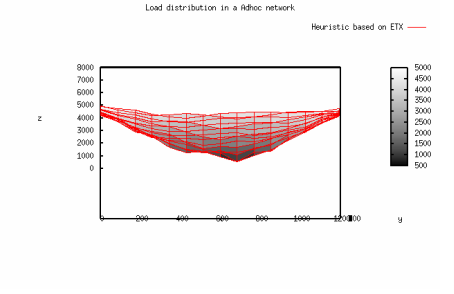


Fig. 7.

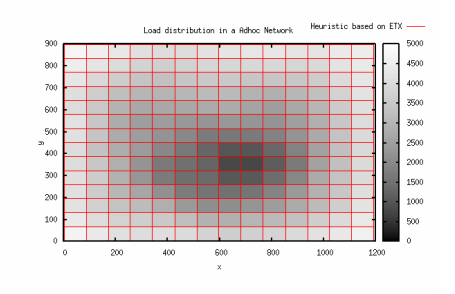


Fig. 8.

The heuristic only based on the remaining capacity, performances are not good. We obtain results quite similar than those obtained in the MESH network. When we focus on the distribution of area with high density of loaded nodes, we also obtain similar results. The heuristic that we proposed here, takes into account two very important criteria. The first one is the quality of links for transmission, and the second one is the load balancing in the network.

To conclude, those two criteria have to be considered when we want avoid congestion in a wireless network.

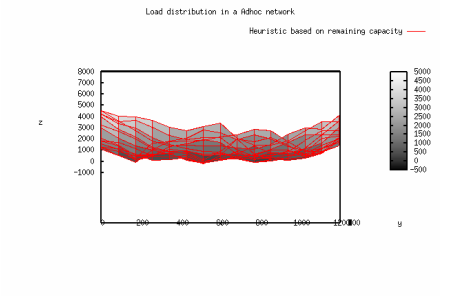


Fig. 9.

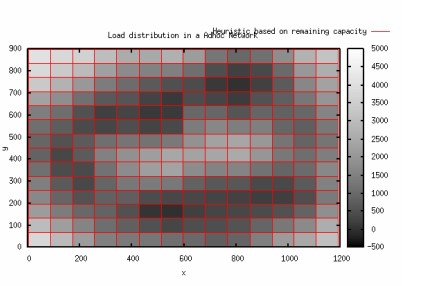


Fig. 10.

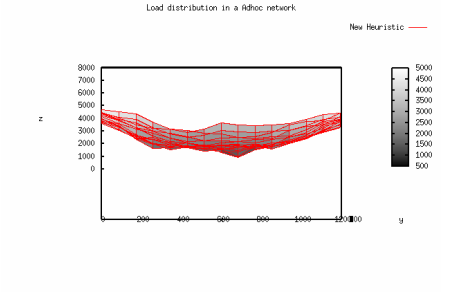


Fig. 11.

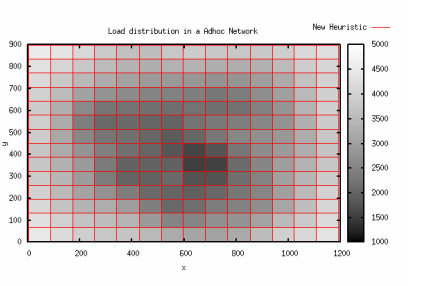


Fig. 12.

5 Conclusion

Resource reservation in wireless multi-hop networks is a challenging issue. In this work, we focus on the problem of resource reservation across the network and also the problem of congestion in the network.

Avoiding congestion consists in maximizing the minimum remaining capacity of nodes in the network. In this work, we propose a heuristic to avoid interferences and ensure a good use of the resources in the network. We compare the performances of this heuristic and two other heuristics by simulations. The proposed heuristic is better than the two others in the case a MESH network and also in the case of an Ad-Hoc network.

In further work we propose to discuss about the two parameters given in Section 3 in order to improve these results.

Acknowledgements

The research has been performed in EU FP6 Integrated Project Chorist N°. 033685.

References

1. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Trans. on Info Theory (2000)
2. IETF Mobile Ad-Hoc Networking group, http://www.ietf.org/rfc/rfc2501.txt
3. Akyildiz, I., Wang, X., Wang, W.: Wireless mesh networks: a survey. Computer Networks 47, 445–487 (2005)
4. Jain, K., Padhye, J., Padmanabhan, V., Qiu, L.: The impact of interference on multi-hop wireless network performances. In: MOBICOM (2003)
5. Fu, Z., Zerfos, P., Luo, H., Lu, S., Zhang, L., Gerla, M.: The impact of multihop wireless channel on TCP throughput and loss. In: INFOCOM (2003)
6. De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: Performance of multihop wireless networks: shortest path is not enough. ACM SIGCOMM Computer Communication Review 33(1), 83–88 (2003)
7. Guerin Lassous, I., Viennot, L., Bertet, K.: Impact of interference on bandwidth reservation for ad hoc networks: a theoretical study. INRIA Research Report RR2001-17 (2001)
8. Jacquet, P., Mans, B., Georgiadis, L.: Bandwidth reservation in multihop wireless networks: complexity and mechanisms. In: INRIA Research Report RR4876 (2003)
9. Allard, G., Jacquet, P.: Heuristics for bandwidth reservation in multihop wireless networks. In: INRIA Research Report RR5075 (2004)
10. Chen, T.W., Tsai, J.T., Gerla, M.: QoS Routing Performance in Multihop Multimedia Wireless Networks. In: Proceedings of IEEE International Conference on Universal Personal Communications (1997)
11. Lin, C.R., Liu, C.C.: An On-demand QoS Routing Protocol for Mobile Ad Hoc Networks. In: IEEE Global Telecommunications Conference (2000)
12. Kanodia, V., Li, C., Sabharwal, A., Sadeghi, B., Knightly, E.: Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints. In: MobiCom 2001 (2001)
13. Yang, Y., Kravets, R.: Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks. In: Technical Report UIUCDCSR-2004-2446 (2004)
14. Mangold, S.,Choi, S., May, P., Klein, O., Hiertz, G., Stibor, L.: IEEE 802.11e Wireless LAN for Quality of Service. In: Proceedings of European Wireless (2002)
15. Chen, L., Heinzelman, W.B.: QoS-aware routing based on bandwidth estimation for mobile ad hoc networks Selected Areas in Communications. IEEE Journal 23(3), 561–572 (2005)
16. Reis, C., Mahajan, R., Rodrig, M., Wetherall, D., Zahorjan, J.: Measurement-based models of delivery and interference in static wireless networks. In: SIGCOMM, pp. 51–62 (2006)
17. Jun, J., Sichitiu, M.: The nominal capacity of wireless mesh. networks. IEEE Wireless Commun. Mag. 10(5), 8–14 (2003)
18. Chiu, C.-Y., Kuo, Y.-L., Wu, H.-K., Chen, G.-H.: Bandwidth constrained routing problem in multi-hop wireless networks. In: MSWiM 2006: Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems, pp. 365–369 (2006)

Low-Cost and Accurate Intra-flow Contention-Based Admission Control for IEEE 802.11 Ad Hoc Networks

Abdelouahid Derhab

Department of Computer Engineering, CERIST, Rue des 3 frères Aissou, Ben-Aknoun, BP 143 Algiers, 16030 Algeria

Abstract. In this paper, we propose a new admission control method for IEEE 802.11 ad hoc networks, called Low-cost and Accurate Admission control (LAAC). The proposed method has two variants: LAAC-Power and LAAC-CS. LAAC-Power estimates channel bandwidth availability through high power transmissions and LAAC-CS through passive monitoring of the channel. Due to the shared nature of the wireless medium, contention occurs among the nodes along a multi-hop path, which leads to intra-flow contention. LAAC accurately estimates the intra-flow contention. In addition, an analytical study demonstrates that LAAC achieves optimal results in terms of overhead and delay compared to the existing intra-flow contention-based admission control methods. LAAC also utilizes two criteria for accepting flows: one during the route request phase and the other during the route reply phase, which helps to reduce message overhead and avoid flooding route requests in hot spots. Simulation results show that LAAC-CS outperforms LAAC-Power in terms of packet delivery ratio, throughput, message overhead, and energy consumption.

1 Introduction

The increasing use of real-time applications such as: teleconferencing and on-demand multimedia retrieval, as well as the adoption of IEEE 802.11 technologies in ad hoc networks raise the issue of how to ensure service guarantee in such environments characterized by unpredictable topology network, shared wireless channel, and which impose different challenges on supporting real-time applications with appropriate QoS.

The admitted flows in the network must not exceed the network capacity. To do so, the wireless channel must be kept from reaching the congestion point. This goal is hard to achieve since the channel is not only shared between nodes that can communicate with each other directly, but extends to all nodes within a certain range, called carrier-sensing range (CSR), through channel access contention. This range is typically much larger than the transmission range. Nodes that are within carrier sensing range detect a transmission but may not be able to decode the packet. Nodes within the sender's transmission range are considered its neighbors, and those which are within the CSR of a sender are called its carrier-sensing neighbors (CSN).

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 401–412, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The admission control must ensure that the network should have sufficient resource before admitting any new flow. Moreover, the flow should not degrade the QoS of existing flows. In IEEE 802.11 MAC protocol, all the CSN of the sender are unable to initiate a packet transmission while the sender is transmitting. Due to the shared nature of the wireless medium, A node's transmission consumes bandwidth at all nodes within its vicinity (i.e., carrier-sensing range). Let us consider a flow f with a bandwidth requirement, B_{req}[1], going through a given route. Multiple nodes on the route may locate within the carrier-sensing range of a given node S, and they all contend for bandwidth. The number of these nodes is called the *contention count* of the route and is denoted as CC. To make admission control decisions over a multi-hop path, it is not enough to only consider the bandwidth available at a single node, since the effective bandwidth consumed by the flow at node S is: $(CC \times B_{req})$.

In this paper, our original contributions are the following. First, we propose a new admission control method called LAAC, and which has two variants: LAAC-Power and LAAC-CS. Second, unlike other intra-flow admission control methods, LAAC guarantees both an accurate estimation of the contention and incurs the lowest cost in terms of message overhead and energy consumption. Using two admission control criteria, the message overhead is reduced. In addition, LAAC does not incur an additional delay over that incurred by regular route discovery to make a multi-hop admission control decision.

The rest of the paper is organized as follows: In Section 2, we discuss related works. Section 3 presents a new admission control method. In Section 4, we analyze the performance LAAC as well as other intra-flow contention-based admission control methods. Section 5 compares the performance of LAAC-Power and LAAC-CS. Section 6 concludes the paper.

2 Related Work

CACP [8] is the first work to introduce the concept of c-neighborhood available bandwidth, which refers to the available bandwidth at a node's CSNs. The admission control is integrated with the route discovery procedure of DSR routing protocol [4]. To ensure that all nodes affected by the transmission of the traffic flow have enough available resources to allow the flow to be admitted, CACP proposes two variants: CACP-Power and CACP-CS. In CACP-Power, a node that receives a Route Reply (RREP)packet, broadcasts using a high power transmission an admission request message, which carries the full route of the flow, to its CSNs. Upon reception of the message, nodes calculate their CC using their known CSN and the the identity of the nodes on the route. In CACP-CS, channel availability is estimated through passive monitoring using a threshold called the Neighbor-carrier-sensing Threshold, which is lower than the Carrier-sensing Threshold. A node can then extend its measurement range to enclose the carrier-sensing ranges of all its CSNs. It assumes that any transmission activity in its neighbor-carrier-sensing range consumes bandwidth at all of

[1] The equations to derive B_{req} from the application rate is given in [8].

its CSNs, which leads that the admission control rejects flows whose bandwidth consumptions are not beyond the capacity of the network. However, CACP has several drawbacks. First, the admission control decision is delayed at each node in order to receive possible rejections before forwarding the reply. Second, CACP operates only on a source routing protocol such as DSR [4] that holds the entire route. Third, it does not propose any strategy to handle mobility and loss of QoS guarantees. Fourth, CACP does not explain how the bandwidth at the node's CSN is released when the flow is rerouted or terminated.

Sanzgiri et al. [7] describe two methods, PRP and RRT, to obtain the CC, in which each node records the duration of the received signal strength corresponding to a packet in a carrier sensing table. Although the packet cannot be decoded, its size can be inferred from its duration. However, PRP and RRT suffer from some drawbacks. A node inside the sender's carrier-sensing range cannot determine the bandwidth consumption because it does not know the value of B_{req}. Moreover, the node that is part of the flow cannot make an accurate admission control decision because it ignores the effects of contending flows. Finally, counting sensed packets of a particular duration can produce erroneous results in the case of retransmissions or collisions at the MAC layer.

To compute CC, AAC [2] and TAC-AODV [1] consider that the carrier-sense range is more than twice the size of the transmission range. Therefore, every node on the path generally interferes with, at most, two upstream and downstream nodes, which means that the nodes are supposed to have the same transmission range. However, this assumption is not always true since a node can increase or decrease its transmission power depending on its own purposes, and hence the CC calculation as it is proposed by the protocols is not accurate.

MACMAN [5] uses the same method described in CACP [8] to calculate CC. It tries to improve the performance of the admission control by maintaining multiple paths to the destination. This allows a source to quickly switch to an alternate path that can support the flow if the current path becomes unusable. To avoid the accumulation of stale routes that no longer can provide the required QoS, MACMAN continuously monitors each alternative route in the cache. To do so, it sends Periodic Route Capacity Query (RCQ) messages along each of the backup paths towards the destination. The disadvantage of this method is that it generates an important overhead on monitoring path that might never be used.

3 Low-Cost and Accurate Admission Control (LAAC)

Our admission control is integrated with a route discovery procedure of a reactive routing protocol similar to AODV [6]. In LAAC, each node i maintains the flow table FT_i that stores for each flow f circulating in its carrier sensing range: (1) the contention count $CC_{i,f}$, and (2) the list of the carrier-sensing neighbors which transmit the flow f. The admission control is performed in two phases of route discovery: (1) route request phase and (2) route reply phase. The aim of performing the admission control during the route request phase is to reduce

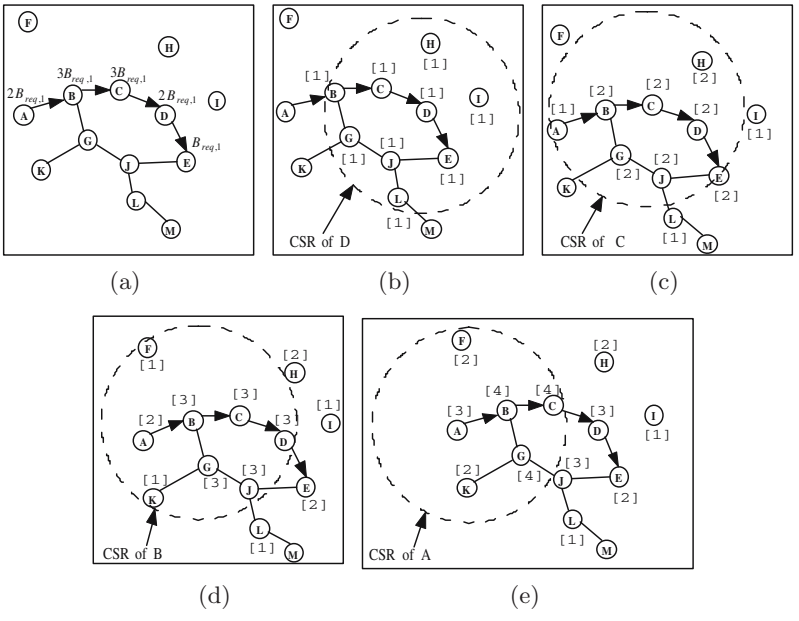


Fig. 1. Admission control acceptance in LAAC-Power

the overhead caused by propagating the RREQ in the whole network. If the available bandwidth at a given node is smaller than the bandwidth requirement of the flow, the admission control fails. The Bandwidth reservation is only carried out during the route reply phase.

3.1 Admission Control during the Route Request Phase

When a source node wants to send a data flow f to its destination node, it broadcasts a RREQ packet to its neighbors. The RREQ contains the bandwidth requirement $B_{req,f}$. Each node that receives the RREQ performs an admission control to check if enough bandwidth is available for the flow. If the admission control fails, the RREQ packet is dropped. If the admission control succeeds, the route RREQ packet can continue its propagation through the network. The question that may arise is how a node i can determine the bandwidth required by the flow f during the request phase without knowing the contention count $CC_{i,f}$. To deal with this issue, we propose to give the lower bound of $CC_{i,f}$. This bound is based on the solution proposed in [10].

In IEEE 802.11, nodes cannot transmit and receive data simultaneously. For any packet transmission, it consumes the same amount of bandwidth resource at all the carrier sensing neighbors, because they should not be able to use that period of time for other transmissions. During route request phase, each node i does not know its carrier sensing neighbors, it only knows the previous node from which it has received the RREQ packet. It also knows its status (i.e., source node, intermediate node or destination node). Based on this knowledge, the

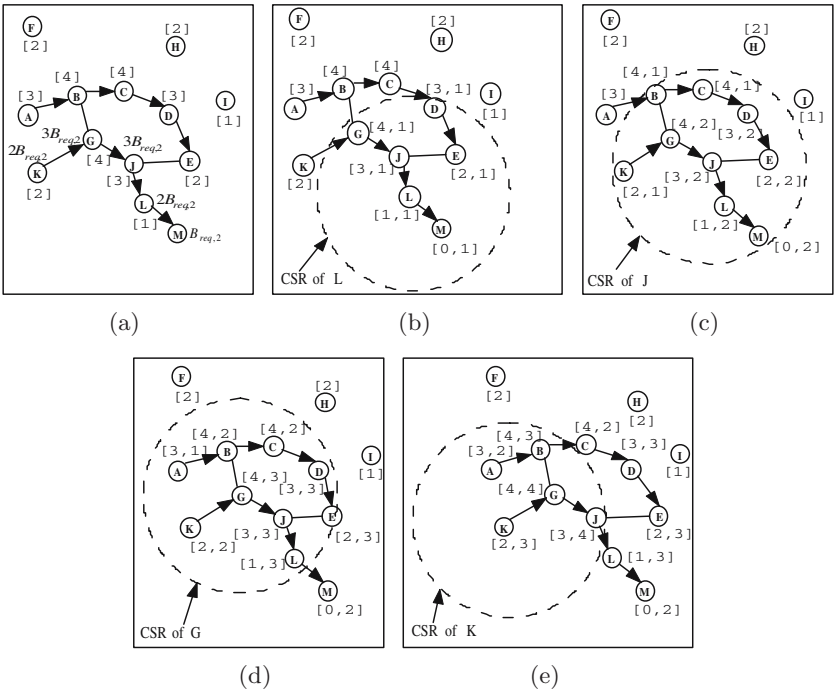


Fig. 2. Admission control rejection in LAAC-Power

lower bound of $CC_{i,f}$, denoted by $LCC_{i,f}$ can be estimated under the following conditions:

- If i is the source node (e.g, node A in Figure 1(a)), it requires $B_{req,f}$ for sending the data, and another $B_{req,f}$ is consumed by its next-hop neighbor. So, $LCC_{i,f} = 2$.
- If i is the destination node (e.g, node E in Figure 1(a)), $B_{req,f}$ is consumed by its previous node. So, $LCC_{i,f} = 1$.
- If i is the last intermediate node (e.g, node D in Figure 1(a)), it requires $B_{req,f}$ for sending the data, and another $B_{req,f}$ is consumed by its previous node. So, $LCC_{i,f} = 2$.
- If i is not the last intermediate node (e.g, node B and C in Figure 1(a)), it requires $B_{req,f}$ for sending the data, and another $B_{req,f}$ is consumed by it previous node and its next-hop neighbor. So, $LCC_{i,f} = 3$.

To admit a new flow f during the request phase, the required bandwidth $B_{req,f}$ for f must meet the following condition: $B_{av} > LCC_{i,f} \times B_{req,f}$.

The variable B_{av} in the condition denotes the available bandwidth. In Figure 1(a), node A wants to introduce a new traffic flow 1 to node E requiring $(\frac{B}{7})$ bits/s, such that: B denotes the channel capacity. The route obtained during the route request phase is shown as a sequence of directed links, and the respective minimum bandwidth requirements are shown adjacent to each node of this route.

3.2 Admission Control during the Route Reply Phase

When the destination node receives the RREQ packet, it sends a RREP back to its previous node (i.e., the next hop toward the source), denoted by *target*. If multiple requests arrive at the destination, the destination only sends the RREP along one route. The other routes are cached for a short period of time as backup in case the first RREP does not reach the source due to link breakage or admission failure. In the reply phase, LAAC can use one of the two variants: LAAC-Power or LAAC-CS. In LAAC-Power, reply packets are sent using a larger transmission power level than the transmission power level used for normal data transmission. Using this approach, the reply packets from the sender can reach all of its c-neighbors. In LAAC-CS, channel availability is estimated in the same way as suggested in [8].

LAAC-Power. In LAAC-Power, a node that receives the RREP packet, executes the pseudo-code presented in Algorithm 1.

If there is enough available bandwidth for the flow f, a soft reservation of bandwidth is set up in the node and a RREP packet is forwarded to its previous node using a high power packet transmission. For example, in Figure 1(b), nodes B, C, E, H, I, J, and L, which are CSN of node D, set their $CC_{i,1}$ to 1 after receiving a RREP packet from node D. The respective contention counts of the flows are shown adjacent to each node. As the reply packet traverses nodes C, B, and A, each node i that receives the high power RREP packet transmission, increases its $CC_{i,f}$ by 1 (See Figures 1(c), 1(d), 1(e)).

In Figure 2, node K wants to introduce a new traffic flow 2 to node M requiring $\frac{B}{7}$ bits/s. After the admission control has succeeded during the route request phase (See Figure 2(a)), node M broadcasts a reply packet with $target = K$ (See Figures 2(b), 2(c), 2(d)). Upon receiving the reply packet, the source node K finds that the total reserved bandwidth is: $(2B_{req,1} + 3B_{req,2}) = (\frac{5}{7})B$. So, it broadcasts a reply packet using a high power packet transmission. When node G receives such a message (See Figure 2(e)), it finds that $(4B_{req,1}+4B_{req,2}) = (\frac{8}{7})B$. It concludes that flow 2 will hinder the existing flow 1. Then, it sends a *Reject* packet to K, which will send an *Error* packet to M.

To refresh or release the bandwidth reservation, we suggest to encapsulate two bits in the IP option of every data packet, which are:

- Bit M (*More*), it is set to 1 if the flow contains other packets that need to be transmitted. Otherwise, M is set to 0 if the the packet is the last one of the flow.
- Bit HP (*High Power*): It is set to 1 if the data packet needs to be transmitted at high power level.

After a bandwidth along the route is established, the source node starts sending data packets with $(M, HP) = (1, 0)$, which indicates that the corresponding flow contains other packets, and the data packet should be sent to the destination node using a normal power packet transmission. To refresh the existing soft-reservation at nodes within the carrier sensing range of the flow, the source

Algorithm 1. LAAC-Power at node i

When *i receives a RREP(target) from j*

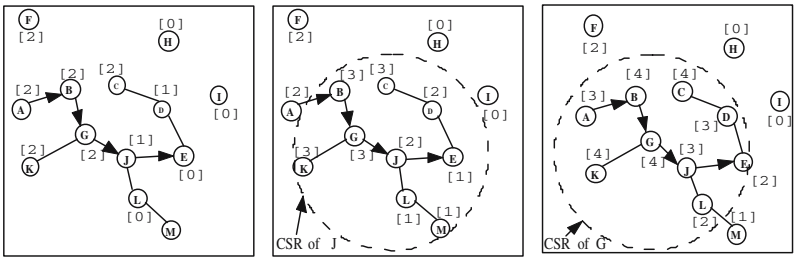
if $(B - \sum_{g \in FT_i} (CC_{i,g} \times B_{req,g}) > B_{req,f})$ **then**
  if (f exists in the flow table) **then**
    $CC_{i,f} := CC_{i,f} + 1$;
  else
    Create a new entry for f in the flow table;
    $CC_{i,f} := 1$;
  end if
  if $(i = target)$ **then**
    if $(i = source\ node)$ **then**
      $target := \phi$;
    else
      $target :=$ *previous node in the route*;
    end if
    Broadcast RREP(target) using a high power packet transmission
  end if
else
  if $(i = target)$ **then**
    Send ERROR packet toward the destination using a high power packet transmission;
  else
    Send Reject to j using a high power packet transmission;
  end if
end if

node periodically sends a data packet with $(M, HP) = (1, 1)$, which means that the packet should be sent to the destination node using a high power packet transmission. The last data packet is sent with $(M, HP) = (0, 1)$. Upon receiving this packet, each intermediate node releases the bandwidth associated with the flow f, and sends in turn the data packet using a high power packet transmission. In this manner, the bandwidth reserved at nodes within the carrier sensing range of the flow is also released.

LAAC-CS. In LAAC-CS, a passive approach is used to obtain c-neighborhood available bandwidth. The node that receives the route reply directly estimates its c-neighborhood available bandwidth using the equation presented in [8] and compares it with the bandwidth consumption of the flow to make admission decisions.

3.3 Node Mobility

If the link between two nodes of the flow route fails, e.g, nodes B and C in Figure 3(a), the bandwidth reservation along the partial route $[C, E]$ is released. Moreover, a node decreases its $CC_{i,f}$ by 1 for each node that belongs to both the partial route and the list of the carrier-sensing neighbors which transmit the flow f (See Figure 3(a)). As AODV is not the source routing protocol, node



(a) (b) (c)

Fig. 3. Flow restoration

B does not need to notify the source about this event, it locally tries to find an alternative route toward the destination, and the bandwidth reservation is established using the same method explained in Section 3 (See Figure 3(b) and Figure 3(c)). If a node suffers from QoS violation due to the mobility of some nodes, and consequently their flows, in its vicinity, it will send *QoS Lost* message toward the source node. Upon reception of this message, the source node will interrupt the generation of its flow. After a back-off random time, the source node will generate a new RREQ for the interrupted flow in order to discover a new route to fulfil its request.

4 Analytical Comparison

In this section, we analyze the performance of the proposed admission control method and compare it with CACP, PRP, RRT, AAC, TAC-AODV, and MAC-MAN. The performance is studied under the following metrics: the number and size of control packets, the additional delay incurred in making the flow admission control decision, the accuracy of CC calculation, and the energy complexity, which measures the energy required to perform a successful admission control. Note that this performance is for a single flow. The results of comparison are shown in Table 1. In the table, we use the notations given in [7], which are as follows: N and M denote the number of nodes in the network and the number of nodes on the path respectively. Q, P, S, I and J denote the size of RREQ, RREP, RPRM, RREQ tail in RRT, node ID, and short integer respectively. D_1 and D_2 are constants used in CACP-Power and PRP respectively. We assume that the nodes are randomly distributed in a region of area A. The node density remains constant when the number of nodes increases, and the area A grows with N. Since the expected distance of two uniformly sampled points within a square of size $a \times a$ scales with a [3], it is expected that the number of hops between two random nodes increases proportional to $\sqrt{N}$. We also assume that the Q, P, and I are proportional to $\log N$. The energy dissipated to transmit K bits using a normal power, and a high power transmission, are proportional to $O(K)$, and $(\alpha \times K)$ respectively. If we hold J, T, and α as constants, therefore we get the energy complexities shown in Table 1.

Table 1. Comparison of intra-flow contention-based admission control methods

Metrics	CACP-Power	CACP-CS	RPR	RRT
RREQ sent	N	N	N	N
RRRP sent	M	M	M	M
Other packet sent	M(High Power)	0	M	0
RREQ size	$Q + M \times I$	$Q + M \times I$	Q	$Q + M \times J + T$
RREP size	$P + M \times I$	$P + M \times I$	$P + J$	$P + M \times J$
Other packet size	$M \times I$	0	S	0
Extra delay	$M \times D_1$	0	D_2	0
Energy complexity	$O(N^{\frac{3}{2}} \log N)$	$O(N^{\frac{3}{2}} \log N)$	$O(N \log N)$	$O(N^{\frac{3}{2}})$
The accuracy of CC calculation	Yes	N/A	No	No

Metrics	AAC/TAC-AODV	MACMAN	LAAC-Power	LAAC-CS
RREQ sent	N	N	N	N
RRRP sent	M	M	M(High Power)	M
Other packet sent	0	N(High Power)	0	0
RREQ size	Q	$Q + M \times I$	Q	Q
RREP size	P	$P + M \times I$	P	P
Other packet size	0	I	0	0
Extra delay	0	0	0	0
Energy complexity	$O(N \log N)$	$O(N^{\frac{3}{2}} \log N)$	$O(N \log N)$	$O(N \log N)$
The accuracy of CC calculation	No	Yes	Yes	N/A

In CACP and MACMAN, RREQ and RREP carry the IDs of the nodes on the route. Thus, the control information piggybacked onto the packets are of the size of $M \times I$. As for PRP, its RREP packet contains the length of the probe packet J sent by the destination. RREQ in RRT carries the lengths of the tails appended by nodes on the path, which causes the packet size to increase by $M \times J$. Additionally, the RREQ packet carries the tail appended by the last node traversed, which causes a further increase of T in the packet size. Our method, AAC and TAC-AODV, on the other hand, do not piggyback any additional control information onto RREQ or RREP.

The forwarding of the RREP in CACP-Power is delayed at each intermediate node by D_1 time units. So, the extra delay incurred to make multi-hop admission control decision is $M \times D_1$. As for PRP, the RREP is delayed D_2 time units by the destination. CACP-CS, RRT, AAC, TAC-AODV, MACMAN and our LAAC all of which require no additional delay over that incurred by the route discovery procedure.

CACP-Power, MACMAN and LAAC-Power can make more accurate admission control decision and CC calculation than the other intra-flow contention-based admission control methods. For example, AAC, TAC-AODV assume that all nodes have the same transmission and carrier-sensing range and hence each node on the path has, at most, two upstream and downstream c-neighbor nodes. This change is not true because nodes are able to change the size of their transmission range. Therefore, AAC and TAC-AODV cannot give an accurate estimation of CC in case of heterogenous ad-hoc network. RPR and RRT do not give an accurate calculation of CC because of several reasons: First, they ignore the effects of contending flows. Second, in order to make a correct admission

control decision, nodes need to know the resources that a flow will consume if admitted, RPR and RRT do not explain how a node can obtain the value of B_{req}. Third, it is not explained how a node that senses packets can distinguish between MAC control packets that have fixed sizes and other packet, and hence the assumption that each node transmits packets using a unique duration is not true.

From this study, we can conclude that among the intra-flow contention-based admission methods presented earlier, LAAC appears to be the one that ensures two properties: (1) it incurs the lowest cost in terms of message overhead, energy consumption, extra delay, and (2) it accurately estimates CC.

5 Simulation Results

In this section, we study the performance of LAAC-Power and LAAC-CS using GloMoSim simulator [9]. Our simulation environment is characterized by 25 nodes moving in the area of 1000m×1000m, with random initial nodes' location. Nodes move according to the waypoint mobility model. In this model, a node randomly selects a location and moves toward it with a constant speed uniformly distributed between zero and a maximum speed $Vmax$, then it stays stationary during a pause time of 1 second before moving to a new random location. In the Glomosim implementation, radio transmission range is set to 376m and the carrier-sensing range is set to 688m. The bandwidth of the channel is 2 Mbps.

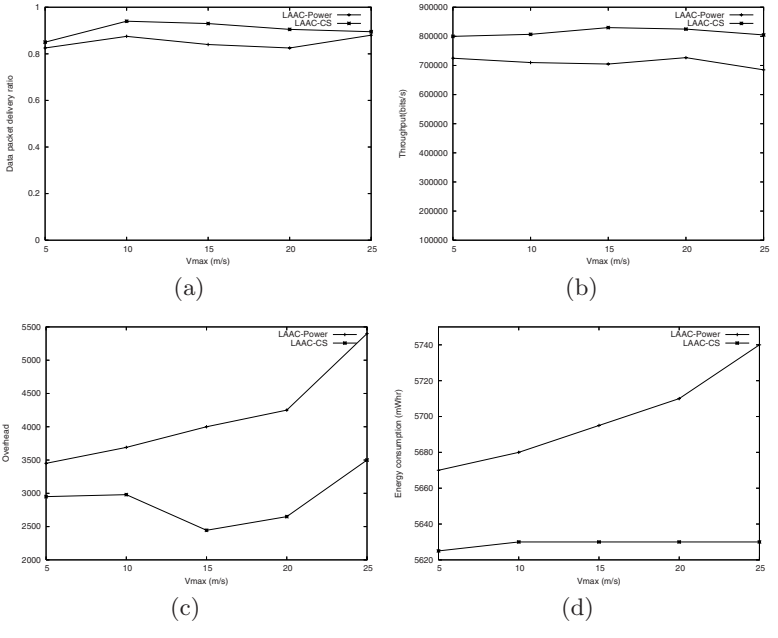


Fig. 4. LAAC Performance

Six pairs of nodes are randomly chosen to establish connections with a 512B 100 packets/s CBR traffic source. The simulation runs for 900 seconds. We evaluate the performance of LAAC-Power and LAAC-CS using the following four metrics:

- *Data packet delivery ratio:* This is the fraction of data packets sent by a source node that reach the destination.
- *Message overhead:* It measures the number of messages generated by the routing protocol as well as the control admission.
- *Throughput:* Is the amount of data packet received by destination nodes.
- *Energy consumption:* Is the total amount of energy consumed during simulation.

Figures 4 shows that LAAC-CS outperforms LAAC-Power in terms of the four metrics. This is due to fact that the c-neighborhood available bandwidth estimation in LAAC-CS is conservative, and hence a few number of flows are accepted. As LAAC-Power accepts more flows than LAAC-CS does, it has to generate more control routing packets to maintain routes, and hence it consumes more energy power. LAAC-Power sends some control and data packets using a high transmission power level and may interfere with more nodes than a message at the normal power level. In addition, due to node mobility, interference between two or more accepted flows can occur. This situation leads that network congestion in LAAC-Power occurs more frequently, and hence it incurs low throughput and low data packet delivery ratio than that in LAAC-CS.

6 Conclusion

In this paper, we have proposed an admission control method, which can be integrated with any reactive routing protocol. LAAC has the advantage that it does not need to carry information about the entire route like in CACP, PRP, RRT, and MACMAN. It can accurately estimate the contention count without incurring high message overhead, energy consumption, and extra delay. Simulation results have shown that LAAC-CS outperforms LAAC-Power in terms of data packet delivery ratio, throughput, message overhead, and energy consumption.

References

1. Cerveira, C.R., Costa, L.H.M.K.: A time-based admission control mechanism for ieee 802.11 ad hoc networks. In: 8th IFIP/IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2006), pp. 217–228 (August 2006)
2. de Renesse, R., Ghassemian, M., Friderikos, V., Aghvami, A.H.: Adaptive admission control for ad hoc and sensor networks providing quality of service. Technical report, Center for Telecommunications Research, King's College, London, UK (May 2005)
3. Dunbar, S.R.: The average distance between points in geometric figures. College Mathematics Journal 28(3), 187–197 (1997)

4. Jhonson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless. In: Imielinski, T., Korth, H. (eds.) Mobile Computing, ch. 5, pp. 153–181. Kluwer Academic Publishers, Dordrecht (1996)
5. Lindgren, A., Belding-Royer, E.M.: Multi-path admission control for mobile ad hoc networks. In: 2nd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2005), pp. 407–417 (2005)
6. Perkins, C.E., Royer, E.M.: Ad hoc on demand distance vector (aodv) algorithm. In: Systems and Applications (WMCSA 1999), pp. 90–100 (1999)
7. Sanzgiri, K., Chakeres, I.D., Belding-Royer, E.M.: Pre-reply probe and route request tail: approaches for calculation of intra-flow contention in multihop wireless networks. Mobile Networks and Applications 11(1), 21–35 (2006)
8. Yang, Y., Kravets, M.-R.: Contention-aware admission control for ad hoc networks. IEEE Transactions on Mobile Computing 4(4), 363–377 (2005)
9. Zeng, X., Bagrodia, R., Gerla, M.: Glomosim: A library for parallel simulation of large-scale wireless networks. In: Workshop on Parallel and Distributed Simulation, pp. 154–161 (1998)
10. Zhu, H., Chlamtac, I.: Admission control and bandwidth reservation in multi-hop ad hoc networks. Computer Networks 50(11), 1653–1674 (2006)

An Energy-Efficient Query Aggregation Scheme for Wireless Sensor Networks

Jun-Zhao Sun

Dept. Electrical & Information Engineering, University of Oulu, 90014 Finland
junzhao.sun@ee.oulu.fi

Abstract. This paper presents a novel method for optimizing sliding window based continuous queries. We deal with two categories of aggregation operations: stepwise aggregation (e.g. COUNT) and direct aggregation (e.g. MEDIAN). Our approach is, by using packet merging or compression techniques, to reduce the data size to the best extent, so that the total performance is optimal. A QoS weight item is specified together with a query, in which the importance of the four factors, power, delay, accuracy and error rate can be expressed. An optimal query plan can be obtained by studying all the factors simultaneously, leading to the minimum cost. Experiments are conducted to validate the effectiveness of the proposed method.

1 Introduction

Sensor nodes have very limited supply of energy, and should be available in function for extremely long time without being re-charged. Therefore, energy conservation needs to be one key consideration in the design of the system and applications. Extensive research work has been devoted to address the problem of energy conservation.

At high level, a sensor network can be modeled with a database view. Continuous query is commonly used for collecting periodical data from the objects under monitoring. This query needs to be carefully designed, in order to minimize the power consumption and maximize the lifetime. Data reduction techniques can be employed to decrease the size of data to be transferred in the network, and therefore save energy of sensor nodes.

This paper presents a method for the optimization of continuous query, and in particular, for the last stage of query processing: query result collection. The key novelty of the method lies on the careful consideration of QoS issue along with data gathering. By taking advantage of the QoS constraints on power, delay, accuracy, and error rate specified with a query, the method can find the optimal combination of transmitting sensor data to sink.

A query representation scheme is proposed, in which SQL based grammar is extended with sliding window, QoS constraint weight item, and sample clause. Then, a sample system model is created to model power consumption and time cost for both computation (data processing) and communication (data transmission). After that, a novel method is proposed for the optimization of continuous query with stepwise

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 413–424, 2008.
© Springer-Verlag Berlin Heidelberg 2008

aggregation (e.g. MAX, MIN, SUM, COUNT, AVERAGE, etc.). The method is described in detail including the determination of both sample rate and data integration. Similar method is presented for the optimization of continuous query with direct aggregation (e.g. MEDIAN). The similarity and variation of the previous method are discussed.

2 Query and System Models

A sensor field is like a database with dynamic, distributed, and unreliable data across geographically dispersed nodes from the environment [1, 2]. Sensor network applications use queries to retrieve data from the networks. Query processing is employed to retrieve sensor data from the network [3, 4].

SQL-based query language is commonly accepted in specifying queries for sensor networks as well. Below are two simple examples query in the form of extended SQL.

```
// Example query 1
SELECT     WINMEDIAN(S.temperature,10min,2min)
FROM       sensors AS S
WHERE      S.location=Area_C
WHILE      delay<10min AND accuracy>0.9 AND error<0.01
WEIGHT     (power,time,accuracy,error)=(0.4,0.1,0.3,0.2)
SAMPLE
    ON     Now + 5 min
    RATE   min 100

//Example query 2
SELECT     WINCOUNT(*,10min,2 min)
FROM         sensors AS S
WHERE      S.location=Area_C AND S.temperature>4C
WHILE      delay<10min AND accuracy>0.9 AND error<0.01
WEIGHT     (power,time,accuracy,error)=(0.4,0.1,0.3,0.2)
SAMPLE
    ON     00:00:00
    RATE   min 100
```

These are two queries to be performed upon streaming data by using a sliding window. Here, delay in WHILE clause is as the QoS constraint. The WEIGHT clause gives the weights of the factors in the quality-cost trade-off, by which the query plan can be optimized. In the two examples above, four factors are considered as the weight items, power consumption, report delay in time, and the accuracy and error rate of the result.

Data reduction is to decrease the size of data that is needed in the communication. Various data reduction techniques exist in this context. Packet merging is a simple data reduction technique, which combines multiple small packets into a big one, without considering the correlations between and the semantics within individual packets. Packet compression is to integrate one or multiple packets into a reduced packet, by employing suitable data compression algorithms. A number of compression algorithms have been studied for sensor networks [5-8]. Data aggregation is used in aggregate query to summarize a set of sensor into a single statistic, like MAX,

MIN, AVERAGE, MEDIAN, COUNT, etc. Data fusion refers to more complex operations above a set of readings and are usually used in multimedia data processing. The complexity of data aggregation and fusion leads to higher cost in terms of both energy and time.

This paper concentrates on the optimization of periodical aggregation queries during result collection. We mainly consider the situation where there are both delay and accuracy constraints, a weight item, and aggregation operations of average and count to be performed over collected data in the query. The paper is targeting to queries that have average/count aggregation operations. In periodical query most information in packet header (e.g. node ID, query ID, addresses, etc.) is the same across all the reading, and therefore can be shared. Also, it is reasonable to expect high spatial-temporal correlation between sample data collected in periodical query from single node, and therefore the data compression rate can be fairly high. Thus, we believe packet merging and compression techniques are suitable in this context.

Data reduction ratio, ru can be defined as $D'(u) = D(u)\ (1 - ru)$, where $D(u)$ is the data size before reduction, and $D'(u)$ is the size after, $D'(u) \leq D(u)$. Obviously, a higher ru is expected. The real value of ru is mostly depending on the merging/compression algorithms utilized as well as the similarity/correlation between data samples.

A query plan is executed with two components, computation and communication. Energy cost resulted from data processing, $E_P(u)$ denotes energy consumption for the data processing at the node u. Energy consumption for single data processing at a specific node u is fixed, and so can be represented by a constant $E_{SDP}(u)$. Energy cost for multiple data processing depends on the amount of data to be processed as well as the algorithms utilized. First the unit processing cost on node u is defined as $E_{PU}(u)$. Then, the cost for processing the $D(u)$ amount of data at node u is given by $E_P(u) = E_{SDP}(u) + E_{PU}(u)\ D(u)$. Here $D(u)$ is usually a set of sample result data for one single or different queries. Similarly, we can define the time for data processing, $T_P(u)$ as $T_P(u) = T_{SDP}(u) + T_{PU}(u)\ D(u)$, where $T_{SDP}(u)$ is the time for single data processing at node u and $T_{PU}(u)$ is the unit processing time at this node.

The E_{PU} and T_{PU} are relevant to data reduction ratio ru, depending on the processing algorithm used. Basically, the higher the ru, the higher the E_{PU}/T_{PU}. For example if a simple packet merge is performed, then ru will be very small, and the corresponding E_{PU} and T_{PU} will be very low. On the contrary, if some complex data compression algorithm is utilized, then a much better ru will be reached with fairly high E_{PU} and T_{PU}.

Transmission cost denotes the cost for transmitting $D(u)$ amount of data (i.e. packet header plus payload) from node u to node v through link $e = (u, v)$. The cost includes the energy consumption at both u and v. Unit cost of the link for transmitting data between two nodes can be abstracted as $E_U(e)$, and thus the transmission cost $E_T(e)$ is given by $E_T(e) = E_{TU}(e)\ D(u)$. The unit transmission cost on each edge, $E_{TU}(e)$, can be instantiated using the first order radio model presented in [9]. According to this model, the transmission cost for sending one bit from one node to another that is d distance away is given by $\beta\ d^{\gamma} + \varepsilon$ when $d < r_c$, where r_c is the maximal communication radius of a sensor, i.e. if and only if two sensor nodes are within r_c, there exists a communication link between them or an edge in graph G; γ and β are

tunable parameters based on the radio propagation, and ε denotes energy consumption per bit on the transmitter circuit and receiver circuit. Similarly, transmission time $T_T(e)$ is given by $T_T(e) = T_{TU}(e)\, D(u)$, where $T_{TU}(e)$ is the unit transmission time, i.e. the reciprocal of bandwidth, whose value is depended on the condition of the link. We note that all the cost and time parameters are all defined on link e, because different link has different conditions e.g. distances, congestion, and reliability.

The model above can be easily extended to the transmission cost and time for a *path*, which are the ones utilized in this paper. The cost and time for $D(u)$ from one node x to another node y through a multihop path x->y can be represented as $E(D(x): x\text{->}y) = \sum_{e \in x\text{->}y} E_T(e) + \sum_{u \in x\text{->}y} E_P(u)$ and $T(D(x): x\text{->}y) = \sum_{e \in x\text{->}y} T_T(e) + \sum_{u \in x\text{->}y} T_P(u)$.

3 Query with Stepwise Aggregation

We first study the problem of sliding window based continuous query on data stream with stepwise aggregation, i.e. the Example query 2 above in Section 2. Stepwise aggregation includes for example MAX, MIN, SUM, COUNT, AVERAGE. Without loosing any generality, in this section, we will take COUNT aggregation as an example.

To formalize the problem to be addressed, there are following assumptions.

1) Sliding window: this is a sliding window based continuous query, with window size (length) of Tw (10 min in the example)and sliding increment Ti (2 min in the example).
2) Constraints: there is a delay constraint d_{MAX} (maximum allowed delay, 1h in the example), an accuracy constraint 1-α_{MIN} (minimum confidence interval, 0.9 in the example), and an error constraint e_{MAX} (maximum packet error rate, 0.01 in the example) specified in the queries. A minimum sample rate is also specified as r_{MIN}.(in samples per hour, 100 in the example).
3) Weight: there is a weight item presented in the query denoting the tradeoff between power consumption and QoS of result report, as (power, time, accuracy, error) = (Wp, Wt, Wa, We).
4) MPS: in this paper, we also assume that there exist a constraint on the Max Packet Size (MPS) of the whole sensor network.

As shown in Figure 1, there are four steps of data processing at each single sensor node. First, a stream of readings are samples, next a count stream is generated according to the window size, then the node receives from other nodes their local results, and finally the node aggregates the results together and conduct data reduction by using techniques introduced in Section 2.

The criteria of choosing the best query plan lies on the satisfaction of query issuer to the best extent, by taking all the factors in the weight item (i.e. power, time delay, accuracy, and error probability) in to account. Moreover, there are two obvious constraints affecting the decision making. First, size of the integrated data packet should be less than MPS. Second, the delay constraint specified with the query should be obeyed.

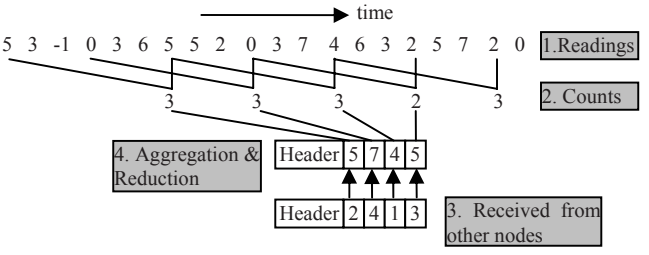


Fig. 1. Data processing in a node

Therefore, the problem under study can be formalized as two questions:
Question 1: how to find the best number of samples, n_s in one window size (i.e. sample frequency, see step 1 in Figure 1), and
Question 2: how to find the maximal number of sample data (can be 1), n_i to be integrated (step 4 in Figure 1),

so that trade-off denoted by the weight item leads to optimal result.

The answer of Question 1 has nothing to do with data transmission, but only data processing. This means only power consumption and accuracy need to be considered, without taking delay and error rate into account. Therefore, following node cost function can be defined as

$$C_1(n_s) = \frac{W_p}{W_p + W_a} \cdot E_P^{'}(n_s) + \frac{W_a}{W_p + W_a} \cdot A_R^{'}(n_s) \tag{1}$$

where Wp and Wa are the weights assigned to factors power and accuracy respectively, and $E_P^{'}(n_s)$ and $A_R^{'}(n_s)$ are normalized energy consumption of data processing and accuracy reciprocal respectively. The cost function denotes the total energy and accuracy costs of one specific window size, the less the better. Normalization of power consumption and accuracy reciprocal needs more study. First, since power consumption results only from data processing, the energy cost is given by $E_P(n_s) = n_s (E_S + E_{CU} D_{RU})$, where E_S is sample energy, E_{CU} is unit energy for *Count* operation, and D_{RU} is the size of one reading data. To perform the formalization, we need to find out the maximum possible number of sample in the sliding window, $n_{s\text{-}MAX}$. Obviously, we can simply assume that the maximum sample rate occurs in the case the node samples each time when the node wakes up. However in practice, the real maximum sample rate must be much lower than the case. This is because the lifetime goal of a sensor network is often explicitly defined. In [2] the maximum sample rate (in samples per hour) is estimated according to the remaining battery capacity of the node, the specified lifetime of the sensor network, and the energy to collect and transmit samples. Taking advantage of this estimation, the formalization of the energy cost is given by

$$E_P^{'}(n_s) = \frac{E_P(n_s)}{E_P(n_{s-MAX})}. \tag{2}$$

The accuracy of the result is represented by the reciprocal of the length of the confidence interval, and thus is relies on the estimation method. The Boolean result of whether an attribution is larger than a threshold (S.temperature > 4 C in the example) is a random variable whose probability distribution is (0 – 1), i.e. *P (ValueOfAttribute > Threshold) = p*. Query is used to estimate this probability, *p* by $p = Count / n_s$. As to (0 – 1) distribution, its expectation is *p* and variance is *p(1-p)*. According to Central Limit Theorem, when number of samples is big, the distribution of $\frac{Count - n_s p}{\sqrt{n_s p(1-p)}}$ is approximately *N* (0, 1). Therefore, the accuracy of the estimation with n_s samples is given by $A(n_s) = \frac{\sqrt{n_s}}{2\sigma \cdot z_{\alpha/2}}$, where σ is the variance and $z_{\alpha/2}$ is the α quantile of standard normal distribution. The minimum accuracy can be easily obtained by $A(n_{s-MIN}) = \frac{\sqrt{n_{s-MIN}}}{2\sigma \cdot z_{\alpha_{MIN}/2}}$, where $n_{s\text{-}MIN}$ is given by $n_{s\text{-}MIN} = r_{MIN}\, Tw$.

The normalized accuracy reciprocal $A_R^{'}(n_s)$ is given by

$$A_R^{'}(n_s) = \frac{A(n_{s-MIN})}{A(n_s)}. \tag{3}$$

Note that even the variance is unknown, the normalization can still be performed.

By using formula (2) and (3) to (1), we can obtain $n_s^{'} = \arg\min_{ns}(C_1(n_s))$. Thus, the answer of Question 1 is given by

$$n_s = (n_{s\text{-}MIN} < n_s^{'} < n_{s\text{-}MAX},\ n_s^{'},\ (n_s^{'} > n_{s\text{-}MAX}: n_{s\text{-}MAX}, n_{s\text{-}MIN})). \tag{4}$$

The answer of Question 2 concerns both local data processing (step 2 to 4 in Figure 1) and data transmission. Suppose a data integration technique is to be utilized above a stream of results (local counts or counts received from other nodes), the algorithm focuses on finding the maximal number of samples, n_i that optimizes trade-off items of the query. According to the MPS, delay, and error constraints, we have the following three arguments. For any selected node *u*, first, $n_{i1}(u) = \arg\max_{ni}(D'(u) <$ MPS), where size of data after data reduction $D'(u) = D(u)(1\text{-}ru) = (\text{Header_Size} + n_i D_{CU})(1\text{-}ru)$, with D_{CU} the size of one count result (most probably attached with a sequence number of time stamp). Second, $n_{i2}(u) = \arg\max_{ni}(d_W(u) < d_{MAX})$, where $d_W(u) = (n_i - 1)\,Ti + T(D'(u): u\text{->}s)$ denotes the time of waiting and sending the integrated packet. Third, $n_{i3}(u) = \arg\max_{ni}(e_p(n_i) < e_{MAX})$, where $e_p(n_i) = 1 - (1 - e_b(u))^{D'(u)}$ is the packet error rate and e_b is the bit error rate of the link from local node to its parent node. The e_b is affected by both the data transmission rate and the signal power margin, and can be obtained from empirical estimation.

Finally, a global cost function, C_2 can be defined as

$$C_2(n_i) = \frac{W_p}{W_{p+d+e}} \cdot E_{P+T}^{'}(n_i) + \frac{W_d}{W_{p+d+e}} \cdot d_{P+T}^{'}(n_i) + \frac{W_e}{W_{p+d+e}} \cdot e_p^{'}(n_i) \tag{5}$$

where $Wp+d+e= Wp+Wd+We$, and E', d' and e' are formalized energy consumption, time delay and packet error rate respectively. The cost function denotes the total costs of one specific query plan, the less the better. Normalization of power consumption, time delay and error rate can be performed as follows.

$$E'_{P+T}(n_i) = \sum_{\forall \text{selected } u} \frac{E(D'(u):u->s)}{n_i \cdot E((D_{CU} + Header_Size):u->s)} \tag{6}$$

$$d'_{P+T}(n_i) = \frac{\underset{\forall \text{selected } u}{MAX}(dw - T((D_{CU} + Header_Size):u->s))}{d_{MAX}} \tag{7}$$

$$e'_p(n_i) = \frac{\underset{\forall \text{related } u}{MAX}(1-(1-e_b(u))^{D'(u)}}{e_{MAX}} \tag{8}$$

Here the energy consumption and time delay concerns both data processing (as depicted in Figure 1) and data transmission during the entire path from local node to sink. According to the cost function in formular (5), we have $n_{i4} = \arg\min_{ni} C_2(n_i)$.

And finally, the number of samples is given by

$$n_i = \min(n_{i1}, n_{i2}, n_{i3}, n_{i4}) \tag{9}$$

4 Query with Direct Aggregation

Direct aggregation is different from stepwise aggregation in the sense that, direct aggregation cannot be performed until all the aggregation data is available. In other words, it cannot be executed upon partial data. In this section, without loosing any generality, we simple take MEDIAN as an example. Figure 2 illustrated the basic idea and key processes of this sort of query.

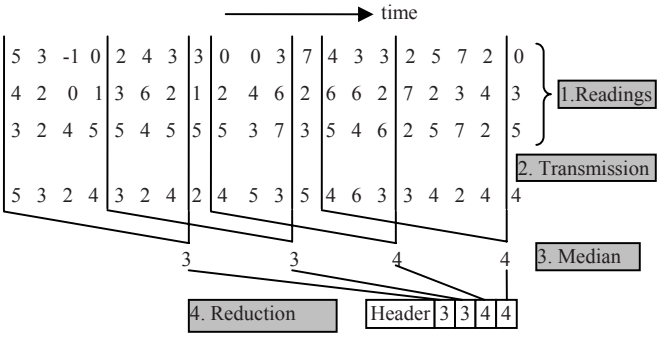


Fig. 2. Data transmission and processing

The most crucial difference the sliding window based continuous MEDIAN query (query example 1) with the one in Section 3 (COUNT query over stream data, query

example 2) is that in query 2 COUNT is a monotonic and summary aggregate which means its value can only get larger as more values are aggregated, while on the other hand allows partial aggregation with other count values (as step 4 in Figure 1). Instead as for query 1, MEDIAN is an exemplary aggregate computing some property over the entire set of values, and therefore does not allow partial aggregation. In other words, the aggregation cannot be performed until all the concerned data is available. This means in query 1 there will be more data transmission.

The query is executed with two steps. First, all the readings are sent to a node who is the common ancestor of all the concerned nodes for MEDIAN aggregation. Next, a stream of median aggregation results is sent to the sink. In the second step, the problem is simple and the problem of data reduction (i.e. find out the best n_i) can be directly solved with the method introduced in Section 3, by simply assuming n_s=1. Therefore without losing any generality, here we assume that the common ancestor node is exactly the sink node.

As to query 2 in the previous section, the answers of the two questions (i.e. to find out n_s and n_i) are independent. While in query 1 the key is still to find the answers for the same questions, they are actually correlated due to the time reason and therefore should be jointly considered. Also, in answering Question 1, the main difference of query 1 with query 2 is that in query 2, we need to simply concentrate on one local node, to decide the window size for it which is actually common to all the rest concerned sensor nodes. Instead, in query 1 we have to first find out the total population of samples for all the selected sensor nodes (hereafter suppose the number is m) according to the WHILE clause (S.location = Area_C as the condition in the example above), and after that uniformly assign to each node a number of samples for one window size.

After the discussions above, a global cost function for query 1 can be defined as

$$C(n_s, n_i) = W_p E'_{P+T}(n_i) + W_d d'_{P+T}(n_s, n_i) + W_a A'_R(n_s) + W_e e'_p(n_i) \qquad (5')$$

All the formula (6) to (9) in previous section are applicable here for query 1, unless the Ti for getting d_W should now be replaced by sample interval Tw/n_s.

Another difference is, instead of (0, 1) distribution as in query 2, here for MEDIAN aggregation, the distribution of the concerned random variable is usually assumed as either normal distribution or uniform distribution. To normal distribution the derivation is the same as previous section. As to uniform distribution, by using the same approximating method of Central Limit Theorem, method in previous section is still applicable.

5 Experiments

In this section, we validate the effectiveness of the proposed algorithm. We assume that all the sensor nodes are homogeneous. Table 1 shows the system and query parameters values used in the performance analysis. We assume an average $T_{TU}(e)$ is available and thus we can define a number of hops (noh) to represent the distance

from the end node to sink (10 in this paper). We study three data reduction scenarios. The first one is based on packet merging (PM), in which the *ru* can be derived by

$$ru = 1 - \frac{D'}{D} = 1 - \frac{Sph + n_i \cdot D_{RU}}{n_i \cdot (Sph + D_{RU})}. \quad (10)$$

where *Sph* is header size (15 bytes in this paper). The second scenario is to employ packet compression (ru1) with *ru* = 0.3 (ru2), and for the third *ru* = 0.8 (ru3). We study the example query 2 with two weight item settings: (*Wp*, *Wt, Wa*, *We*) = (0.6, 0.1, 0.1, 0.2) and (0.1, 0.1, 0.5, 0.3). The target sensor network is simplified so that there is only one selected node which is of 10 hops to sink.

Table 1. Parameters values used in performance analysis

System parameters			
parameter	value	parameter	value
β	100 pJ/bit/m^2	*Es*	10 nJ/bit
γ	2	*ru*	PM, 0.3, 0.8
ε	90 nJ/bit	Header Size	15 bytes
d	10 m	D_{RU}	2 bytes
E_{PU}	20 nJ/bit	MPS	2k bytes
T_{PU}	0 ns/bit	*noh*	10
E_{TU}	100 nJ/bit	$n_{s\text{-}MAX}$.	1 / ms
T_{TU}	0.02 ms/bit		
Query parameters			
parameter	value	parameter	value
d_{MAX}	1 h	*Tw*	10 min
$1\text{-}\alpha_{MIN}$	0.9	*Ti*	2 min
e_{MAX}	0.01	r_{MIN}	100
e_r	10 E-5		
(*Wp*, *Wt, Wa*, *We*)		w1: (0.7, 0.0, 0.1, 0.2) w2: (0.1, 0.0, 0.6, 0.3)	

Two experiments are designed to valid the methods for answering Questing 1 and 2 respectively. Figure 3 shows the result of experiment 1. The figure clearly demonstrates the effect of the weight item. In case of w1, power consumption is deemed more important than accuracy (0.7 vs. 0.1), therefore a relatively small *ns* (204) is obtained than the case of w2 (*ns*=2455) in which accuracy is emphasized more than energy (0.6 vs. 0.1).

Figure 4 – 8 illustrate the results of experiment 2, in which the number of data integration *ni* is being found. In Figure 4, *ni1* is found via data size. Obviously, *ru* and MPS play key roles for this calculation. In Figure 5, *ni2* is found via waiting time. In our setting the maximum delay allowed is large, and so the time for data processing and transmission is tiny. The resulted number thus is depended mostly on sliding increment vs. delay. This is why the three scenarios of *ru*1-3 all return the same result. Figure 6 depicts the result of finding *ni3* via packet error rate. Again *D'(u)* is the key factor influencing the results.

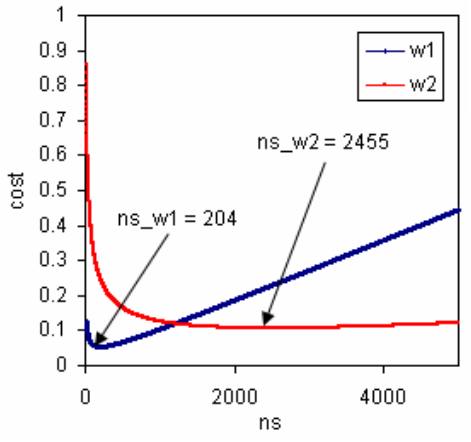


Fig. 3. Finding sample rate *ns*

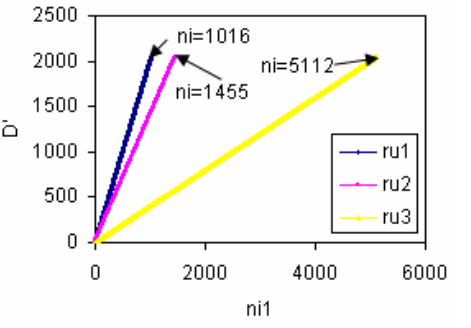


Fig. 4. Finding integration number *ni1*

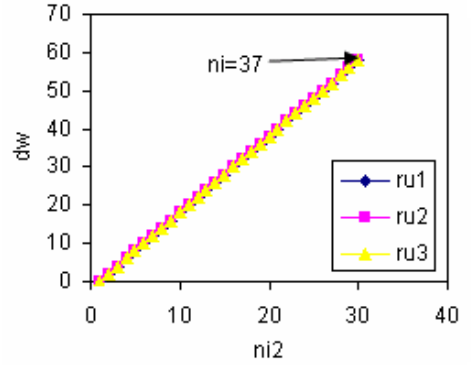


Fig. 5. Finding integration number *ni2*

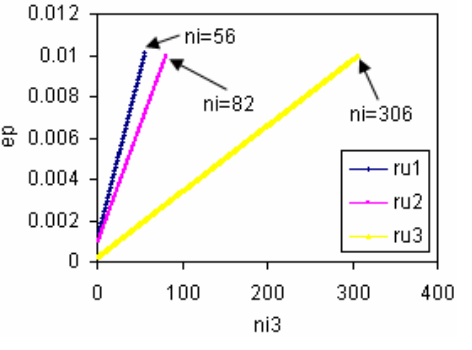


Fig. 6. Finding integration number *ni3*

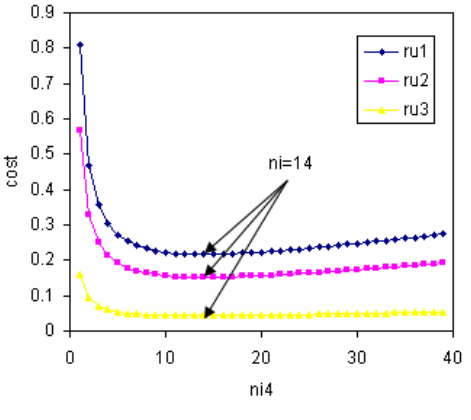


Fig. 7. Finding number *ni4* for w1 case

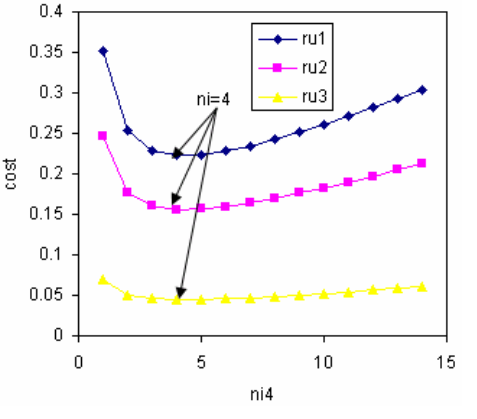


Fig. 8. Finding number *ni4* for w2 case

Figure 7 and 8 are more interesting, illustrating the trade-offs between error rate and power consumption. The effect of time delay is neglected because the setting is large (1 hour) and therefore the item in the weight is set to 0 for both w1 and w2. When n is small, the cost is relatively high. Then cost decreases with increasing n. This is the benefit gaining from energy saving due to data reduction. From some point of n, the cost starts to increase again. This is because the increase in packet size leads to the increase of packet error rate. This effect is much clearer in case of w2, if comparing the two figures. This is due to the fact that w2 considers error rate more than power consumption. Finally, it is easy to understand that large *ru* always performs better.

6 Conclusions

A novel method is proposed to optimize the execution of periodical queries with COUNT and AVERAGE aggregations, by jointly considering four QoS factors including energy consumption, time delay, result accuracy and packet error rate. Algorithm is described in detail. Experiments are conducted to validate the method. Results show that the proposed method can achieve the goal of query optimization. Future work includes to study the effectiveness of adaptive sampling rate, smart sampling (not with fixed interval), and sample dropping schemes.

Acknowledgment. Financial support by Academy of Finland (Project No.: 209570) is gratefully acknowledged.

References

1. Bonnet, P., Gehrke, J.E., Seshadri, P.: Towards Sensor Database Systems. In: Proceedings of the Second International Conference on Mobile Data Management. Hong Kong (January 2001)
2. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: An Acquisitional Query Processing System for Sensor Networks. ACM Transactions on Database Systems 30(1), 122–173 (2005)
3. Madden, S., Franklin, M.J., Hellerstien, J.M., Hong, W.: The design of an acquisitional query processor for sensor networks. In: Proceedings ACM SIGMOD, San Diego, CA, USA, June 2003, pp. 491–502 (2003)
4. Gehrke, J., Madden, S.: Query processing in sensor networks. IEEE Pervasive Computing 3(11), 46–55 (2004)
5. Ju, H., Cui, L.: EasiPC: A Packet Compression Mechanism for Embedded WSN. In: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005), pp. 394–399 (2005)
6. Kimura, N., Latifi, S.: A survey on data compression in wireless sensor networks. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005), vol. 2, pp. 8–13 (2005)

7. Xu, R., Li, Z., Wang, C., Ni, P.: Impact of data compression on energy consumption of wireless-networked handheld devices. In: Proc. 23rn International Conference on Distributed Computing Systems (ICDCS 2003), May 2003, pp. 302–311 (2003)
8. Chen, M., Fowler, M.L.: Data compression trade-offs in sensor networks. In: Schmalz, M.S. (ed.) Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications. Proceedings of the SPIE, vol. 5561, pp. 96–107 (October 2004)
9. Heinzelman, W.R., Chandrakasan, A., Blakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. 33rd Ann. Hawaii Int'l Conf. System Sciences (January 2000)

Novel Algorithms for the Network Lifetime Problem in Wireless Settings

Michael Elkin[1,⋆], Yuval Lando[2], Zeev Nutov[3], Michael Segal[2,⋆⋆], and Hanan Shpungin[1,⋆⋆⋆]

[1] Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
[2] Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
[3] Computer Science Division, The Open University of Israel, Raanana 43107, Israel

Abstract. A wireless ad-hoc network is a collection of transceivers positioned in the plane. Each transceiver is equipped with a limited, non-replenishable battery charge. The battery charge is then reduced after each transmission, depending on the transmission distance. One of the major problems in wireless network design is to route network traffic efficiently so as to maximize the *network lifetime*, i.e., the number of successful transmissions. This problem is known to be NP-Hard for a variety of network operations. In this paper we are interested in two fundamental types of transmissions, broadcast and data gathering.

We provide polynomial time approximation algorithms, with guaranteed performance bounds, for the maximum lifetime problem under two communication models, omnidirectional and unidirectional antennas. We also consider an extended variant of the maximum lifetime problem, which simultaneously satisfies additional constraints, such as bounded hop-diameter and degree of the routing tree, and minimizing the total energy used in a single transmission.

1 Introduction

Wireless ad-hoc networks gained much appreciation in recent years due to massive use in a large variety of domains, from life threatening situations, such as battlefield or rescue operations, to more civil applications, like environmental data gathering for forecast prediction. The network is composed of numerous transceivers (nodes) located in the plane, communicating by radio. A transmission between two nodes is possible if the receiver is within the transmission range of the transmitter. The underlying physical topology of the network is dependent on the distribution of the wireless nodes (location) as well as the transmission power (range) assignment of each node. Since the nodes have only a limited, non-replenishable initial power charge (battery), energy efficiency becomes a crucial factor in wireless networks design.

⋆ Supported by the Israeli Academy of Science, grant 483/06.
⋆⋆ Supported by REMON (4G networking) consortium.
⋆⋆⋆ Supported in part by the Lynn and William Frankel Center for Computer Science.

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 425–438, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The transmission range r_v of node v is determined by the power assigned to that node, denoted by $p(v)$. It is customary to assume that the minimal transmission power required to transmit to distance d is d^α, where the *distance-power gradient* α is usually taken to be in the interval $[2, 4]$ (see [1]). Thus, node v receives transmissions from u if $p(u) \geq d(u, v)^\alpha$, where $d(u, v)$ is the Euclidean distance between u and v. There are two possible models: symmetric and asymmetric. In the symmetric model, also referred to as the undirected model, there is an undirected communication link between two nodes $u, v \in T$, if $p(u) \geq d(u, v)^\alpha$ and $p(v) \geq d(v, u)^\alpha$, that is if u and v can reach each other. The asymmetric variant allows directed (one way) communication links between two nodes. Krumke et al. [2] argued that the asymmetric version is harder than the symmetric one. This paper addresses the asymmetric model.

Ramanathan and Hain [3] initiated the formal study of controlling the network topology by adjusting the transmission range of the nodes. Intuitively, an increase to the transmission range assignment allows more distant nodes to receive transmissions. But at the same time, it causes a quicker battery exhaustion, which results in a shorter network lifetime. We are interested in maximizing the network lifetime under two basic transmission protocols, data broadcasting and data gathering. **Data broadcasting**, or in short broadcast, is a network task when a source node s wishes to transmit a message to all the other nodes in the network. **Data gathering** - a less popular, nevertheless important network task, is also known as convergecast. Opposite to broadcast, there is a destination node d, and all the other nodes wish to transmit a message to it. We consider data gathering *with aggregation*.

Each node v, has an initial battery charge $b(v)$. The battery charge decreases with each transmission. The network lifetime is the time from network initialization to the first node failure due to battery depletion. It is possible to look at two formulations of the maximum network lifetime problem. In the *discrete* version, node v can transmit at most $\lfloor b(v)/d^\alpha \rfloor$ times to distance d. Whereas, the *fractional* variant states that a transmission from node v to distance d is valid for b/d^α time units. For example, for $b(v) = 15$, $d = 2$, and $\alpha = 2$, the discrete version of the problem would allow $\lfloor 15/4 \rfloor = 3$ *separate* transmissions, while the fractional formulation determines that node v can have a valid transmission for $15/4 = 3.75$ time units. Most of the current research addresses the fractional formulation. The discrete version was introduced by Sahni and Park [4]. They provided a number of heuristics without guaranteed performance bounds. This paper studies the discrete version, which seems to be more problematic.

An additional consideration in wireless networks design, is the type of the antenna used for communication. In this paper we consider two types of communication antennas, *omnidirectional* and *unidirectional*. For a node $u \in \mathcal{V}$ equiped with an omnidirectional antenna, a single message transmission to the most distant node in a set of nodes X is sufficient so that all the nodes in X receive the message. While, if u uses a unidirectional antenna, then it has to transmit to each of the nodes in X separately.

The paper is organized as follows. In the rest of the section, we introduce our model, discuss previous work and outline our contribution. In Sections 2 and 3 we present our results for the unidirectional and omnidirecitonal antenna types, respectively.

1.1 The Model

Graph Preliminaries. Here we provide some graph theory related definitions used in this paper.

- For any graph H, let $V(H)$ and $E(H)$ be the node and edge sets of H, respectively.
- In a directed graph H, let $\delta_H(v)$ be the set of outgoing edges from v in $V(H)$.
- For a weighted graph H, with a weight function w, we alternately use the notation $w(e)$ and $w(u,v)$, to specify the weight of edge $e=(u,v) \in E(H)$. The weight of H is given by $C(H) = \sum_{e \in E(H)} w(e)$.
- The weight function w of graph H is said to be uniform, if $\forall e \in E(H),\ w(e)= w_0$, for some non-negative value w_0.
- The cube of graph H, denoted H^3, contains an edge (u,v) if there is a path from u to v in H with at most 3 edges.
- A Hamiltonian circuit $h = (u_1, u_2, \ldots, u_{n+1} = u_1)$ in graph H, where $u_i \in V(H)$ for $1 \leq i \leq n$, is a graph cycle that visits each node in $V(G)$ exactly once and also returns to the starting node. The weight of h is given by $C(h) = \sum_{i=1}^{n} w(u_i, u_{i+1})$, where w is the weight function of H.
- Given an undirected graph H, let $MST(H)$ be a minimum spanning tree of H.

Network Model. We have n wireless nodes $\mathcal{V}$ positioned in a Euclidean plane. The wireless network is then modeled by a complete, weighted, and undirected graph $G_{\mathcal{V}}$ with a weight function $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$, $w(u,v) = d(u,v)^{\alpha}$. It is easy to verify that the weight function obeys the weak triangle inequality with coefficient $2^{\alpha-1}$, i.e., for any $u, v, w \in \mathcal{V}$, $w(u,w) \leq 2^{\alpha-1}(w(u,v) + w(v,w))$.

Both types of messages, broadcast or convergecast, are propagated by using a directed spanning tree of $G_{\mathcal{V}}$, called a *transmission tree*. A broadcast message, originating in $s \in \mathcal{V}$, is propagated by an arborescence T_s rooted at s, also called a *broadcast tree*. In the case of a convergecast to $d \in \mathcal{V}$, the messages from all nodes are propagated by a reversed arborescence T_d rooted at d, also called a *convergecast tree*. In the case of a broadcast message, a node may be required to transmit to multiple recipients (its children in the broadcast tree), while a convergecast message is transmitted once to the parent in the convergecast tree.[1]

[1] We consider data gathering with aggregation, which means that each node v combines the messages sent by the nodes in a subtree rooted at v into one message, and then propagates it to its parent.

Every node $v \in \mathcal{V}$ has an initial battery charge $b(v)$. After each message propagation, its residual energy decreases. The energy decrease depends on the recipient nodes location, as well as the antenna type used, either omnidirectional or unidirectional. Formally, the power consumption of $v \in \mathcal{V}$ due to a transmission tree T is,

$$\beta_T(v) = \begin{cases} \max\limits_{e \in \delta_T(v)} w(e), & \text{omnidirectional,} \\ \sum\limits_{e \in \delta_T(v)} w(e), & \text{unidirectional.} \end{cases}$$

Note that the reverse of a broadcast tree is a convergecast tree. Due to this symmetry property, and in an attempt to keep the definitions simple, from this point, we refer to the broadcast transmission protocol only. Although there is symmetry in definitions, nevertheless not all the results work well for both cases. We provide explicit statements whenever the results are relevant for convergecast as well. In this paper we assume $\alpha = 2$ for simplicity, though our results can be easily extended to any constant value of α.

Problems Definition. The general maximum lifetime broadcast (MLB) problem is defined as follows. **The input** to the MLB problem is graph $G_{\mathcal{V}}$, initial battery charges $b : \mathcal{V} \to \mathbb{R}$, and a sequence of m source nodes $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$, where $s_i \in \mathcal{V}$, for $1 \leq i \leq m$. Each of the source nodes has one broadcast message to transmit to all the other nodes. **The output** is a sequence of broadcast trees $\mathcal{T}_B = \{T_1, T_2, \ldots, T_k\}$, where T_i is rooted at s_i, for $1 \leq i \leq m$, so that for all $v \in \mathcal{V}$, $\sum_{i=1}^{k} \beta_{T_i}(v) \leq b(v)$. **Our objective** is to maximize k. Intuitively, given a sequence of source nodes, we wish to maximize the number of successful broadcast message propagations, while satisfying the battery constraint. That is, all the nodes have enough battery charge to support message propagation in a sequence of broadcast trees.

There are two possible relaxations of the general maximum lifetime broadcast problem. **The first relaxation** is to set $s_i = s$, for all $s_i \in \mathcal{S}$, that is one source node s generates all broadcast messages. **The second relaxation** is to require that all the broadcast trees would be an orientation of one undirected tree. In this paper we consider the following three problems.

Problem 1. [Single Source Maximum Lifetime Broadcast (SSMLB)]
Input: Graph $G_{\mathcal{V}}$, initial battery charges $b : \mathcal{V} \to \mathbb{R}$, and a source node $s \in \mathcal{V}$.
Output: A sequence of broadcast trees $\mathcal{T}_B = \{T_1, T_2, \ldots, T_k\}$, so that T_i is rooted at s, and for all $v \in \mathcal{V}$, $\sum_{i=1}^{k} \beta_{T_i}(v) \leq b(v)$.
Objective: Maximize k.

Problem 2. [Single Source/Topology Maximum Lifetime Broadcast (SSTMLB)]
Input: Graph $G_{\mathcal{V}}$, initial battery charges $b : \mathcal{V} \to \mathbb{R}$, and a source node $s \in \mathcal{V}$.
Output: A directed spanning tree T of $G_{\mathcal{V}}$ rooted at s, and an integer k, $1 \leq k \leq m$, so that for all $v \in \mathcal{V}$, $k\beta_T(v) \leq b(v)$.
Objective: Maximize k.

Problem 3. [Single Topology Maximum Lifetime Broadcast (STMLB)]
Input: Graph $G_{\mathcal{V}}$, initial battery charges $b : \mathcal{V} \to \mathbb{R}$, and a sequence of m source nodes $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$, where $s_i \in \mathcal{V}$.
Output: An undirected spanning tree T of $G_{\mathcal{V}}$ and an integer k, $1 \leq k \leq m$, so that for all $v \in \mathcal{V}$, $\sum_{i=1}^{k} \beta_{T_i}(v) \leq b(v)$, where T_i, $1 \leq i \leq k$, is a broadcast tree rooted at s_i, and is obtained by orienting the edges of T.
Objective: Maximize k.

The analogous problems for convergecast, SSMLC, SSTMLC, and STMLC are defined in a similar way.

1.2 Previous Work

Numerous studies were conducted in the area of maximizing the network lifetime under various transmission protocols. In addition to broadcast and convergecast, it is common to find references to multicast and unicast[2] as well. Different formulations of the maximum lifetime problem are due to the single/multiple source/topology relaxations. These relaxations, mixed together with the antenna type, have impact on the complexity of the problem.

As mentioned previously, to the best of our knowledge, there is no reference to the discrete version of the maximum lifetime problem, except for [4]. Instead, we survey the state of current results for the fractional case, grouped in accordance to the communication model used.

Omnidirectional Model. Orda and Yassour [5] gave polynomial-time algorithms for broadcast, multicast and unicast in the case of **single source/single topology**, which improved previous results by [6]. Segal [7] improved the running time of the MLB problem for the broadcast protocol and also showed an optimal polynomial-time algorithm for convergecast with aggregation. Additional results may be found in [6,8]. By allowing **single source/multiple topology**, the broadcast and multicast become NP-Hard [5], while convergecast and unicast have polynomial-time optimal solutions. In [5], the authors establish an $O(\log n)$ and $O(k^{\varepsilon})$ approximation algorithms for broadcast and multicast, respectively, where k is the size of the multicast destination set and ε is any positive constant. The same paper shows an optimal solution for the unicast case by using linear programming and max-flow algorithms. Liang and Liu [9] prove that the convergecast problem without aggregation is NP-Complete for general costs. An easier version, with aggregation, does have a polynomial solution [10] in $O(n^{15} \log n)$ time. To counter the slowness of the algorithm, Stanford and Tongngam [11] proposed a $(1 - \varepsilon)$-approximation in $O(n^3 \frac{1}{\varepsilon} \log_{1+\varepsilon} n)$ time based on Garg and Könemann [12] algorithm for packing linear programs. They also propose several heuristics and evaluate their performance by simulation. Generally, a common approach to solving the fractional problem is to use

[2] *Multicast* is a more general case of broadcast. A source node is required to transmit to a set of nodes; *unicast* is more specific, a source node is required to transmit to a single node.

Table 1. Current results for the fractional case

Single Source - Omnidirectional Model		
Topology	*Broadcast*	*Convergecast (with agg.)*
Single	OPT [5,6,7]	OPT [7]
Multiple	$6(1-\varepsilon)$ approx. (follows from [11] and [16])	OPT [10]
Single Source - Unidirectional Model		
Topology	*Broadcast*	*Convergecast (with agg.)*
Single	NP-Hard [5]	OPT [7]
Multiple	OPT [5]	OPT [10]

various LP formulations that reduce the problem to one of finding the maximum multicommodity flow in a network. See also [13,14,15].

Unidirectional Model. The authors in [5] show that for broadcast, the problem is NP-Hard in the case of **single source/single topology** and has a polynomial solution in the case of **single source/multiple topology**. They also show that it is NP-Hard in both of these cases for multicast. To the best of our knowledge, this is the only paper to address the unidirectional communication model. Note that for convergecast there is no difference between the two models (omnidirectional and unidirectional), as the node is required to transmit to its parent in the convergecast tree only. Therefore, the results from [7] and [10] hold.

A summary of the results for the fractional case under the omnidirectional model is given in Table 1 (OPT represents that the problem can be solved optimally). The result for single source/multiple topology in case of broadcast is derived from the simple fact that when the Garg-Könemann $(1-\varepsilon)$-approximation algorithm uses λ-approximation minimum length columns it produces a $\lambda(1-\varepsilon)$ approximation to the packing LP defined by [11] if used for broadcasting. We can choose a 6-approximation by Ambühl [16] as the λ-approximation algorithm for the minimum energy broadcast problem. The 6-approximation can be improved by using the result in [17].

1.3 Our Contribution

We study the discrete version of the maximum lifetime problem under broadcast/convergecast transmissions. We provide polynomial time approximation algorithms, with guaranteed performance bounds, for the maximum lifetime problem under two communication models, omnidirectional and unidirectional antennas. We also consider an extended variant of the maximum lifetime problem, which simultaneously satisfies additional constraints. In particular, our main contributions are:

1. Under the unidirectional model, we state the NP-Hardness of the SSMLB and SSTMLB problems. We provide an $O(\log n)$-approximation to the SSTMLB problem. Then, for the SSMLB problem we find a sequence of broadcast trees of optimal length k^*, so that the battery constraint is violated by at

Table 2. Our contribution in the discrete case

Single Source - Unidirectional Model		
Topology	*Approx.*	*Remarks*
Single	$O(\log n)$	
Multiple	1	battery violation by $O(\log(nk^*))$, k^* is OPT
Multiple Source - Omnidirectional Model		
Topology	*Approx.*	*Remarks*
Single	2	with additional bi-criteria
Multiple	$O(\rho^2)$	with $n/\rho + \log \rho$ hop-diameter, and additional bi-criteria

most $O(\log(nk^*))$ times. That is, the energy consumed by node v is at most $O(\log(nk^*))b(v)$.

2. Under the omnidirectional model, we develop two approximation algorithms for the STMLB problem. We assume uniform initial battery charges and present a 2-approximation algorithm by using the $MST(G)$ as the broadcast tree. This immediately yields constant bounds for the total energy consumed in a single transmission and the maximum degree. We then construct a broadcast tree which is a $O(\rho^2)$-approximation to the problem. In addition, it has a bounded hop-diameter $n/\rho + \log \rho$, where $1 \leq \rho \leq n$, a constant maximum degree, and the energy consumed in a single transmission is at most ρ times the optimum for a broadcast transmission. We argue that the tradeoff between the maximum lifetime and the hop-diameter is optimal. That is, our multi-criteria approximation is tight.
3. Finally, we show that the results for the STMLB problem, can be applied for the STMLC problem as well.

To the best of our knowledge, these are the first theoretic results for the discrete formulation of the problem. Our results are summarized in Table 2.

2 Unidirectional Communication Model

The unidirectional model implies that each node is charged for every outgoing edge in the transmission tree. The power consumption of $v \in \mathcal{V}$ due to a single message transmission, in a directed tree T, is $\beta_T(v) = \sum_{e \in \delta_T(v)} w(e)$.

In this section we consider two variants of the MLB problem under the single source relaxation. First the more general case is addressed, where multiple topologies are allowed, which is the SSMLB problem. Then, we show that by doing slight modifications to the proposed algorithms, we establish a similar result in the case of single topology relaxation, namely the SSTMLB problem. We slightly modify the original problems, by allowing a violation of the battery constraint by γ. That is, we require that the energy consumption of every $v \in \mathcal{V}$ is at most $\gamma b(v)$.

Assuming P≠NP, both the single and the multiple topology cases cannot achieve a $1/\gamma$-approximation algorithm for any constant $\gamma > 0$, since deciding whether even one tranmission is possible is equivalent to the so called **Degree Constrained Arborescence** problem. This implicates that the SSMLB and SSTMLB problems are NP-Hard (take $\gamma = 1$).

Note that in the single topology case, k transmissions with initial battery charges $\{\gamma b(v) : v \in \mathcal{V}\}$ imply $\lfloor k/\gamma \rfloor$ transmissions for initial battery charges $\{b(v) : v \in \mathcal{V}\}$. Indeed, since we are using the same arborescence, the power consumption of every node in every message propagation is identical and there are k message propagations, then for the original charges $\{b(v) : v \in \mathcal{V}\}$ the number of propagations is at least $\lfloor b(v)/(\gamma b(v)/k) \rfloor = \lfloor k/\gamma \rfloor$. Unfortunately, for the multiple topology case, we do not have a method to convert the battery violation to a standard approximation.

Although the input to the SSMLB problem, is a weighted, undirected graph $G_{\mathcal{V}}$, we can alternatively look at the directed version $G'_{\mathcal{V}}$, i.e., for every edge $e = (u, v) \in E(G_{\mathcal{V}})$, create the instances $(u, v), (v, u)$ in $E(G'_{\mathcal{V}})$. The weight of the directional edge is the same as of the original one. In the rest of the section we prove the next theorem, which summarizes our main results for the unidirectional model.

Theorem 1. *Given a weighted, directed graph $G'_{\mathcal{V}}$ and a source node $s \in \mathcal{V}$, let k_1^* and k_2^* be the number of successful message propagations in the optimal solutions of the SSTMLB and SSMLB problems, respectively. Then, (i) there exists a broadcast tree T rooted at s, so that for all $v \in \mathcal{V}$, $(k_1^*/\log n)\beta_T(v) \leq b(v)$; (ii) there exists a sequence of broadcast trees $\mathcal{T}_B = \{T_1, T_2, \ldots, T_{k_2^*}\}$, each rooted at s, and for all $v \in \mathcal{V}$, $\sum_{i=1}^{k_2^*} \beta_{T_i}(v) \leq (\log(nk_2^*))b(v)$.*

2.1 Weight Scaling Reduction

We start by showing a simple scaling of weights, which allows us to manipulate the input graph $G'_{\mathcal{V}}$. If for some node $v \in \mathcal{V}$ and constant $c > 0$, we set $b(v) \leftarrow b(v)/c$ and for every outgoing edge $e \in \delta_{G'_{\mathcal{V}}(v)}$, set $w(e) \leftarrow w(e)/c$, we obtain a similar instance to our problem. Note that an instance with uniform weights[3] is easily transformed into an instance with *unit* weights (all weights being 1), by applying the weight scaling reduction described above.

2.2 The SSMLB Problem

We start with the multiple topology case of the MLB problem under the single source relaxation and prove part (ii) of Theorem 1.

A directed graph H is *k-edge-outconnected from s* if it contains k-edge disjoint paths from s to any other node. By Edmond's Theorem [18], a graph is k-edge-outconnected from s if, and only if, it contains k edge-disjoint spanning arborescences rooted at s. Let us introduce the following decision problem.

[3] Though graph $G'_{\mathcal{V}}$ does not necessarily has uniform weights, nevertheless we use this scaling in future developments.

Problem 4 (Bound Constrained k-Outconnected Subgraph (BCkOS)).
Input: A directed graph G with a weight function w, bounds $b : V(G) \to \mathbb{R}$, a source node $s \in V(G)$, and a positive integer k.
Question: Does G have a k-edge-outconnected spanning subgraph H, so that for all $v \in V(G)$, $\beta_H(v) \leq b(v)$.

Given a positive integer k, the problem of finding a sequence of broadcast trees of length k in $G'_{\mathcal{V}}$ can be reduced to the BCkOS problem as follows. As an edge in $E(G'_{\mathcal{V}})$ may be used several times, we add $k-1$ copies of each edge to the graph.[4] Call this graph $G^k_{\mathcal{V}}$. Then we solve the BCkOS problem for $G^k_{\mathcal{V}}$.

To solve the SSMLB problem, we need to search for the maximum value of k, for which the BCkOS returns a positive answer given $G^k_{\mathcal{V}}$. This can be done by a simple binary search in the range $\{1, \ldots, K\}$, where $K = \max_{e \in \delta_{G_{\mathcal{V}}}(s)} b(s)/w(e)$. The upper bound is due to the source node battery constraint. The BCkOS problem is NP-hard even for uniform weights and $k = 1$. We therefore consider the optimization problem that seeks to minimize the factor of the weight-degree bounds violation.

Problem 5 (Weighted-Degree Constrained k-Outconnected Subgraph (WDCk OS)).
Input: A directed graph G with a weight function w, bounds $b : V(G) \to \mathbb{R}$, a source node $s \in V(G)$, and a positive integer k. Graph G has a k-edge-outconnected spanning subgraph H^* satisfying, for all $v \in V(G)$, $\beta_{H^*}(v) \leq b(v)$.
Output: Find a k-edge-outconnected spanning subgraph H of G, so that for all $v \in V(G)$, $\beta_H(v) \leq \gamma \cdot b(v)$.
Objective: Minimize γ.

Clearly, guaranteeing a factor of γ for the WDCkOS problem also guarantees a γ violation in our case. Let the Degree Constrained k-Outconnected Subgraph (DCkOS) problem be the restriction of WDCkOS problem to instances with unit (or uniform) weights; in this case the bounds $b(v)$ are just the degree constraints, and thus assumed to be integral. The following statement follows from Theorems 1 and 4 in [19] ($d_H(v)$ is the outdegree of v in H).

Theorem 2 ([19]). *There exists a polynomial time algorithm that given an instance of* DCkOS *finds a k-edge-outconnected spanning subgraph H of G so that $d_H(v) \leq b(v) + 2$ if $k = 1$ and $d_H(v) \leq b(v) + 4$ if $k \geq 2$.*

It is easy to verify that DCkOS admits a 3-approximation algorithm for $k = 1$ and a 5-approximation algorithm for $k \geq 2$. For every node v with $b(v) = 0$, remove from G the edges leaving v, and then compute a k-edge-outconnected from s spanning subgraph H of G using the algorithm as in Theorem 2. Then

[4] Instead of adding $k-1$ copies of an edge, we may assign to every edge capacity k, and consider the corresponding "capacited" problems; this will give a polynomial algorithm, rather than a pseudo-polynomial one. For simplicity of exposition, we will present the algorithm in terms of multigraphs, but it can be easily adjusted to the terms of capacitated graphs.

$d_H(v) = 0$ for every $v \in V(G)$ with $b(v) = 0$. For every $v \in V$ with $b(v) \geq 1$ we have $d_H(v) \leq b(v) + 2 \leq 3b(v)$ if $k = 1$, and $d_H(v) \leq b(v) + 4 \leq 5b(v)$ if $k \geq 2$.

The following lemma (proof is omitted due to lack of space), in conjunction with the $O(1)$-approximation to DCkOS, proves part (ii) of Theorem 1.

Lemma 1. *An α-approximation algorithm for the DCkOS problem implies an $\alpha \cdot O(\log(kn))$-approximation algorithm for the WDCkOS problem.*

2.3 The SSTMLB Problem

The single topology case of the MLB under the single source relaxation is to find a spanning arborescence T of $G_{\mathcal{V}}$ rooted at s, so that the number of transmissions is maximized under the battery constraints. The problem can be reduced, similar to the multiple topology case, to that of finding a 1-edge-outconnected from s (namely, an arborescence rooted at s) spanning subgraph H of G, satisfying the constraints $k \cdot \beta_H(v) \leq b(v)$ for all $v \in \mathcal{V}$. By setting $B(v) \leftarrow b(v)/k$, we obtain the weighted-degree constraints $\beta_H(v) \leq B(v)$. This defines an instance of the WDCkOS problem with $k = 1$. Thus, we can compute in polynomial time a 1-outconnected from s spanning subgraph H of G so that for every $v \in V(G)$ we have $\beta_H(v) \leq \gamma \cdot B(v) = b(v)/k$, namely, $k \cdot \beta_H(v) \leq \gamma \cdot b(v)$. This means that we can guarantee k transmissions using H with battery capacities $\gamma \cdot b(v)$. Consequently, we can guarantee $\lfloor k/\gamma \rfloor$ transmissions with the original battery capacities $b(v)$, which proves part (i) of Theorem 1.

3 Omnidirectional Communication Model

In this section we consider the omnidirectional model. This model defines that the transmission of some node $v \in \mathcal{V}$ is received by *all* the nodes within the transmission range of v. Therefore, the power consumption of node $v \in \mathcal{V}$ due to a single message transmission, in a directed tree T, is $\beta_T(v) = \max_{e \in \delta_T(v)} w(e)$. We assume uniform initial battery charges, that is for all $v \in \mathcal{V}$, $b(v) = B$. Without loss of generality we may assume $B = 1$.

Recall the STMLB problem. We look for a spanning tree T of $G_{\mathcal{V}}$, so that the number of broadcast messages routed by using its orientations is maximized. We call T the broadcast backbone. In this section we show two different constructions of T, each satisfying additional multi-criteria constraints. In the end, we state that T can be used for convergecast (the STMLC problem) as well.

We are given a weighted, undirected graph $G_{\mathcal{V}}$, and a sequence $\mathcal{S}$ of m source nodes. Let $\langle T^*, k^* \rangle$ be an optimal solution for the SSMLB problem. We start by deriving an upper bound on k^*.

Lemma 2. *Let $e^* = (u, v)$ be the longest edge in T^*. Then $k^* \leq 2/w(e^*)$.*

Proof. Let T_i, $1 \leq i \leq k^*$, be a broadcast tree rooted at s_i, and obtained by orienting the edges of T^*. Note that either u transmits to v ($(u, v) \in E(T_i)$) or v transmits to u ($(v, u) \in E(T_i)$), but not both. Out of the k^* broadcast trees, let

k_u be the number of trees in which u transmits to v. Without loss of generality, let $k_u \geq k^*/2$ (otherwise we take v). Since e^* is the longest edge in T^*, we can lower bound the total power consumption of u, $\sum_{i=1}^{k^*} \beta_{T_i}(u) \geq k_u w(e^*) \geq w(e^*)k^*/2$. Due to the power consumption constraint, $\sum_{i=1}^{k^*} \beta_{T_i}(u) \leq B = 1$. As a result, $k^* \leq 2/w(e^*)$. □

3.1 Multi-criteria Broadcast Backbone

In this section we show that if we take T to be $MST(G_{\mathcal{V}})$, then we obtain a 2-approximation algorithm for the STMLB problem, as well as additional multi-criteria.

Lemma 3. *Let k be the maximum value, so that for all $v \in \mathcal{V}$, $\sum_{i=1}^{k} \beta_{T_i}(v) \leq b(v)$, where T_i, $1 \leq i \leq k$, is a broadcast tree rooted at s_i, and is obtained by orienting the edges of $MST(G_{\mathcal{V}})$. Then $k \geq k^*/2$.*

Proof. Let $e' = (u', v')$ be the longest edge in $MST(G_{\mathcal{V}})$. Since the longest edge in any minimum spanning tree is not greater than the longest edge of any spanning tree, $w(e') \leq w(e^*)$. Clearly, nodes u', v' have the largest possible power consumption $w(e')$ in any broadcast tree T_i, $1 \leq i \leq k$. Therefore, $k > 1/w(e')$. From Lemma 2, $k^* \leq 2/w(e^*)$. We conclude $k \geq k^*/2$. □

Note that using $MST(G_{\mathcal{V}})$ as the broadcast backbone, also provides some additional valuable multi-criteria guarantees, as concluded in the next theorem.

Theorem 3. *Given a weighted, undirected graph $G_{\mathcal{V}}$, and a sequence of m source nodes $\mathcal{S}$. Setting $T = MST(G_{\mathcal{V}})$; (i) provides us with k successful broadcast message propagations, where $k \geq k^*/2$; (ii) T has a bounded degree of 6; (iii) the total energy consumption in one broadcast tree is at most c times of the optimum, where $6 \leq c \leq 12$.*

Proof. (i) From Lemma 3, $k \geq k^*/2$; (ii) the maximum degree of $MST(G_{\mathcal{V}})$ is at most 6, since the minimum spanning tree of $G_{\mathcal{V}}$ is identical to the Euclidean minimum spanning tree on the node set $\mathcal{V}$, and the latter has a bounded degree of 6; (iii) in [20] the authors prove that for any node set in the plane, the total energy required by broadcasting from any node is at least $\frac{1}{c} \sum_{e \in E(MST(G_{\mathcal{V}}))} w(e)$, where $6 \leq c \leq 12$. Therefore the total energy consumption in one broadcast tree is of a constant factor from the best possible.

3.2 Bounded Hop-Diameter Multi-criteria Broadcast Backbone

Our construction is based on a Hamiltonian circuit. Sekanina [21] showed that the cube of any tree, with at least 3 vertices, is Hamiltonian. Andrea and Bandelt [22] give a linear time algorithm for the construction of the Hamiltonian circuit in T^3, given T. They also show that the weight of the Hamiltonian circuit is at most $(\frac{3}{2}\tau^2 + \frac{1}{2}\tau)$ times the weight of the tree, where τ is the weak triangle inequality parameter (under our assumption that $\alpha = 2$, $\tau = 2^{\alpha-1} = 2$). Moreover, it can be shown that the longest edge in the Hamiltonian circuit is at most $O(1)$ times the longest edge in T. The following theorem applies the above to $MST(G_{\mathcal{V}})$.

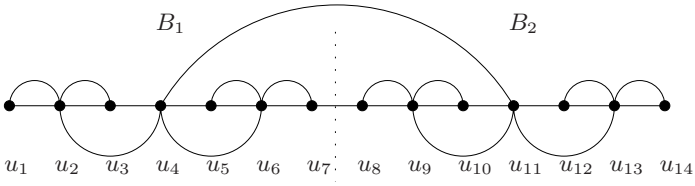


Fig. 1. Bounded hop-diameter broadcast backbone for $h = (u_1, u_2, \ldots, u_{14})$ and $\rho = 7$. There are $14/2 = 7$ node sequences $U_1 = \{u_1, u_2, \ldots, u_7\}$ and $U_2 = \{u_8, u_2, \ldots, u_{14}\}$. The center nodes of U_1 and U_2 are u_4 and u_{11}, respectively. Each of the trees B_1, B_2 spans the corresponding nodes in U_1 and U_2, respectively.

Theorem 4 ([22]). *Let $h = (u_1, u_2, \ldots, u_{n+1} = u_1)$, where $u_i \in \mathcal{V}$ for $1 \leq i \leq n$, be the Hamiltonian circuit as a result of applying the construction in [22] on $MST(G_{\mathcal{V}})$. Define e^*_{MST} and e^*_h to be the longest edges in $MST(G_{\mathcal{V}})$ and h, respectively. Then $C(h) = O(C(MST(G_{\mathcal{V}}))$ and $w(e^*_h) = O(w(e^*_{MST}))$.*

Next, we describe the construction of the broadcast backbone T_h, based on the Hamiltonian circuit $h = (u_1, u_2, \ldots, u_{n+1} = u_1)$ from Theorem 4. Let ρ be an integer parameter, $1 \leq \rho \leq n$. The node set of T_h is $\mathcal{V}$. We divide the sequence of nodes $U_h = \{u_1, u_2, \ldots, u_n\}$ into n/ρ consecutive sequences U_i with ρ nodes each, so that $U_i = \{u_{\rho(i-1)+1}, u_{\rho(i-1)+2}, \ldots, u_{\rho i}\}$, $1 \leq i \leq n/\rho$.

The *center node* of a sequence $U = \{x_1, x_2, \ldots, x_j\}$, denoted $c(U)$, is the median node with an index $\lfloor \frac{j+1}{2} \rfloor$. There are two types of edges in T_h, $E(T_h) = E_1 \cup E_2$. The first type of edges connects the center nodes of every two adjacent node sequences, $E_1 = \{(c(U_i), c(U_{i+1}))\}_{i=1}^{n/\rho-1}$. The second type of edges, E_2, induces n/ρ complete binary trees $B_1, \ldots, B_{n/\rho}$. Each tree B_i, $1 \leq i \leq n/\rho$ spans the nodes in U_i and is rooted at $c(U_i)$. The tree B_i is constructed recursively. The children of $c(U_i)$ are the center nodes in subsequences $U_i^1 = \{v_{\rho(i-1)+1}, \ldots, v_{\rho(i-1)+\frac{\rho-1}{2}}\}$ and $U_i^2 = \{v_{\rho(i-1)+\frac{\rho+3}{2}}, \ldots, v_{\rho i}\}$. We then continue to construct a complete binary tree in each of the subsequences, U_i^1, U_i^2, in a similar way. Note that each tree B_i has $\log \rho$ levels (see example in Figure 1).

Denote by $e^*_{T_h}$ and e^*_h the longest edges in T_h and h, respectively. The next lemma shows some valuable bounds for T_h (the proof is omitted due to lack of space).

Lemma 4. *The graph T_h is a spanning tree of $G_{\mathcal{V}}$ and has a bounded hop-diameter of $O(n/\rho + \log \rho)$, a bounded degree of 4, and it holds $C(T_h) = O(\rho C(h))$ and $w(e^*_{T_h}) = O(\rho^2 w(e^*_h))$.*

Note that the tradeoff between the approximation of the longest edge and the hop-diameter bound presented in Lemma 4 is optimal. Consider the unweighted n-path: any tree of hop-diameter at most D for it, contains an edge with an interval length of at least $(n-1)/D$, and so its squared length is at least $(n-1)^2/D^2$. Since the longest edge of the n-path has a squared length of 1, we get an increase of the longest edge by a factor of at least $\Omega(n^2/D^2)$. Finally, substitute $D = n/\rho$ to obtain $\Omega(\rho^2)$.

Similar to the first construction, the broadcast backbone T_h satisfies multiple constraints according to Lemma 4. We can therefore derive the next theorem.

Theorem 5. *Given a weighted, undirected graph $G_{\mathcal{V}}$, and a sequence of m source nodes $\mathcal{S}$. Setting $T = T_h$; (i) provides us with k successful broadcast message propagations, where $k \geq k^*/2\rho^2$; (ii) T has a bounded hop-diameter of $n/\rho + \log \rho$; (iii) T has a bounded degree of 4; (iv) the total energy consumption in one broadcast tree is at most $O(\rho)$ times of the optimum.*

Proof. Conditions (ii) and (iii) are immediate from Lemma 4. From the same lemma in conjunction with Theorem 4, $w(e^*_{T_h}) = O(\rho^2 w(e^*_{MST}))$. By following similar arguments as in the proof of Lemma 3, we obtain (i). Combining Theorem 4 and Lemma 4 also yields the bound $C(T_h) = O(\rho C(MST(G_{\mathcal{V}})))$. Following the same arguments as in Theorem 3 condition (iv) follows. □

3.3 Applicability to the STMLC Problem

The two constructions for the broadcast backbone may be used for convergecast, which will result in similar asymptotic bounds. The similarity follows from Lemma 2, which can be applied for convergecast transmissions, since it does not rely on any broadcast specific characteristics. This results in the same approximation ratios for the network lifetime (number of successful message propagations). The hop-diameter and degree bounds follow immediately from the constructions. Finally, we have to show that the total power consumption bound also holds. In [23], the authors showed that the total power consumption needed for one convergecast propagation is at least $C(MST)$.

References

1. Pahlavan, K., Levesque, A.H.: Wireless information networks. Wiley-Interscience, Chichester (1995)
2. Krumke, S.O., Liu, R., Lloyd, E.L., Marathe, M.V., Ramanathan, R., Ravi, S.S.: Topology control problems under symmetric and asymmetric power thresholds. In: Pierre, S., Barbeau, M., Kranakis, E. (eds.) ADHOC-NOW 2003. LNCS, vol. 2865, pp. 187–198. Springer, Heidelberg (2003)
3. Ramanathan, R., Hain, R.: Topology control of multihop wireless networks using transmit power adjustment. In: INFOCOM 2000, pp. 404–413 (2000)
4. Park, J., Sahni, S.: Maximum lifetime broadcasting in wireless networks. IEEE Transactions on Computers 54(9), 1081–1090 (2005)
5. Orda, A., Yassour, B.-A.: Maximum-lifetime routing algorithms for networks with omnidirectional and directional antennas. In: MobiHoc 2005, pp. 426–437 (2005)
6. Kang, I., Poovendran, R.: Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. Mobile Networks and Applications 10(6), 879–896 (2005)
7. Segal, M.: Fast algorithm for multicast and data gathering in wireless networks. Information Processing Letters (2007)
8. Adamou, M., Sarkar, S.: A framework for optimal battery management for wireless nodes (2002)

9. Liang, W., Liu, Y.: Online data gathering for maximizing network lifetime in sensor networks. IEEE Transactions on Mobile Computing 6(1), 2–11 (2007)
10. Kalpakis, K., Dasgupta, K., Namjoshi, P.: Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. Computer Networks Journal 42(6), 697–716 (2003)
11. Stanford, J., Tongngam, S.: Approximation algorithm for maximum lifetime in wireless sensor networks with data aggregation. In: SNPD 2006, pp. 273–277 (2006)
12. Garg, N., Könemann, J.: Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In: FOCS 1998, pp. 300–309 (1998)
13. Chang, J.-H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: INFOCOM 2000, pp. 22–31 (2000)
14. Calinescu, G., Kapoor, S., Olshevsky, A., Zelikovsky, A.: Network lifetime and power assignment in ad hoc wireless networks. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 114–126. Springer, Heidelberg (2003)
15. Xue, Y., Cui, Y., Nahrstedt, K.: Maximizing lifetime for data aggregation in wireless sensor networks. Mobile Networks and Applications 10(6), 853–864 (2005)
16. Ambühl, C.: An optimal bound for the mst algorithm to compute energy efficient broadcast trees in wireless networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1139–1150. Springer, Heidelberg (2005)
17. Caragiannis, I., Flammini, M., Moscardelli, L.: An exponential improvement on the mst heuristic for minimum energy broadcasting in ad hoc wireless networks. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 447–458. Springer, Heidelberg (2007)
18. Edmonds, J.: Matroid intersection. Annals of discrete Mathematics 4, 185–204 (1979)
19. Bansal, N., Khandekar, R., Nagarajan, V.: Additive gurantees for degree bounded directed network design. IBM Research Report RC24347 (2008) (to appear in STOC 2008)
20. Wan, P.-J., Calinescu, G., Li, X., Frieder, O.: Minimum-energy broadcast routing in static ad hoc wireless networks. In: INFOCOM 2001, pp. 1162–1171 (2001)
21. Sekanina, M.: On the ordering of the set of vertices of a connected graph. Publication of the Faculty of Sciences of the University of Brno 412, 137–142 (1960)
22. Andreae, T., Bandelt, H.J.: Performance guarantees for approximation algorithms depending on parametrized triangle inequalities. SIAM J. Discret. Math. 8(1), 1–16 (1995)
23. Shpungin, H., Segal, M.: Low energy construction of fault tolerant topologies in wireless networks. In: DIALM-POMC 2007 (2007)

Message Quality for Ambient System Security

Ciarán Bryce

INRIA-Rennes, France
Ciaran.Bryce@inria.fr

Abstract. In ambient systems, a principal may be a physical object whose identity does not convey useful information for taking security decisions. Thus, establishing a trusted channel with a device depends more on the device being able to demonstrate what is does, rather than who it is. This paper proposes a security model that allows a principal to establish the intent of an adversary and to make the adversary prove its trustworthiness by furnishing proof of current and past behavior.

1 Introduction

The technological combination of portable devices (e.g., smart phones, PDAs), wireless networking and processor cards embedded in everyday devices has led to the emergence of ambient computing systems. Ambient systems are employed for process control and person-centric applications like mobile health [7], domotics [11] and payment [8]. Security is a major concern for these systems, especially with personal devices containing private information and increasingly sensitive applications. The major security risks faced include device theft, virus infections and spam.

Ambient systems pose a challenge for information security enforcement. Communicating devices can be unknown to each other, and since the network might use short-range radio, it might not possess a trusted third party that can act as a certificate authority [14] or reputation server [16] that facilitates the establishment of trusted channels between devices. Solutions for security need to be scalable. This means that when two peers want to establish a trusted channel, then there should be sufficient information on their devices to establish the channel, and thus minimize reliance on third parties.

The goal of this paper is to examine the security requirements for ambient systems and to propose a security model. This model is entitled *message quality* since its role is to examine each message that a device receives from a partner device, and to determine if that message is consequent to a security attack on, or by, the partner device. The model's implementation leverages the support of the *Trusted Platform Module* (TPM) [17] – a general purpose hardware chip designed for secure computing that is used by a platform to demonstrate that its software has not been tampered with.

A key feature of the security model is the deprecated role of principal identity: it can be more important for a device to prove *what it does* than it is to prove *who it is*. For instance, when a user PDA interacts with a soda vending machine,

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 439–450, 2008.
© Springer-Verlag Berlin Heidelberg 2008

it is more useful for the PDA to establish that the machine returns sodas in return for payment than it is to know the vending machine's serial number. This approach requires that a security model validates the software running on the device, whereas traditional security models are designed to validate the identity of a principal. A second aim the model is to enable principals to estimate the trustworthiness of a partner principal, by being able to determine if their partner has behaved in a trustworthy way in the past.

This paper is organized as follows. Section 2 presents the security challenges of ambient computing systems that the paper addresses. The security model is presented in Section 3, and we comment on its strengths and weaknesses. In Section 4, the model is integrated into a programming model that is often used in ambient computing – the tuple space model [5]. An example of the security model in use is presented. Related work is presented in Section 5 which concludes the paper.

2 Security Challenges

Viruses remain one of the most common and most costly source of security attacks. Handheld devices are not immune to viruses, the Cabir virus for instance being the first major virus on the Symbian operating system and the Duts virus infecting PocketPCs[1]. A mobile device virus is potentially more pernicious than an Internet virus since a device can be engaged in a communication without the owner being aware. (One can always disconnect a wired communication link and thus be sure that the computer is not engaged in hidden communication).

A second major security concern in today's information systems is information misuse [4]. This is the problem of information being exposed or destroyed through ineffective access controls to physical and information resources. A concrete example for handhelds is theft: the French interior ministry reports that up to 200 000 mobile phones are stolen in France each year[2].

A further risk for ambient systems is spam – a well known problem for the Internet, with up to 70% of e-mail traffic consisting of unsolicited messages [9]. Spam is attractive for attackers (spamers) due to the low cost of sending messages. A similar situation can arise in ambient systems, where devices can send information to others at no cost – apart from the energy consumed by the device's battery. An example spam scenario is one where a user passes near a supermarket and picks up unwanted messages from items on sale.

We choose in this paper to concentrate on the above risks since they are relatively novel. There are of course other risks, like network blocking attacks, as well as traditional risks such as cracking the trusted hardware on devices and PIN theft.

There are a number of challenges to implementing a security infrastructure:

– There is a potentially huge number of peers without centralized management. No peer knows all others, and most peers know few others. Strictly speaking,

[1] http://www.virusthreatcenter.com/
[2] http://www.afom.fr/v3/TEMPLATES/acces_elus_l2.php?rubrique_ID=115

peer Alice *knows* Bob if she can link Bob's identity to his expected behavior. This lack of knowledge would normally imply the use of trusted third parties such as recommendation servers and certificate authorities. However, given the potential size of systems and weak connectivity of wireless networks, these solutions must be minimized in favor of decentralized scalable ones. In a decentralized solution, when Alice sends a message M to Bob, then the peers and message M contain sufficient data for Bob to verify that it is safe to act upon the message.

- Personal computing devices need to minimize physical resources like energy, and thus optimize network communications and security processing, as well as memory space.
- Each device owner is autonomous and has complete control over his device. He can install any software on his device and access any service. He can manipulate information on the device in any way, unless that data is stored on a trusted zone, possibly with the support of trusted hardware. A device may be stolen and then used or misused by a non-owner. It is crucial that the manipulations made by a user to his device be bounded to prevent spoofing attacks. These are a high risk in ambient systems if users are able to generate (false) identities.
- Systems in practice degrade over time through ware of hardware and software (as patches are applied for upgrades and virus/bug fixes). Thus, a hitherto trusted principal can start behaving badly, so a security infrastructure must be able to detect this.

3 Message Quality Security

3.1 Approach

There are two cornerstones to our security model. The first is that a principal must prove *what it does* rather than *who it is* (i.e., its identity). A principal's identity might not change, but its ability to service a request – what it does – can change as new functionality is added, viruses spread, etc. The second cornerstone is decentralization. That is, as suggested in Figure 1, the security framework on a principal's device decides on the security of each message. If the message is acceptable, then it is passed up to higher application layers for processing. Otherwise, the message gets rejected without the application being informed. For decentralization, there must be sufficient information in the message and pre-distributed to Bob to take this decision. This is necessary to allow a decision to be taken without real-time recourse to a (perhaps absent from network) trusted third party.

Consider a message M sent from Alice to Bob, c.f., Figure 1. There are three properties of M that define its security:

1. M **is plausible**. This is the property that M is a message that Alice is likely to utter, given the behavioral profile of Principal Alice. For instance, if Alice is a frequent taxi passenger, then the messages she is likely to utter include requests for a taxi.

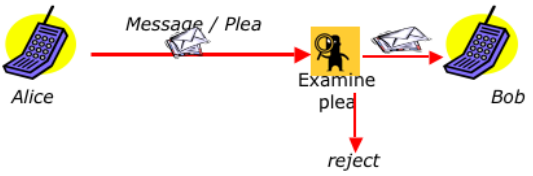


Fig. 1. Message Quality

2. M **is trustworthy**. This is the property that permits Bob to believe that the contents of message M are true. For example, if Alice sends a message asking for a taxi, then Bob – the taxi driver – can stop the taxi with sufficient confidence that there is a passenger waiting to mount. Trustworthiness is not an absolute value of a message. Rather, it is a feature whereby Alice can complement a message with proof that the contents of the message are trustworthy. This aspect leverages work done in trust-based frameworks, e.g., [16,18]. In Figure 1, the proof is represented in the *plea* object.
3. M **is useful**. This is the property that Bob is interested in M. The message is rejected as spam by Bob's device if it does not correspond to the class of messages that Bob wishes to receive.

These properties are collectively known as *message quality* and, as is later argued, are crucial to addressing the risks outlined. Note how the onus is on the message sender to convince the receiver of the quality of its message.

3.2 Management of Message Quality

The elements needed to implement message quality are **profiles**, **policies** and **pleas**. All are first-class objects in the message quality model.

A profile defines the expected behavior of a principal. A policy is an element of the receiver and is evaluated whenever it receives a message. It determines if a message is trustworthy, plausible and useful from the receiver's point of view. A plea is a copy of the sender's profile information and history information that a sender includes in a message in order to argue for quality. Both profile and policy objects are declarative; they are downloaded with an application and installed on a device. Installation is PIN-protected.

Principals. The role of identity for principals is deprecated in the message quality model. While most principals do not need to furnish or even possess an identity, there is a small population of identities that are well-known. For instance, software providers (e.g., application and OS providers), service providers (e.g., a taxi company where passengers use their PDAs to gain access to the service) can be considered well-known since all passengers must interact with them at some stage, e.g., to download the software or renew their subscription. Another class of well-known principals are hardware providers.

The role of the well-known principals is to define programs and policies for devices since many users might not be in a position to define these for themselves.

Profiles. Principal behavior is defined in a profile. A profile is qualified by the set of installed programs. A program, in turn, is qualified by the following information:

- The *actions* of the program. These are expressed as the set of messages that the program sends and receives.
- A certification of the program origin that describes where or by whom the program was developed. (The createdBy certificate).
- A certification of the program installation describing who installed the program on the device. (The installedBy certificate).
- Any other application or service specific certifications that are considered useful in an application context, e.g., inspectedBy.

The certificates in the profile are termed *profile certificates*. The role of a certificate is to bind a public key to a profile, which are keys belonging to well-known principals. We conjecture that the number of certificates is nonetheless minimized since they certify behavior, and not identity. For instance, there can be millions of taxi clients for a single taxi client behavioral profile. Further, since the number of well-known principals is small, certificates can be pre-distributed, e.g., during software installation.

Policies. When Bob receives a message, its security kernel rejects the message if it does not conform to the profile of Alice. Further, for utility, Bob can specify a *policy* on the acceptable profile of the sender. A sender whose message does not conform with this policy has its message automatically rejected. The two elements of the policy are **i)** the *evidence* – a message history that the sender must demonstrate for trustworthiness, and **ii)** the required profile certificates – specified as the role (i.e., createdBy, installedBy, etc.) and identity of well-known principals. Like programs, policies can be downloaded and installed on the device (since ordinary PDA users are not expected to understand the intricate details).

Platform. The structure of a ambient device platform using the message quality model is illustrated in Figure 2. The OS or runtime environment runs alongside

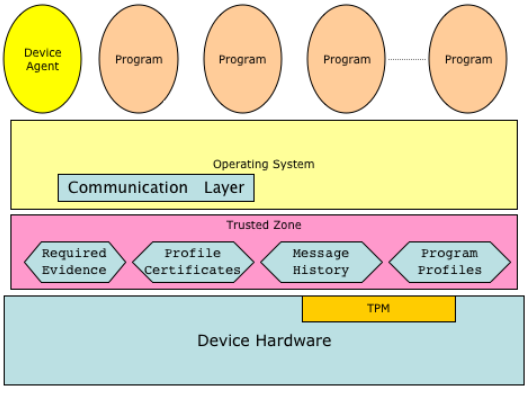


Fig. 2. Ambient device environment

a software layer called the *trusted zone*. This zone stores profiles, policies and history data. The profile and policy can only be set by furnishing a PIN that supposedly, only the owner of the platform knows. One of the programs running on the device is a device *agent*; it is a trusted program that interacts with well-known principals for downloading programs and profiles.

Verifying Message Quality. A profile is used to construct a plea for each message sent. The other component of a plea is a *history* of messages sent and received by a principal. A plea sent along with a message M from Alice to Bob is used in the following way by Bob's trusted zone to verify message quality.

- Plausibility is verified by i): ensuring the M belongs to Alice's profile; ii) validating any profile certificates in the profile.
- Trustworthiness is verified by ensuring that Alice's history of messages matches a series of messages that Bob specifies as required evidence.
- Utility is verified by ensuring that M belongs to Bob's profile.

4 Model Implementation

4.1 Prototype

The programming model chosen in this paper is based on the Linda tuple space model [5]. Our implementation is based on the Lana system [2], each principal has its own tuple space in which it publishes its tuples. The tuples in a principal's tuple space can be read by all other principals in its network vicinity; see Figure 3. Many systems designed for ambient environments employ Linda's tuple space model for the reasons cited above, e.g., Lime [12], Spread [3].

The tuple space `rd` operation returns a tuple matching the pattern argument from the tuple space of any device in the neighborhood of the device issuing the request. The `out` operation publishes a tuple in the space of the current principal. The tuple space primitives do not contain explicit reference to the message quality model, so we consider its implementation in the programming model as transparent.

A key requirement for the message quality model is that a trusted zone be present on each device participating in the model. One approach to building trusted zones is to use software protection techniques, now possible using strongly typed languages like Java [6]. Another approach is to rely on the Trusted

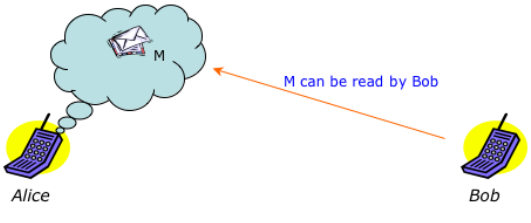


Fig. 3. Each device has its own tuple space

Platform Module (TPM) to enable a platform to demonstrate that the software it is running has not been tampered with. A TPM is a hardware device whose functionality is specified by the Trusted Computing Group [17]. The TPM is becoming commodity hardware, with 200 million TPM-enabled PCs having been shipped by the end of 2007.

A TPM is used to store measures of the software in the form of secure code and data hashes internally in its Platform Configuration Registers (PCRs). To verify that a device is running correct (non-modified) software, its TPM can be challenged to produce its stored PCR values signed using an Attestation Identity Key (AiK). This is created by a TPM and certified by a well-known principal (or privacy authority). The privacy authority that certifies the AiKs can be the application software provider. On contacting the provider, the client device downloads the correct digest values. As illustrated in Figure 4, when Alice sends a message m to Bob, this is in fact in reply to a tuple space query from Bob and a challenge to Alice's TPM. Bob can verify Alice's PCRs (and software) using his copy of the digest that he got when downloading the software. Trust is thus built in a transitive fashion: the digest permits Bob to believe that a correct version of the trusted zone is running on Alice's device, and then trust in Alice's trusted zone provider is sufficient for Bob to believe that Alice is correctly implementing the message quality model.

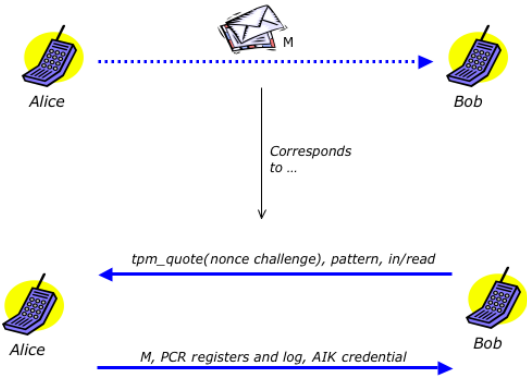


Fig. 4. Message quality in tuple space model

4.2 Example

The example illustrates a shuttle service that people can call using their PDAs. There are two classes of service: the fast service enables clients to call taxis from any location, the second requires them to go to a shuttle depot where they take the shuttle. The principal interactions are illustrated in Figure 5. The first three messages, for calling a shuttle, are only for fast shuttles; the final two message exchanges occur in all shuttles at the end of the trip. To call a shuttle, a customer sends a message –"Please" – with the desired destination. The shuttle replies with a fare quote, which the customer can accept by sending a "Stop"

Fig. 5. Shuttle scenario

message. At the end of a trip, the shuttle sends an "Arrived" message to the customer, which is acknowledged by "Bye".

There are two key security requirements that we want to implement in this application. One is plausibility for shuttles – customers can be sure that they are communicating with real shuttles, rather than with rogue devices masquerading as shuttles. The second requirement is customer trustworthiness: shuttles that receive requests need to believe that the request comes from a customer who wants to take a shuttle, rather than from someone playing customer messages "for fun". To increase trustworthiness, fast shuttles require that potential customers furnish evidence of previous shuttle rides. The taxi service assumes here that someone who has previously taken a taxi is less likely to lie about wanting to take the service again. The only inconvenience for customers is that their first ride via the service is on a slow shuttle.

The first extract shows how the message exchange part of a profile (which is denoted *protocol*) can be simply defined. This is the profile of a fast shuttle and for the part of taking a customer. Recall, the principal is unable to send messages that do not correspond to its profile since they get rejected by receiving principals due to message quality failure.

```
// Take a client protocol
TupleSequence.ExchangedTuple inc1, outc, inc2;
TupleSequence clientSeq = new TupleSequence();
inc1 = new InTuple(new Tuple(
   new Entry[]{new Entry.String("Please"),
      Entry.Type.Strings}));
outc = new OutTuple(new Tuple(
   new Entry[]{Entry.Type.Strings, Entry.Type.Ints}));
inc2 = new InTuple(new Tuple(
   new Entry[]{new Entry.String("Stop"),
```

```
      Entry.Type.Strings, Entry.Type.Ints}));
clientSeq = new TupleSequence(
   new TupleSequence.ExchangedTuple[]{inc1, outc, inc2});
fastTaxiProtocol.append(clientSeq);
```

When operating, the fast service shuttle accepts requests. Its first task is to define the evidence for servicing clients.

```
// Set up stuff -- in main()
TrustedZone tz = TrustedZone.getTrustedZone();
PIN pin = new PIN(111);
tz.setProfile(pin, TaxiProtocol.getFastTaxiProtocol());
tz.setRequiredEvidence(pin, TaxiProtocol.getEvidence());
tz.addRequiredCertificate(pin, Role.installedBy,
   Shuttle.pubKey);

while ( true ) {
    String destination = takeClient();
    handleReceipt(destination);
}

private static String takeClient() {
   String destination;
   // Detect passenger
   Entry te1, te2; Tuple t1;
   te1 = new Entry.String("Please");
   te2 = Entry.Type.Strings;
   t1 = TupleSpace.rd(new Tuple( new Entry[]{te1, te2} ));
   destination = ((Entry.String)t1.get(1)).extractString();
   // Give fare
   Entry te3, te4;
   te3 = new Entry.String(destination);
   te4 = new Entry.Int(30);
   TupleSpace.out(new Tuple(new Entry[]{te3, te4}));
   // Get OK from passenger
   Entry te5 = new Entry.String("Stop");
   TupleSpace.rd(new Tuple(new Entry[]{te5, te3, te4}));
   return destination;
}

private static void handleReceipt(String destination) {
   Entry te1, te2; Tuple t1, t2;
   te1 = new Entry.String("Arrived");
   te2 = new Entry.String(destination);
   t1 = new Tuple(new Entry[] { te1, te2 } );
   TupleSpace.out(t1);
   t2 = TupleSpace.rd(new Tuple(new Entry[]{ Entry.Any }));
```

The program starts by setting the profile of the principal. This can only be done by furnishing the correct PIN (which is something that a thief presumedly

cannot know). The remainder of the code simply implements the protocol with the passenger. The call setRequiredCertificates specifies a profile certificate that must be present in the plea. This certificate attests that the shuttle program was installed by the shuttle company. This permits the client to distinguish a real shuttle from someone pretending to be one by installing the same program. The key used in this certificate is the public key of the shuttle company: this is loaded on the principal when the customer program is installed.

4.3 Analysis of Model

Message quality is useful in addressing the risks outlined in Section 2. With respect to theft, while this risk can never be eliminated, the goal of a security framework is to reduce the benefit to a thief. For instance, employing PINs to access a device and having the owners re-enter his PIN at the start of each session, reduces benefit to thieves. For this reason, PINs are part of our security solution. Further, the plausibility feature of message quality ensures that if Charlie steals Alice's device and sends a message to Bob, then Bob can detect that the device is stolen if the request does not correspond to a message that Alice would normally send. The thief can only use the device for actions (behavior) that Alice specified, and only in the time window that exists before the principal owner is required to re-enter a PIN.

Similarly, viruses exhibit their presence on a device through behavior that is not typical of the owner of the device. Malware that manifests itself in this way is detected through violations of plausibility since messages are sent that are incompatible with the device's profile. Thus, even if validly installed programs get corrupted via stack smashing or buffer overflow, their invalid behavior gets detected. Finally, the usefulness and trustworthiness properties tackle spam.

In the model, the history is stored in the trusted zone. A principal may choose to remove parts of its history from the trusted zone – for instance to economize space or to eliminate redundancy – but it may not add messages explicitly. We contend that most scenarios only need to record a small part of their history. The shuttle service was an example where a principal's history is used as evidence to argue for a message's quality. Another approach to eliminating the history log is for a principal to ask a well-known authority principal to sign a profile certificate (with a role historyValidatedBy) for the requesting principal. The requestor only needs to present its history log in exchange for the profile certificate. The principal can transmit this profile certificate in pleas.

5 Conclusions and Related Work

This paper has presented a security model for ambient information systems. The model concentrates on the property of message quality, which is the cornerstone for implementing other security protocols. The framework has the advantage of not undermining the attractive properties of ambient systems, notably, anonymity and spontaneity of communication. The model is prototyped in Java

and the prototype's implementation uses the TPM. A longer version of this paper is found in [1].

Property-based attestation [13] looks at how the TPM can be used to enable devices to deliver proofs that specific security properties hold. The work does not specify how properties are derived from the measures taken by the TPM. Nonetheless, it shows that the TPM can be used for more elaborate security guarantees than binary (code) attestation.

The plausibility aspect of message quality is similar in concept to proof-carrying code [10] whose main aim is to protect a platform from untrustworthy code. In this approach, a downloaded program is accompanied by a proof of the program's (good) behavior. The host environment can verify the proof mechanically, and if this passes, can then trust the program to run securely. An example of the properties that can be proven in this approach is the sequence of system calls made by the program, e.g., [15] where the host environment verifies that the program does not leak platform information. The message quality model is nonetheless computationally less expensive than the verification mechanisms of proof-carrying code.

Acknowledgments. This work is partly conducted in the context of the PRIAM (Privacy in Ambient Systems) project - an INRIA financed collaboration examining the representation of privacy legislation in modern information systems.

References

1. Bryce, C.: Message quality for ambient system security. Technical Report P11896, IRISA (2008)
2. Bryce, C., Razafimahefa, C., Pawlak, M.: Lana: An approach to programming autonomous systems. In: Magnusson, B. (ed.) ECOOP 2002. LNCS, vol. 2374, pp. 281–298. Springer, Heidelberg (2002)
3. Couderc, P., Banâtre, M.: Ambient computing applications: an experience with the SPREAD approach. In: HICSS, p. 291 (2003)
4. FBI/CSI. 12th annual csi/fbi computer crime and security survey (2007)
5. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (1985)
6. Gosling, J., Joy, B., Steele, G., Bracha, G.: The Java Language Specification, 3rd edn. The Java Series. Addison-Wesley, Boston (2005)
7. Konstantas, D., Jones, V., Herzog, R.: Mobihealth - innovative 2.5/3G mobile services and applications for health care. In: IST Mobile & wireless telecommunications Summit 2002, Thessaloniki, Greece, June 17-19 (2002)
8. Mckitterick, D., Dowling, J.: State of the art review of mobile payment technology. Technical report, June 13 (2003)
9. MetaGroup. Spam, viruses, and content compliance: An opportunity to strategically respond to immediate tactical concerns. Technical Report 800-945-META [6382], International Computer Science Institute (January 2005)
10. Necula, G.C.: Proof-carrying code. In: POPL, pp. 106–119 (1997)
11. Pellegrino, P., Bonino, D., Corno, F.: Domotic house gateway. In: Haddad, H. (ed.) SAC, pp. 1915–1920. ACM Press, New York (2006)

12. Picco, G.P., Murphy, A.L., Roman, G.-C.: LIME: Linda meets mobility. In: Proceedings of the 21st International Conference on Software Engineering, May 1999, pp. 368–377. ACM Press, New York (1999)
13. Poritz, J., Schunter, M., Van Herreweghen, E., Waidner, M.: Property attestation—scalable and privacy-friendly security assessment of peer computers. Technical Report RZ 3548, IBM Research (May 2004)
14. Rivest, R., Shamir, A., Adleman, L.: On digital signatures and public key cryptosystems. Comm. A.C.M. 21, 120–126 (1978)
15. Sekar, R., Venkatakrishnan, V.N., Basu, S., Bhatkar, S., DuVarney, D.C.: Model-carrying code: a practical approach for safe execution of untrusted applications. In: SOSP, pp. 15–28 (2003)
16. Shmatikov, V., Talcott, C.L.: Reputation-based trust management. Journal of Computer Security 13(1), 167–190 (2005)
17. Trusted Computing Group. TPM main specification. Main Specification Version 1.2 rev. 85, Trusted Computing Group (February 2005)
18. Zannone, N.: A survey on trust management languages [reduced]. Technical report, August 01(2004)

Request Satisfaction Problem in Synchronous Radio Networks

Benoît Darties, Sylvain Durand, and Jérôme Palaysi

LIRMM, Université Montpellier II
161 rue Ada, 34392 Montpellier Cedex 5 - France
{benoit.darties,sylvain.durand,jerome.palaysi}@lirmm.fr

Abstract. We study two algorithmical problems inspired from routing constraints in a multihop synchronous radio network. Our objective is to satisfy a given set of communication requests in the following model: nodes send omnidirectional radio transmissions in synchronous slots; during a given slot, a node can receive a message from an adjacent node if and only if no other neighbour is transmitting - otherwise, radio interferences may occur if two or more neighbors transmit in the same slot -. The objective is to minimize the number of slots used. The two problems differ in that the routing policy may be imposed (DAWN-path), or not (DAWN-request). In this second case, a path must be assigned for each request, to define the nodes to use to reach the destination from the source. We present some complexity results, in particular showing that both problems are NP-hard when the network is restricted to a tree. We also present a polynomial algorithm in $O(n^{2K})$ when the number of requests is bounded (by above) by a constant K.

Keywords: Radio Network, Request Satisfaction, Complexity.

1 Introduction

A radio network is a collection of transmitter-receiver stations (or nodes) communicating with one another via multihop wireless links. The use of the radio medium implies some restrictions and properties: whenever a node transmits, all the nodes in its communication range may receive the transmission. Incoming messages have to be forwarded to reach nodes which are located more than one hop away from the source. Since all nodes share the same frequency channel, a collision may occur if two or more neighbors transmit simultaneously, preventing correct reception of the message.

In this paper, we study two communication problems inspired from routing constraints in this kind of network. We consider the following simplified communication model, which has been widely used for the broadcast problem[1, 4–6, 10, 12, 13] or the gathering problem [2, 3] in multihop radio networks : nodes send messages in synchronous slots; during each slot each node acts either as a transmitter or a receiver. A node acting as a transmitter sends a message which can potentially reach the nodes that are in its communication range. A node

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 451–462, 2008.
© Springer-Verlag Berlin Heidelberg 2008

acting as a receiver successfully receives a message from a transmitter node if no other neighbor transmits in this slot. If two or more neighbors of a receiver node u transmit simultaneously in a given slot, then the messages may interfere with each others (collide) and the messages are not transmitted successfully to u. We note that two neighbors u and v may successfully transmit in the same slot to u' and v', if we assume that u' and v are not adjacent, and respectively v and u' : the node u acting as transmitter simply ignores the transmission of v and reciprocally. Such a network is characterized as a Δ-port transmission, 1-port reception, half-duplex, synchronous network. We suppose that the network topology is fixed, at least during the time the problem must be solved. All those properties specify the model we use in this work.

In this context, we consider the problem of satisfying a set of communication requests within a minimum timeframe, by indicating for each node the slots on which it has to relay transiting packets. A request is a couple of source-destination nodes representing the starting and ending nodes of a given message. The second section of this paper details the model and introduces the DAWN-path and DAWN-request problems. General complexity results are discussed in a third section where we will show that these problems are quite difficult even for particular cases. However we present in the fourth section a polynomial algorithm when the number of requests is bounded by a constant K.

2 Describing the Model and Expressing the Problem

2.1 The Model

The network is represented as an undirected graph G where the set $V(G)$ of vertices corresponds to the set of nodes of the network. An edge $e = \{u, v\} \in E(G)$ denotes that u can directly communicate to v (no additional node is required to relay the message) and reciprocally[1] .

A *request* r is a couple $(s,t)|s,t \in V(G)$, where s represents the source and t the destination of the request.

A *path* of length k in a graph G is an ordered list $(v_0, v_1, \ldots, v_k)$, where $v_i \in V(G)$ for any $i \in [0,k]$, and such that the edge (v_i, v_{i+1}) exists in $E(G)$ for any $i \in [0, k-1]$, and all the edges are different. Throughout the paper all the considered paths are simple paths, that is, paths which visit a vertex at most once. Paths are used here to represent a communication road in the network.

Given a graph G and a collection of communication requests R, let P be a *routing function* on R which associates to each $r = (s,t) \in R$ a path $P(r)$ in G, (also denoted by P_r), beginning with s and ending with t.

Given a graph G, a collection of requests R, and a routing function P, we define a *date assignment* d to be a function which takes two arguments r and x, with $r = (s,t) \in R$, $x \in P_r$ and $x \neq t$, and returns an integer $d(r,x)$.

[1] We consider that if x can directly communicate with y then y can directly communicate with x. We can deduce that the graph is symmetrically directed, and will be represented by an undirected graph.

This integer corresponds to a slot, such that x transmits the message of r to the next hop during this slot. Multiplexing is not allowed, this implies that each transmission only contains a single message. A date assignment d is said to be **valid** if and only if for each request $r = (x_0, x_k)$ with $P_r = (x_0, x_1, ..., x_k)$ the proposition $d(r, x_0) < d(r, x_1) < ... < d(r, x_{k-1})$ is true. Moreover we say that a valid assignment is *conflict-free* if and only if for each $d(r, x_i) = d(r', y_j)$ where $r \neq r'$, the following holds :

1. $x_i \neq y_j$: prevents multiplexing
2. $x_{i+1} \neq y_j \wedge y_{j+1} \neq x_i$: a node cannot receive and transmit simultaneously
3. $\{x_i, y_{j+1}\} \notin E(G) \wedge \{y_j, x_{i+1}\} \notin E(G)$: Δ-port-transmission, 1-port-reception.

We note $\max(d) = \max_{r \in R, x \in V(G)} d(r, x)$ the cost of d, i.e. the number of slots used by a date assignment d.

2.2 The DAWN Problem

Given a set of requests to satisfy in a synchronous radio network and a maximum number of slots, the problem DAWN (*Date Assignment in Wireless Network*) consists in finding a conflict-free date assignment along communication paths. According to whether the paths are given (e.g. by the routing function) or not, we distinct two main problems: DAWN-paths and DAWN-request.

The *DAWN-path* problem is stated as follows:

INPUT: An undirected graph G, a collection of requests $R = \{r_i = (s_i, t_i) | 1 \leq i \leq K\}$, a routing function P on R which associates to each request r_i a path $P(r_i)$ linking the vertices of r_i, a natural integer D.

QUESTION: Does a valid and conflict-free date assignment exist in such a manner that the number of required slots is lower than or equal to D?

Let min-DAWN-path be the optimization version of DAWN-path. For each natural integer D we define the D-DAWN-path problem as the subclass of DAWN-path where the maximum number of allowed slots is D. Note that D is bounded by above by $|V(G)| \times |R|$ otherwise the answer is obviously "yes". Figure 1 presents an instance of DAWN-path (1(a)) and a solution (conflict-free assignment) to it (1(b)).

We observe that there is no optimal fixed routing for the DAWN-path problem [8, page 97]. This leads us to propose the *DAWN-request* problem:

INPUT: An undirected graph G, a collection of requests $R = \{r_i = (s_i, t_i) | 1 \leq i \leq K\}$, a natural integer D.

QUESTION: Does a valid and conflict-free date assignment exist in such a manner that the number of required slots is lower than or equal to D?

(a) (b)

Fig. 1. An instance (G, R, P) of min-DAWN-path containing 2 requests $r_1 = (a, f)$, $r_2 = (g, k)$, $P(r_1) = (a, b, c, d, e, f)$ and $P(r_2) = (g, h, i, j, k)$ (sub-fig. a). A valid and conflict-free date assignment and within a minimum number of slots (sub-fig. b).

As stated before, we call min-DAWN-request the optimization version of DAWN-request, and D-DAWN-request the subclass of DAWN-request where D is the maximum number of allowed slots.

3 Complexity Results

We adopt the terminology of [7]: a problem is not approximable within a constant factor if no polynomial approximation algorithm with a constant performance guarantee exists. Moreover an approximation algorithm has a constant performance guarantee of ρ if for each instance I of a problem it finds a solution the cost of which is at most ρ times the cost of the optimal solution for instance I.

In the following subsection we show that in general min-DAWN-path and min-DAWN-request are NP-hard and not approximable within a constant factor. These results are based on the complexity of coloring problems on graphs. The D-COLORING problem [11] consists in assigning a color (represented by a number bounded by above by D) to each vertex assuming that two adjacent vertices are assigned different colors. It is known that D-COLORING is NP-complete for any constant $D \geq 3$, and that the corresponding minimization problem min-COLORING is NP-hard and not approximable within a constant factor.

In a second subsection we show these problems remain NP-hard even when the network is a tree, but here the reduction does not enable us to prove the inapproximability (within some constant). In the third subsection we locate the boundaries between polynomiality and NP-completeness for D-DAWN-path and D-DAWN-request when only D varies.

3.1 Two Difficult Problems

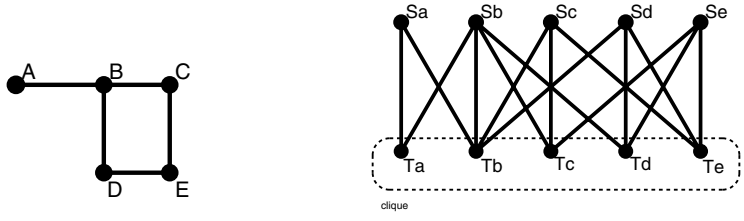
The first theorem proves the NP-completeness of both problems in general.

Theorem 1. *Problems min-DAWN-path and min-DAWN-requests are NP-hard and not approximable within some constant factor. For any $D \geq 3$, decision problems D-DAWN-path and D-DAWN-request are NP-complete.*

Proof. We first prove the NP-completeness of D-DAWN-request by a reduction to D-COLORING.

D-DAWN-request is in NP : given a routing function P and a date assignment d as a solution of an instance, one can check in a polynomial time if P enables each message to reach their destination, and if d is a valid conflict-free date assignment using fewer than D slots.

Let $I_C = (G_C)$ be an instance of D-COLORING, and let us note $n = |V(G_C)|$. From I_C we define an instance $I = (G, R)$ of D-DAWN-request, where G is a graph such that $V(G) = \{s_x, t_x | x \in V(G_C)\}$ and $E(G) = \big\{\{s_x, t_x\} | x \in V(G_C)\big\}$ $\cup\big\{\{s_x, t_y\} | \{x, y\} \in E(G_C)\big\}$ $\cup\big\{\{t_x, t_y\} | x, y \in V(G_C)\big\}$. We define the set $R = \big(r_x = (s_x, t_x) | x \in V(G_C)\big)$ of n communication requests. Clearly, the instance I can be constructed in a polynomial time. Figure 2(b) gives an example of a graph G constructed from the graph G_C of figure 2(a).



(a) A graph G_C (b) The resulting graph G

Fig. 2. Construction of G from G_c

We show that, if there exists a valid conflict-free date assignment for I using $k \leq D$ slots, then there exists a solution to the instance I_C of D-COLORING with cost $k \leq D$ and reciprocally.

Let $S = (P, d)$ be a solution of $I = (G, R)$ where P is a routing function for R and d a valid and conflict-free date assignment for (G, R, P) with cost $k = \max(d) \leq D$. Let us suppose there exists a request $r_i = (s_i, t_i)$ in S such that the message is not directly emitted from s_i to t_i, but requires at least one relay node $t_j | j \neq i$. If t_j transmits in slot u, then no other node s_l may transmit in the same slot, while $\cup_{i \leq n} t_i$ is a clique. Then we can extract a solution S' from S with cost $z' \leq z$ in which s_i transmits directly to t_i at slot u. Thus from any solution S with cost k, we compute a *proper* solution $S' = (P', d')$ with cost $k^* \leq k \leq D$ such that each message is directly transmitted from its source to its destination. Clearly $P'(r_i) = (s_i, t_i) \forall r_i \in R$. Let c be the function which assigns to each vertex $x \in V(G_c)$ the color $d'(r_x, s_x)$. Let us note that $\max(d') = \max(c) \leq D$. The resulting coloring is valid because if x and y are adjacent in G_c then by construction the edges $\{r_x, t_y\}$ and $\{r_y, t_x\}$ exist in G and imply that $d(r_x, s_x) \neq d(r_y, s_y)$.

Reciprocally, let c be a vertex coloring of G_c with cost $k \leq D$. Let P the routing function such that $P(r_i) = (s_i, t_i) \forall r_i \in R$, and d be the date assignment which associates the date $c(x)$ to each couple (r_x, s_x). Then d is a valid conflict-free date assignment such that $\max(c) = \max(d)$.

To conclude, we claim that to any vertex coloring c of I_C corresponds a solution to I composed of a routing function P a valid and conflict-free date

assignment d of (G, R) such that $max(c) = max(d)$, and reciprocally. Since D-COLORING is NP-complete for any $D \geq 3$ [11] and D-DAWN-request belongs to NP, then D-DAWN-request is also NP-complete for any $D \geq 3$. This proof can be extended to prove the NP-completeness of D-DAWN-path for any $D \geq 3$, by adding to the instance I the routing function P such that $P(r_i) = (s_i, t_i) \forall r_i \in R$. By adapting this proof to the optimization versions of these problems we show that min-DAWN-path and min-DAWN-request are NP-hard by a reduction to min-COLORING. Therefore the reduction preserves the inapproximability of min-COLORING, which is NP-hard and not approximable within some constant factor. This allows to conclude. □

We now show that DAWN-path and DAWN-request are still NP-complete even if the network is a tree. This resolves an open question suggested in [9]. The proof can be extended to binary tree or Unit Disk Graph (intersection graph of disks with equal diameters). UDG are often used to model the topology of ad-hoc wireless communication networks. Let us introduce the following propositions:

Lemma 1. *Let $I = (G, R, P, D)$ be an instance of DAWN-path, let x a vertex from $V(G)$ and i an integer. Then there exists an instance $I' = (G', R', P', D)$ of the same problem with $V(G) \subseteq V(G')$, $R \subseteq R'$, such that :*

- *each valid and conflict-free date assignment d on I' requires exactly D slots, and is also a valid and conflict-free date assignment on I,*
- *for each valid and conflict-free date assignment d on I' and each request $r \in R'$ we have $d(r, x) \neq i$,*
- *the instance I can be constructed in a polynomial time.*

Proof. Let us consider the following items:

- an instance $I = (G, R, P, D)$ of DAWN-path,
- a request $r \in R$ such that $P(r) = (s, \ldots, x, y, \ldots, t)$,
- a chain $C = \{c_1, c_2, c_3, \ldots, c_{D+1}\}$ of length $D + 1$, with $V(G) \cap V(C) = \emptyset$,
- a request $r' = (c_1, c_{D+1})$,
- a natural integer $i \in [1, D]$.

We note $H = (V(G) \cup V(C), E(G) \cup E(C) \cup \{\{y, c_i\}\})$ and define $P(r') = (c_1, c_2, c_3, \ldots, c_{D+1})$. Note that if G is a tree, then H is also a tree. Figure 3 schematically illustrates such a construction. We claim the following :

1. The instance $I' = (H, R \cup \{r'\}, P, D)$ can be constructed in polynomial time from $I = (G, R, P, D)$,
2. let d be a valid and conflict-free date assignment for I', then we have $d(r', c_i) = i, d(r, x) \neq i$, and d is a valid date assignment for I. □

This construction presented in the proof of lemma 1 will be used in the proof of theorem 2 to prevent some nodes from transmitting during certain slots. In the following, we say that *a request $r = (s, t)$ starts* at slot t if the source node s proceeds to the transmission of the message of r during the tth time slot.

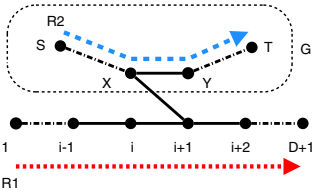


Fig. 3. How to prevent a node x from transmitting in slot i

Let u and D be two natural integers such that $D \geq 6$. We define a tight (u, D) DAWN-path instance as the DAWN-path instance $(C_{[1,D+7u]}, R, P, D)$ where $C_{[1,D+7u]}$ is a chain having $D + (7u)$ vertices $\{1, 2, \ldots, D + 7u\}$ and the edges $\{i, i+1\}|\forall 0 \leq i \leq D+7u-1$. R is the set of $2u$ requests $\{r_1, \bar{r}_1, r_2, \bar{r}_2, \ldots, r_u, \bar{r}_u\}$ with $r_i = \{(7i-5, 7i+D-9)\}$ and $\bar{r}_i = \{(7i-6, 7i+D-8)\}|\forall i \leq u$.

Lemma 2. *Let us consider a tight (u, D) instance for some integer u and D. Let us suppose d is a valid and conflict-free date assignment, and $i \in [1, u]$. We make the following observations:*

1. $(d(r_i, 7i-5), d(\bar{r}_i, 7i-6)) \in \{(5, 1), (1, 3)\}$,
2. $d(r_i, j+1) = d(r_i, j) + 1, \forall j \in P(r_i) - \{7i+D-9\}$,
3. $d(\bar{r}_i, j+1) = d(\bar{r}_i, j) + 1, \forall j \in P(\bar{r}_i) - \{7i+D-8\}$.

Given two natural integers $i \in [1, u]$ and $j \in P(r_i)$:

1. *if $d(r_i, 7i-5) = 1$ then $d(r_i, j) = j - 7i + 6$ and $d(\bar{r}_i, j) = j - 7i + 9$*
2. *if $d(r_i, 7i-5) = 5$ then $d(r_i, j) = j - 7i + 10$ and $d(\bar{r}_i, j) = j - 7i + 7$*

Proof. This obvious proof is left to the reader. □

Theorem 2. *DAWN-path and DAWN-request remain NP-complete even if the graph representing the network topology is a tree.*

Proof. The proof is based on a polynomial reduction of any instance of 3-SAT problem [11] to an instance (G, R, P, D) of DAWN-path where G is a tree.

Let $I_{SAT} = (U, W)$ be an instance of 3-SAT, composed of a set of variables $U = \{x_1, x_2, ..., x_n\}$ and a set of clauses of 3 literals $W = \{c_1, c_2, ..., c_m\}$. We note $n = |U|$, $m = |W|$, and set $D = m + 7n + 3$.

Let us consider the tight (n, D) instance (G_1, R_1, P, D). The main idea of the proof consists in assigning two requests r_i and $\bar{r}_i$ to each variable $x_i \in W$. Thus for the sake of clarity we note r_{x_i} the request r_i and $r_{\bar{x}_i}$ the request $\bar{r}_i$.

Let G_2 be the tree such that $V(G_2) = V(G_1) \cup (c_i, 1), (c_i, 2)|i \in [1, m]$ and $E(G_2) = E(G_1) \cup \big\{\{(c_i, 1), (c_i, 2)\}, \{7n+i, c_i^s\}|i \in [1, m]\big\}$. In the following we note c_i^s the couple $(c_i, 1)$ and c_i^t the couple $(c_i, 2)$ for any integer i. Let R_2 be the set of requests $\{r_{c_i} = (c_i^s, c_i^t)|i \in [1, m]\}$. Now, consider the instance $(G_2, R_1 \cup R_2, P, D)$ (see Figure 4 for an example).

By applying the construction of lemma 3.1 several times, we create an instance $I = (G, R, P, D)$ by adding elements to $(G_2, R_1 \cup R_2, P, D)$ as follows : for each

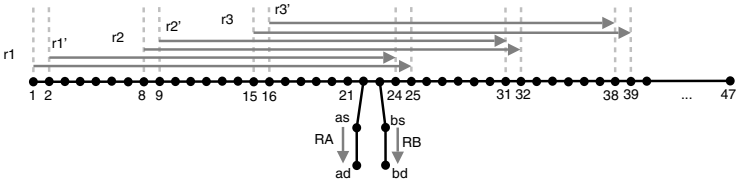


Fig. 4. A graph G_2 and a set of requests $R_1 \cup R_2$ constructed from an instance $I_{SAT} = (U, W)$ where $U = \{x_1, x_2, x_3\}$ and $W = \{c_1, c_2\}$. Here $D = 26$.

request $r_{c_i} = (c_i^s, c_i^t)$ we prevent c_i^s from transmitting to c_i^t the message of request r_{c_i} during each slot $t \leq D$, except for 3 specific ones which are defined from the literals of the clause c_i : if c_i contains the positive (resp. negative) literal associated to the variable $x_j | j \leq n$, then c_i^s is allowed to transmit in slot $7n + i - 7j + 6$ *(resp.* $7n + i - 7j + 8$*)*. Since G_2 is a tree, G is also a tree and the routing function is still obvious.

The size of I is polynomial in the size of I_{SAT}, and it can be constructed in a polynomial time. We claim that if there is a solution in D slots to I, then we can deduce a solution to the instance I_{SAT} and reciprocally.

Let us consider a valid and conflict-free date assignment d on the instance I. According to lemma 2 and for each integer $i \in [1, n]$, one request of $\{r_{x_i}, r_{\bar{x}_i}\}$ must start at slot 1, and the other as soon as possible, and once a request has been started, its progression cannot be stopped. Moreover each request $r_{c_i} | i \in [1, m]$ is clearly satisfied by d. The slot which has been assigned to (r_{c_i}, c_i^s) is one of the three allowable values defined from the literals of clause c_i. Our construction implies that there exists j such that $d(r_{c_i}, c_i^s)$ is of the form $7n + i - 7j + 6$ or $7n + i - 7j + 8$, according to x_j is a positive or a negative literal. If x_j is a positive literal, then the source of r_{x_j} is located on vertex $7j - 5$, i.e. at distance $7n + i_7 j + 5$ from the vertex $7n + j$ which is adjacent to c_i^s. Then r_{x_j} necessarily starts at slot 1 - and we have $d(r_{x_j}, 7n + i) = d(r_{c_i}, c_i^s)$ - otherwise messages would collide. We adopt a similar reasoning when x_j is a negative literal. Then for each clause c_i, there exists at least one (positive or negative) literal $l \in c_i$, such that the request r_l starts before $r_{\bar{l}}$. By affecting the value "True" to all variables x_i where r_{x_i} has been started at slot 1 (i.e. before $r_{\bar{x}_i}$) and "False" otherwise, we obtain a solution to the instance I_{SAT}.

Reciprocally we can deduce a solution to the instance I from a solution to I_{SAT} : for each variable x_i we start the request r_{x_i} before $r_{\bar{x}_i}$ if and only if x_i has the value "True". For each clause c_i, we start r_{c_i} at the first valid and available slot (this slot exists since the clause c_i is satisfied by at least one literal).

To conclude we point out that 3-SAT is NP-complete and that DAWN-path belongs to NP. Then the DAWN-path is NP-complete even if the network is a tree. This implies the NP-completeness of DAWN-request, since there is one unique path linking each source to its destination when the network in a tree.□

3.2 Locating the Boundaries between Polynomiality and NP-Completeness

We have shown that DAWN-path and DAWN-request are NP-complete even when the network topology is very restrictive. In the following subsection, we focus our interest on the influence of the maximum number of slots D on the complexity of these problems.

Theorem 1 already affirms that D-DAWN-path and D-DAWN-request are NP-complete when $D \geq 3$. By the way when $D = 1$ one can verify in polynomial time if an instance can be satisfied: for each request $r = (s, t)$, s must be adjacent to t, and can only emits at the first slot. Hence t cannot be adjacent to another source node s_2, and s must be a source only for t. The next theorem states that one can check in a polynomial time if there is a solution to a D-DAWN-path instance when $D \leq 2$.

Theorem 3. *The 2-DAWN-path decision problem is polynomial*

Proof. Let $I = (G, R, P)$ an instance of 2-DAWN-path. One can suppose that for each request $r = (s, t)$ the path $P(r)$ contains at most one vertex l between s and t, otherwise the instance is clearly insolvable. We propose an algorithm in three steps :

- Dates are forced for transmitters belonging to two-hop requests.
- Therefore dates 1 (resp. 2) are spread to every transmitter which cannot transmit during the second (resp. first) slot.
- Remaining dates are computed using a 2-SAT-like algorithm. □

Theorems 1 and 3 show that D-DAWN-path is polynomial when the maximum number of slots D is lower than or equal to 2, and becomes NP-complete when $D \geq 3$. Theorem 4 proves that D-DAWN-request is NP-complete for $D = 2$.

Theorem 4. *The 2-DAWN-request decision problem is NP-complete*

Proof. Let $I_{SAT} = (U, W)$ be an instance of 3-SAT where $U = \{x_1, \ldots, x_n\}$ denotes a set of variables and $W = \{c_1, \ldots, c_m\}$ a set of clauses. For each variable $x_i \in U$ let $H_{x_i} = (X, Y, E)$ be the complete bipartite graph $K_{3,2}$ such that $X = \{(1, x_i), (1, \bar{x_i})\}$ and $Y = \{(2, x_i), (2, \bar{x_i}\}$. For each clause $c_i = \{l_1, l_2, l_3\} \in W$, let $F_{c_i} = (X, Y, E)$ be the complete bipartite graph $K_{3,2}$ such that $X = \{(1, c_i), (2, c_i)\}$ and $Y = \{(c_i, l_1), (c_i, l_2), (c_i, l_3)\}$. Note that l_1 to l_3 can be positive or negative literals, each literal corresponding to a variable $x_i \in U$. Let $I = (G, R)$ be a 2-DAWN-request instance with $V(G) = \big\{ \{\bigcup_{x_i \in U} V(H_{x_i})\} \cup \{\bigcup_{c_i \in W} V(F_{c_i})\}\big\}$ and $E(G) = \big\{ \{\bigcup_{x_i \in U} E(H_{x_i})\} \cup \{\bigcup_{c_i \in W} E(F_{c_i})\} \cup \{(c_i, l), (2, l) \mid c_i \in W, l \in c_i\}\big\}$. Figure 5 presents an example of graph G constructed from a 3-SAT instance $I_{SAT} = (U, W)$. The requests collection R contains exactly all the requests of the form $((1, c_i), (2, c_i)) | c_i \in W$, and $((1, l), (2, l))$ where l is a literal corresponding to a variable $x_i \in U$.

Let d be valid date assignment d for I using only 2 slots. We assign to any literal l the value "True" if and only if $(1, l)$ emits at slot 1 , and "False" otherwise. Given a clause $c_i \in W$, exactly one node of the form (c_i, l) emits at

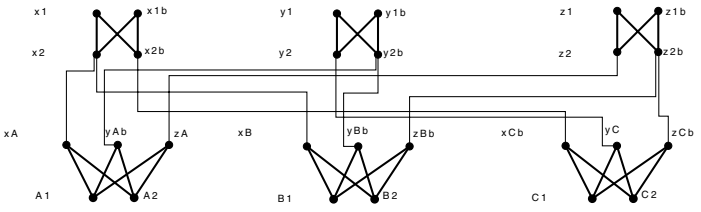


Fig. 5. graph constructed from $I_{SAT} = (U, W)$ with $U = \{x_1, x_2, x_3\}$ and $W = \{c_1, c_2, c_3\}$ with $c_1 = \{x_1, \bar{x_2}, x_3\}, c_2 = \{x_1, \bar{x_2}, \bar{x_3}\}$ and $c_3 = \{\bar{x_1}, x_2, \bar{x_3}\}$

slot 2. This node is adjacent to a node $(2, l)$, which could receive the message of the request $((1, l), (2, l))$ at slot 1 only. Thus $(1, l)$ is true and c is satisfied.

Reciprocally, suppose that I_{SAT} admits a solution. For each literal l fixed at "True", let us assign the date 1 to vertex $(1, l)$ and the date 2 to $(2, \neg l)$. Let c_i be a clause from W. Date 1 is assigned to vertex $(1, c_i)$. Date 2 must be assigned to exactly one adjacent vertex of $(1, c_i)$. We can choose any couple with the corresponding literal l fixed at "True". This date is available since (c_i, l) is only adjacent to $(2, c_i)$ (the destination) and $(2, l)$, which has already received the message at slot 1. Since 3-SAT is NP-complete and 2-DAWN-request is in NP, 2-DAWN-request is indeed a NP-complete problem. □

Thus we have shown that knowledge of the routing policy plays a role in the complexity of both problems, since the limit between polynomiality and NP-completeness is located between 2 and 3 for DAWN-path, but between 1 and 2 for DAWN-request.

4 Solving Instances with a Bounded Number of Requests

We give a polynomial algorithm for min-DAWN-path problem and min-DAWN-request problem when the number of requests is bounded by above by a constant K. The following notation and definition will be used :

- For $i \in [1, n]$, let $\pi_i(t)$ denotes the ith element of a n-tuple $t = (x_1, x_2, ..., x_n)$.
- The *contracted form of a tuple* $(x_1, x_2, \ldots, x_n)$ is the tuple $(x_i)_{(i \in [\![1,n]\!]) \wedge (x_i \neq x_{i-1})}$.

We propose a polynomial algorithm to solve instances I with a number of requests bounded by K. We build a state graph, where each vertex describes a possible state of the network at a given slot. An edge links X and Y if and only if one can go from state X to state Y or reciprocally in only one slot. For a given min-DAWN-path instance $I = (G, R = (r_1, r_2, ..., r_k), P)$ with $|R| \leq K$ the state graph $S(I)$ is defined as follows:

- the vertex set is the cartesian product $P(r_1) \times P(r_2) \cdots \times P(r_K)$. A vertex $X = (x_1, x_2, \ldots, x_K)$ indicates that for each $i \leq K$, the message of request r_i has reached the node $\pi_i(X) = x_i \in V(G)$.

– there is an edge between $X = (x_1, x_2, \ldots, x_K)$ and $Y = (y_1, y_2, \ldots y_K)$ of $S(I)$ if and only if the simultaneous emission of nodes $\{x_i | x_i \neq y_i, 1 \leq i \leq K\}$ allows to deliver each message from $x_i \neq y_i$ to y_i in one slot only. Formally, for $X = (x_1, x_2, \ldots, x_K)$ and $Y = (y_1, y_2, \ldots y_k)$, $(X, Y) \in E(S(I))$ if we have, for each i such that $x_i \neq y_i$:
  - x_i and y_i are immediately consecutive in $P(r_i)$,
  - there is no $j \neq i$, such that $x_j \neq y_j$ and $\{x_j, y_i\} \in E(G)$,
  - for each $j \neq i$ such that $x_j \neq y_j$, we have $|x_i, y_i, x_j, y_j| = 4$.

The state graph S(I) of a min-DAWN-request instance I is constructed according to the same method, except that the set of vertices is the set $V(G) \times V(G) \times \cdots \times V(G) = V(G)^K$. These state graphs can be constructed in a polynomial time, since K is a constant. We can distinguish two vertices in $S(I)$: the *source* $(s_1, s_2, \ldots, s_K)$ and the *sink* $(t_1, t_2, \ldots, t_K)$ where s_i and t_i are respectively the source and the target of the request r_i for any $i \in [1, K]$.

We conclude the section with this theorem:

Theorem 5. *min-DAWN-path and min-DAWN-request can be solved by a polynomial-time algorithm with complexity $O(n^{2K})$ when the number of requests is bounded by above by a constant K.*

Proof. (sketch of the proof) Consider $I = (G, R = (r_1, r_2, \ldots, r_K), P)$ a min-DAWN-path instance with $|R| \leq K$, and let us construct the state graph $S(I)$.

One may check that a shortest path between the source and the sink in $S(I)$ can be associated with an optimal conflict-free date assignment and reciprocally. Such a path can be found with a $O(n^{2K})$ complexity algorithm. □

5 Conclusion and Perspectives

We have studied the complexity of the request satisfaction problem in a synchronous radio network. Table 1 summarises the results of this paper.

We have suggested other results [8] on particular cases i.e. on dynamic network, or when requests cannot be paused as soon as they have started. Possible perspective for this research work consist in studying the complexity of DAWN-path and DAWN-request on specific topologies, in order to discover polynomial cases even when the number of requests is unbounded. Particularly the complexity when the network is a chain is an open question (however for this case, we

Table 1.

DAWN-path:		**Complexity:**	**DAWN-request:**	
Min-DAWN-path		NP-hard (even in trees), not approximable within some constant factor.	Min-DAWN-request	
D-DAWN-path	$D \leq 2$	Polynomial	$D \leq 1$	D-DAWN-request
	$D \geq 3$	NP-complete	$D \geq 2$	
min-DAWN-path, $\|R\| \leq K$		Polynomial : $O(n^{2K})$	min-DAWN-request, $\|R\| \leq K$	

have a constant factor approximation algorithm). Moreover, finding heuristics with performance guarantee for difficult instances constitutes a natural extension of this work.

References

1. Alon, N., Bar-Noy, A., Linial, N., Peleg, D.: A lower bound for radio broadcast. J. Comput. Syst. Sci. 43(2), 290–298 (1991)
2. Bermond, J.-C., Galtier, J., Klasing, R., Morales, N., Pérennes, S.: Hardness and approximation of gathering in static radio networks. In: FAWN 2006, Pisa, Italy (2006)
3. Bermond, J.-C., Peters, J.: Efficient gathering in radio grids with interference. In: Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel 2005), May 2005, pp. 103–106 (2005)
4. Chelius, G.: Architectures et Communications dans les réseaux spontanés sans fil. PhD thesis, INSA de Lyon, INRIA Rhône Alpes, France (April 2004)
5. Chlamtac, I., Kutten, S.: On broadcasting in radio networks - problem analysis and protocol design. IEEE Transactions on Communications 33, 1240–1246 (1985)
6. Chlamtac, I., Weinstein, O.: The wave expansion approach to broadcasting in multihop radio network. IEEE Transaction Communication (39), 426–433 (1991)
7. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 2nd edn. Dunod (2001)
8. Darties, B.: Problémes algorithmiques et de complexité dans les réseaux sans fil. PhD thesis, LIRMM, Université Montpellier 2, France (December 2007)
9. Darties, B., Palaysi, J.: Satisfaction de requêtes par affectation de dates d'émissions dans les réseaux radios. In: Rencontres francophones du Parallélisme (RenPar'17), pp. 157–163 (2006)
10. Fraigniaud, P., Lazard, E.: Methods and problems of communication in usual networks. In: Proceedings of the international workshop on Broadcasting and gossiping, pp. 79–133. Elsevier North-Holland, Inc. (1994)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A guide to the theory of NP-completeness. W.H. Freeman, New York (1979)
12. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks 18, 319–349 (1986)
13. Kowalski, D.R., Pelc, A.: Centralized deterministic broadcasting in undirected multi-hop radio networks. In: APPROX-RANDOM, pp. 171–182 (2004)

A Novel Mobility Model from a Heterogeneous Military MANET Trace

Xiaofeng Lu[1,3], Yung-chih Chen[2], Ian Leung[1], Zhang Xiong[3], and Pietro Liò[1]

[1] Computer Laboratory ,University of Cambridge,
15 JJ Thomson Avenue, Cambridge CB3 0FD
{firstname.lastname}@cl.cam.ac.uk
[2] Department of Computer Science
University of Massachusetts at Amherst
140 Governors Drive, Amherst, MA 01003-9264
yungchih@cs.umass.edu
[3] College of Computer Science
Beijing University of Aeronautics and Astronautics
XueYuan Road, Beijing 100083
xiongz@buaa.edu.cn

Abstract. In this paper we describe our analysis of a real trace and propose a mobility model. The trace data we used for this study was collected from a military experiment carried out in Lakehurst, N.J., U.S.A. The structure of these entities in the trace is novel, say layered and heterogeneous—some nodes moved on the ground whilst some hovered in the sky. Evaluation results show our mobility model well and truly captures some aspects of the spatial and temporal characteristics of the real traces. This mobility model can be used to generate synthetic traces with different travel schedules. Such experiments are costly to be realized in real world scenarios

1 Introduction

Mobility models are important in simulation-based studies of wireless Ad hoc networks. However, the majority of models have little relevance to real-world movements, such as Random Waypoint. In order to thoroughly analyze the performance of network protocols of Ad hoc networks, it is imperative to capture the gist of the movements by providing a realistic mobility model [1, 2, 6].

Many mobility models have been developed to simulate the movements of real life systems. There are two types of mobility models used in the simulation of networks: traces and synthetic models [2, 5]. Traces log the movements of individuals in real life systems. If number of entities in the traces is large and the period time is long enough, they can provide real and accurate information of mobility pattern. An increasing number of researchers extract the mobility characteristics from actual traces to build more realistic mobility models [7, 11-14, 16-18].

In this paper, we analyze a novel set of mobility traces from an actual military training trace. Suppose a number of soldiers navigate through a highly hostile terrain where they are highly susceptible to adversities such as an ambush and traps. In the

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 463–474, 2008.
© Springer-Verlag Berlin Heidelberg 2008

hope of maximizing the security, a common practice is to traverse through the area along the same path one squad after another. These squads depart along the same path only when the frontal squad arrives somewhere of confirmed security. And the frontal squads may also have to rest and wait for the following squads for supply and back up. During the movement, if a squad is in danger, squads nearby would immediately move to (or, of course, near if it is unsafe) the scene for assistance. Such mobility patterns are common in military settings, and are also highly applicable in scenarios such as large-scale survivor search and even planetary explorations.

At a high level of abstraction, a mobility model consists of a set of rules that defines the spatial and temporal characteristics of the mobile nodes. Spatial characteristics dictate the choice of a new direction or a new destination. Temporal characteristics define how fast a node should travel to its destination or when a node should depart or stop if required [3, 10]. Structural characteristics are also an important part of a mobility model, which is often neglected by researchers. In a network in which nodes move separately, the structure of these nodes is not so important as they are disordered and unsystematic. In an organized network, however, the structure of the network depends on the motion of nodes. Only when both the mobility and structural characteristics are clearly known can we define a realistic mobility model.

This paper is organized as follows. Section 2 is a survey of related research in the area of mobility modeling. The structural, spatial and temporal characteristics we extracted from the traces are described in Section 3, 4 and 5. We introduce our mobility model in Section 6. We conclude the work in Section 7.

2 Related Work

The beginning stage of mobility model research saw numerous influential synthetic models when real movement traces were difficult to obtain [1, 2, 5]. Some widely used synthetic mobility models are Random Walk, Random Way Point, Random Direction Walk, etc. Random Walk Mobility Model [5] was developed to simulate erratic movements. In this model, a node moves from one location to a new location by randomly choosing a direction and speed. Random Direction model in [4] is a modified version of the random mobility model. Another widely used random mobility models is the Random Waypoint model [1]. In this model, a node selects its destination randomly and the speed from a velocity range. When it reaches its destination, it pauses for some time and then selects a new destination and speed and continue to move.

Recent years also saw an increasing number of researches on mobility models with emphasis on real mobility traces [7, 8]. Tuduce and Gross proposed a WLAN mobility model from a WLAN trace [8]. The trace was gathered from a campus wireless network consisted of 166 APs. They presented a framework to extract the mobility parameters to build the WLAN model. But the spatial parameters and temporal parameters in the model are independent of each other. The study in [10] suggests that there exists a strong correlation between the space and time dimension. Treating space and time independently is not adequate. Kim et al. collected a campus Wi-Fi network trace at Dartmouth College from nearly 10,000 users [7]. The space and time dimension in this

model is therefore correlative. The users in their traces traveled separately. Students usually go to somewhere by themselves, they do not move in organized groups, which is a significant difference from the entities in our trace.

Some researchers studied application dependent traces. Aschenbruck et al. studied a disaster scenario trace in which there were 150 communication devices [16]. Their model shows specific characteristics like heterogeneous node density, because in disaster area scenarios, nodes move in a structured way based on civil protection tactics. Zhang et al. looked into a bus mobility trace taken from UMass DieseNet which consisted of 40 buses [14]. They found the inter-meeting time between buses was not constant but random because of traffic conditions whereas the inter-contract time of buses was indeed periodic. The periodicity in buses traces exists because they always travel along a closed route. It is fundamentally different from node behavior in our trace.

3 Structual Characteristic

The trace data we used for this study was collected from a MANET consists of 64 jeep vehicles and four Unmanned Aerial Vehicles (UAVs). These vehicles traveled over an area of approximately 240 square kilometers near Lakehurst, New Jersey, USA for 180 minutes. The system logged every vehicle's ID, GPS location and communicational pathloss data throughout the period per second time. We believe the structure of the network in our traces differs from those of the usual traces [7, 9, 11, 12]. For instance, the nodes in [7, 9, 11, 12] move individually, the motion of one user having no relation to the others. Nodes in military scenarios do not move individually; instead they often form groups to accomplish a task cooperatively.

We say that two nodes are out of communication when the pathloss between the two nodes is higher than a threshold 130dB, which is a known minimum requirement of radio communication. With this threshold, we are able to investigate the nodes' mobility and their interaction with each other.

A node is said to belong to a specific group when it within the communication range of any member in that group (ie., pathloss smaller than 130dB). In Figure 1, we

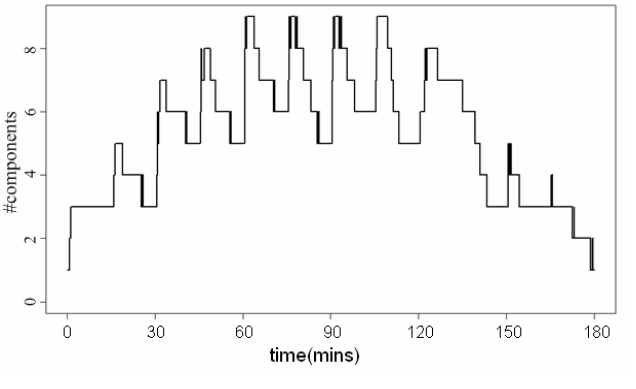


Fig. 1. The number of disconnected components with pathloss threshold 130 dB

show that the number of disconnected groups of the network varies from 1 to 9. When we keep track of those nodes belonging to specific groups, we discover that the 64 vehicles are of nine groups. Nodes of the same group always move as a unit group. They move in the same direction, with relatively the same speed, and exhibit identical mobility behavior.

The clustering coefficient of a vertex in a graph quantifies how close the vertex and its neighbors are from being a complete graph. We study the clustering coefficient to know how stable the cluster structure is. Denote the graph by G = (V, E). Let N_i = {j∈ V: (i,j) ∈ E} and $k_i = |N_i|$. The clustering coefficient of the graph, $\overline{C}$, is given by

$$\overline{C} = \frac{1}{n}\sum_{i \in V} C_i \tag{1}$$

Where $C_i = \frac{\sum_{j,k \in N_i} |\{(j,k)\}|}{k_i(k_i - 1)}$.

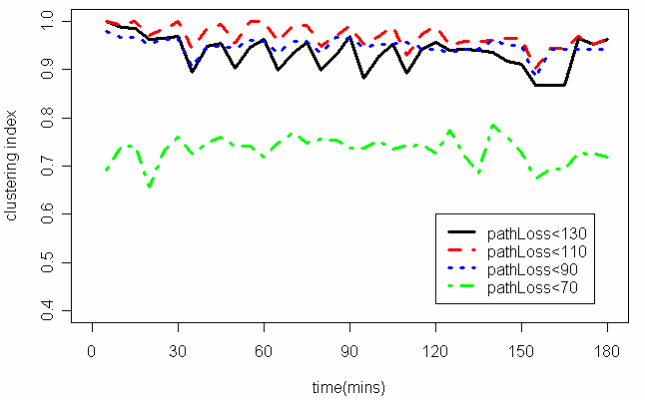


Fig. 2. Clustering coefficient change with different pathloss thresholds

Figure 2 shows the clustering coefficient over time for different pathloss thresholds. Not surprisingly, the clustering coefficient is close to one, reflecting the very strong clustering behavior associated with the nine groups of vehicles.

4 Spatial Characteristic

The mobile nodes in the traces were heterogeneous. The vehicles were organized in teams which moved on the ground whilst the UAVs hovered above them. Since UAVs travelled with much greater flexibility than the vehicles, the movement behavior of former were different from that of the latter. Figure 3 displays the layered heterogeneous structure of the mobile Ad hoc network.

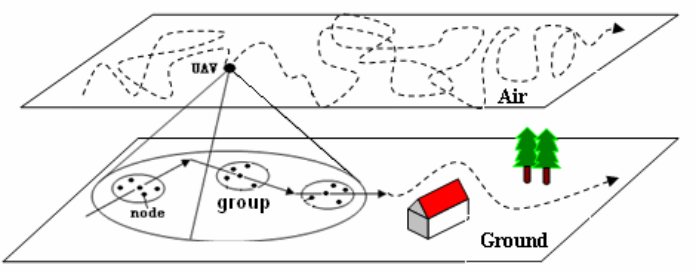


Fig. 3. Heterogeneous nodes

In this section, we discuss the spatial characteristics of our traces. We begin by analysing the direction characteristics. Direction characteristic describes a node's direction of travel. We employ the relative direction angle as the metric of direction change. We define the relative direction angle θ to be the angular distance between the new direction $\overrightarrow{BC}$ and the former direction $\overrightarrow{AB}$ as illustrated in Figure 4 (a). Since the traces do not contain direction information, we employ equation (2) to compute the absolute angular change in direction, θ.

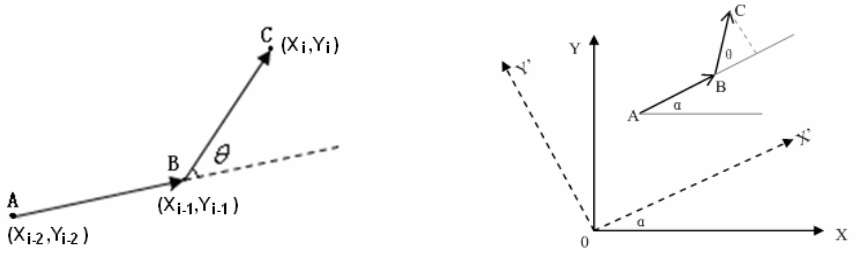


Fig. 4. (a) Direction vector and Relative direction angle. (b) Coordinate transform.

$$\theta = Arc\cos\left(\frac{\overrightarrow{AB}\cdot\overrightarrow{BC}}{\left|\overrightarrow{AB}\right|\cdot\left|\overrightarrow{BC}\right|}\right) \tag{2}$$

Since θ does not tell us whether the vehicle turned left or right, we obtain this information by coordinate transform. In the original coordinate system, the angular distance between last direction vector $\overrightarrow{AB}$ and the X axis is α. In coordinate transform, we define the direction of the last direction vector $\overrightarrow{AB}$ as the positive direction of X axis in the new coordinate system as Figure 4 (b) shows. Then we calculate the new coordinates of point C (x', y') in the new coordinate system using equation (3). If this vehicle's new coordinate y' in the new coordinate system is positive, it means that the vehicle turns left, otherwise the vehicle turns right, and we reverse θ to $-\theta$.

$$(x', y') = (x, y) \cdot \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \tag{3}$$

Figure 5 shows the trajectories of UAVs and ground vehicles when they were moving in the 180 minutes. In this Figure, the circle and triangle paths are the trajectories of the two UAVs respectively and two solid lines are the track of jeeps. As the Figure shows, the UAV turned to left and right with larger relative direction angle.

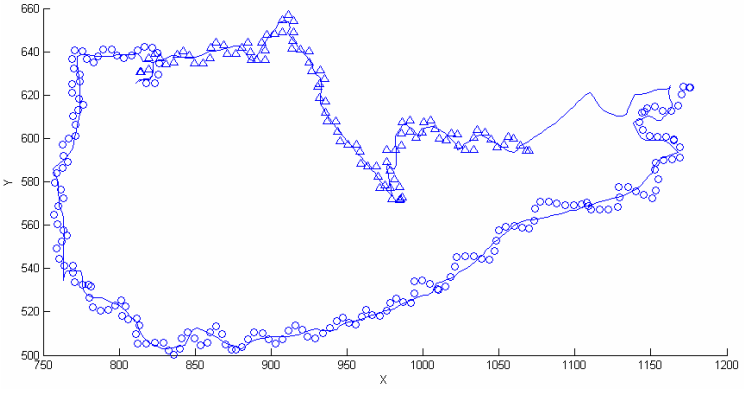


Fig. 5. The trajectories of UAVs and ground vehicles

Table 1 depicts the distribution of absolute relative direction angle during the whole 180 minutes. From the statistics we can simply conclude that a mobile node does not select its new direction randomly from a uniform distribution (0^o, 180^o). Most of the absolute relative direction angles are smaller than 30^o. Compared with vehicles, UAVs move with much greater flexibility, there is about 30.5% of the absolute relative direction angle being in the range of (30^o, 90^o).

Table 1. Distribution of absolute relative direction angle of UAV and vehicle

Angle(degree)	0-30	30-60	60-90	90-120	120-150	150-180
UAV	67.4%	19.8%	10.7%	1.5%	0.3%	0.3 %
Vehicle	93.08%	3.9%	1.73%	0.43%	0.43%	0.43%

5 Temporal Characteristics

Travel duration and pause duration characterize vehicles' temporal behavior. We analyse the distance a vehicle covered during two sampled time to estimate vehicle's motion phase. If the distance a vehicle covered between two sampled positions is too short, we assume that the vehicle was not in motion during that interval of time. In our traces, we sample the positions of a vehicle at 10 second intervals to determine its

movement. As the ground vehicles in the traces were jeeps, their speeds ranged from 30 km per hour to 120 km per hour; hence we set the threshold to be 10 meters for deciding whether or not the vehicle is moving.

We define the pause duration to be the length of time from when a vehicle stopped to when it started off again. Plotting the change in position at 10 seconds intervals effectively yields a speed time graph as depicted in Figure6. As this figure shows, the vehicle would travel from one place to another place at an average speed of about 16m per second, which roughly translates to 57 km/h. The speed tends to increase and drop quite sharply at the start and stop of node movement.

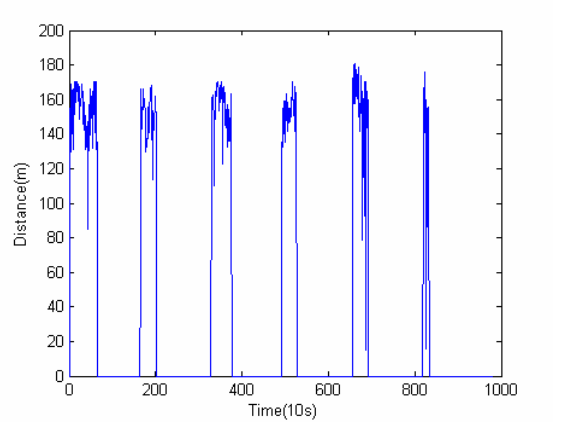


Fig. 6. The distance between time intervals

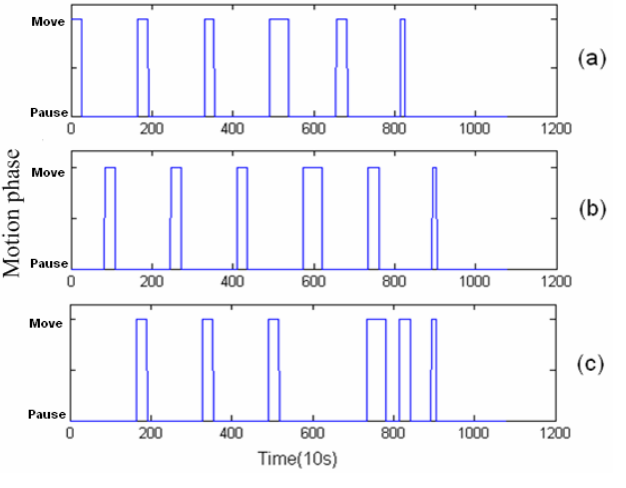


Fig. 7. The travel schedule of three groups

Figure 7 illustrates the variation of motion phase of three groups along the same path. It shows that all three different groups had 6 travel phases even though they did not start off at the same time. Since all three groups traveled along same path but paused at different times, we deduce that when vehicle groups arrived at a checkpoint, they would stop and pause for some time. We observe that the six pause durations from each group were not exactly the same length, but the travel durations from checkpoint i to checkpoint i+1 of the three groups are roughly equal. The length of travel duration depends on the distance from a checkpoint to the next checkpoint and vehicle's speed. This means different groups proceeded at roughly equal velocities.

As discussed, vehicle groups departed from the start point at different times with different travel and pause durations. Some groups converged and separated periodically such as groups (a) and (b) in Figure 7, while others never met each other as groups (a) and (c) in Figure 7.

6 Mobility Model

6.1 Mobility Model

Here we employ the various studied characteristics of our traces to develop a mobility model. As we have mentioned above, mobility model consists of a set of rules that defines the movement characteristics of the mobile nodes.

(1) Structural rules set:

- Our model assumes that nodes are organized into groups and nodes within a group have the same travel duration and pause duration.
- Within a group, nodes can be heterogeneous and have different mobility flexibility, e.g. mobility radius.
- Each group has a FIFO destination queue to save the location of destinations.
- Groups start off at different times and have heterogeneous travel schedules.
- Nodes of a group start at the same place in the beginning.

(2) Mobility rules for a group:

Step 1: All nodes of a group pause for a certain time, given by the pause duration.

Step 2: After the pause duration, if the destination queue is empty, go to Step 3, otherwise go to Step 5.

Step 3: Generate a group destination with relative direction angle being in the range (-30^o, 30^o). Broadcast this new destination and put into all groups' destination queues.

Step 4: Calculate the group's travel duration according to the distance to this destination and the average group velocity.

Step 5: Pop a destination from the group's queue. Every node of this group calculates its destination around the group destination randomly in a circle region whose radius depends on the type of node. Heterogeneous nodes have different mobility radius. For example, the radius of UAV is about 2 or 3 times of that of vehicle as Figure 8 shows.

Step 6: Every node of this group starts off to its destination.

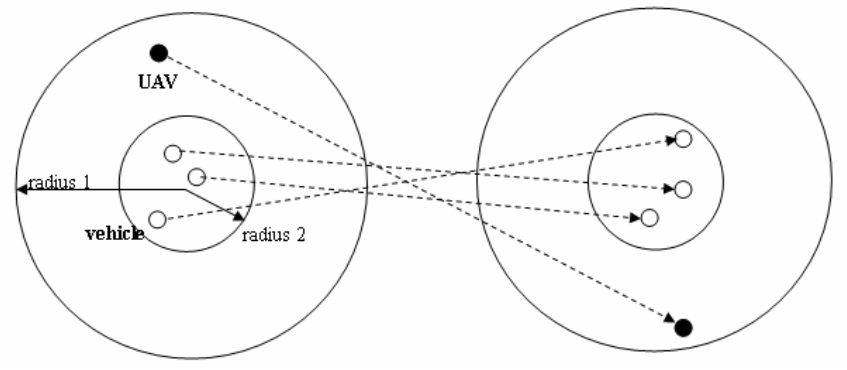


Fig. 8. The movements of different types of node

Note that we do not add the end restrict in the model. We can use different restricts to end the program, such as program running time or cycle times.

In this model, if a group which is behind of some groups departs from a place, it does not need to calculate its next destination. Its destination queue has at least one destination, because it passes by fewer destinations than the frontal groups. To those groups in front of all other groups, when one of them wants to depart ahead of others, it checks its destination queue firstly. If its destination queue is empty, actually its destination queue should be empty because no other groups depart ahead of it, this group will calculate the newest destination and put it into all other groups' destination queues. Hence, all these groups have a common new destination.

Those groups which start off from the start point behind of others still have the opportunity to catch up with and exceed the frontal groups, if they have shorter pause durations. We can assign different travel schedules to different groups to study the structural and spatial variations between these groups.

6.2 Evaluation

We divided the data set into two sets: a training set and a test set. Firstly, we analyzed the mobility characteristics of training set. Then we make the mobility model to generate synthetic data according to the values of these mobility characteristics. We compare the spatial and temporal characteristics of the synthetic data our model generated with the real mobility characteristics of the traces to evaluate our model [12]. To acquire a more accurate quantity of the difference between the synthetic values and the data set, we use the relative error to estimate the model. We calculate the relative error using the equation (4)

$$relative\ error = \frac{\sum_{i=i}^{n} | r_i - s_i |}{\sum_{i=1}^{n} r_i} \tag{4}$$

where r_i is the value of testing data and s_i is the simulation data, and n is the number of performance time

Firstly, we evaluate the spatial characteristic of our model. The vehicles changed their directions according to the terrain and traffic situation in real world, but our model do not consider these kinds of conditions now, so we could not compare the directions between the test set and that in model directly. We compare the distributions of relative direction angle of test set with that of synthetic data as Figure 9 and 10 shows. The relative error of the distribution of relative direction angle of the vehicle is as low as 1.7%, and that of the UAV is about 2.1%.

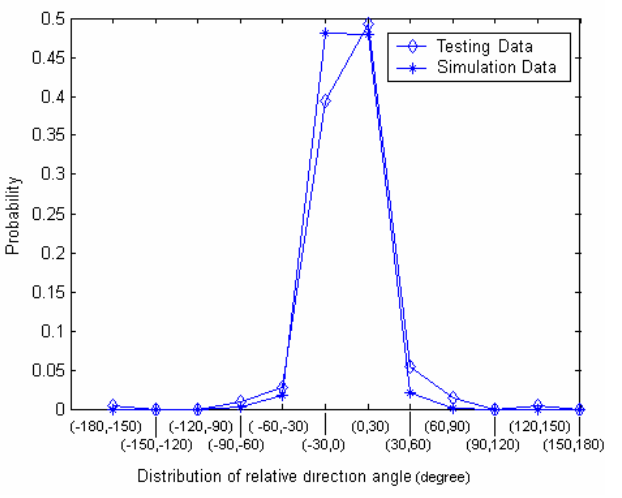


Fig. 9. Vehicle's distributions of relative direction angle of testing data and simulation data

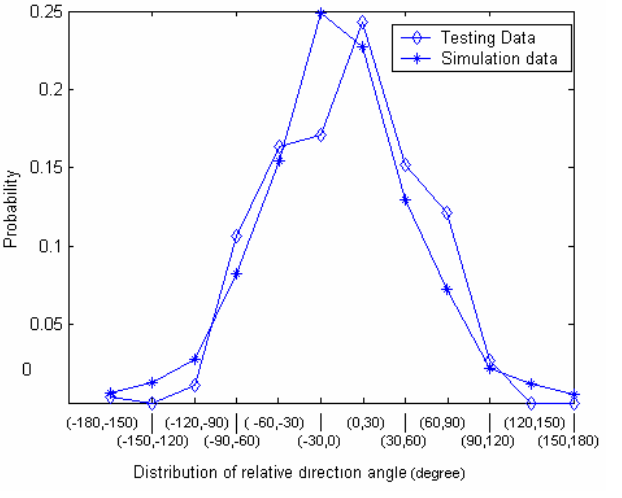


Fig. 10. UAV's distributions of relative direction angle of testing data and simulation data

Secondly, we evaluate the temporal characteristics: the travel schedule. We use the training set to get the values of the travel duration and pause duration and use these values in our model. The relative error of travel schedule is 4.3%. This value is rather low and indicates our model matches the temporal characteristics of the real system.

With this mobility model, we can study the distances between any two groups, the network connectivity and the performance of routing protocols with different travel schedules. We can also understand what combinations of travel schedules allow the nodes to attain maximum average network connectivity or the shortest average inter group distance, which is important in military scenarios.

7 Conclusion

In this paper, we propose a novel mobility model based on the mobility characteristics extracted from a layered heterogeneous military MANET trace collected from Lakehurst, New Jersey. In the model, nodes are divided into many groups and these groups travel along the same route but with different time schedules. When a group reaches its current destination, it pauses for some time and then departs again to the next destination. If there is no destination in the group's destination queue, it will generate a new destination and notify all other groups. All the nodes of a group have the same travel schedule, but they select themselves destinations around its group's destination randomly with different mobility radiuses.

Evaluation results show our mobility model well and truly captures some aspects of the spatial and temporal characteristics of the real traces. The mean relative error of distribution of relative direction angle is 1.9%, and the mean relative error of travel schedule is 4.3%. Both of the spatial and temporal relative error is very low. Therefore, this mobility model can be employed to generate synthetic data for a long time to do some searches, such as the longest average network connectivity or the shortest average group distance, with different travel schedules. Such kinds of experiments are costly to be realized in real world.

For future work, we intend to further study the motions of these nodes not only in common condition but also in some accidental scenarios.

Acknowledgments

We would like to make a grateful acknowledgement for Don Towsley. Don gave many suggestions on this research. Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence.

References

[1] Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. Mobile Computing (1996)

[2] Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless Comm. and Mobile Computing (WCMC) 2(5), 483–502 (2002)

[3] Kotz, D., Essien, K.: Analysis of a campus-wide wireless network. In: MOBICOM 2002 (September 2002)
[4] Royer, E., Melliar-Smith, P.M., Moser, L.: An analysis of the optimum node density for ad hoc mobile networks. In: Proceedings of the IEEE International Conference on Communications (ICC) (2001)
[5] Davies, V.: Evaluating Mobility Models Within an Ad hoc Network. Master's thesis, Colorado School of Mines (2000)
[6] Song, L., Kotz, D., Jain, R., He, X.: Evaluating location predictors with extensive Wi-Fi mobility data. In: Proc. IEEE INFOCOM 2004 (2004)
[7] Kim, M., Kotz, D., Kim, S.: 2006. Extracting a Mobility Model from Real User Traces. In: Proc. IEEE INFOCOM (2006)
[8] Yoon, J., Noble, B.D., Liu, M., Kim, M.: Building Realistic Mobility Models from Coarse-Grained Traces. In: MobiSys 2006, June, pp. 177–190 (2006)
[9] Schwab, D., Bunt, R.: Characterising the Use of a Campus Wireless Network. In: Proc. IEEE INFOCOM 2004 (2004)
[10] Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In: Proc. IEEE INFOCOM 2006 (2006)
[11] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of Human Mobility on the Design of Opportunitic Forwarding Algorithms. In: Proc. IEEE INFOCOM 2006 (2006)
[12] Tuduce, C., Gross, T.: A Mobility Model Based on WLAN Traces and its Validation. In: Proc. IEEE INFOCOM 2005 (2005)
[13] Garetto, M., Leonardi, E.: Analysis of Random Mobility Models with PDE's. In: Proc. IEEE MobiHoc 2006 (May 2006)
[14] Zhang, X., Kurose, J., Levine, B.N., Towsley, D., Zhang, H.: Study of a Bus-based Disruption-Tolerant Network:Mobility Modeling and Impact on Routing. In: Proc. IEEE MobiCom 2007 (2007)
[15] Lelescu, D., Kozat, U.C., Jain, R., Balakrishnan, M.: Model T++: An Empirical Joint Space-Time Registration Model. In: Proc. IEEE MobiHoc 2006 (May 2006)
[16] Aschenbruck, N., Gerhards-Padilla, E., Gerharz, M., Frank, M., Martini, P.: Modelling Mobility in Disaster Area Scenarios. In: MSWiM 2007 (October 2007)

Measuring Energy-Time Efficiency of Protocol Performance in Mobile Ad Hoc Networks

Ida Pu[1], Yuji Shen[2], and Jinguk Kim[1]

[1] Department of Computing, Goldsmiths
University of London, London SE14 6NW, UK
[2] School of Medicine, University of Birmingham
Birmingham B15 2TT, UK

Abstract. This paper introduces two new metrics for assessment of mobile ad hoc network performance in terms of energy-time efficiency. The combined effect of both energy and time consumption is considered and represented in mathematical terms. The measures have demonstrated a number of advantages over the conventional ones in which the energy and time were often considered separately. The proposed new metrics are simple, generic and flexible. As an application, we have compared the energy-time efficiency of Blocking Expanding Ring Search (BERS) and Expanding Ring Search (ERS), two similar Time to Live (TTL)-based expanding ring search algorithms using our new metrics. The results show that the new metrics can be applied efficiently in assessment of different protocols.

Keywords: Energy-time, efficiency, metric, algorithm.

1 Introduction

Energy efficiency is one of the fundamental issues for Mobile ad hoc networks (MANETs) and has drawn attentions of researchers in recent years [1,2]. Many research papers are concentrated on design of energy efficient protocols [1,2,3] and others dedicated to the energy efficiency analysis [4,5,6]. Characteristics of certain aspects of energy waste have been identified. It is known, for example, that the communication between nodes consumes substantial amount of energy in wireless networks [7]. Many challenges, however, are still ahead due to high dependency on the cooperative ad hoc environment, the variety of the physical elements involved in MANETs and the complex requirements to different network layers and interfaces.

One interesting phenomenon is that the energy saving does not come for free. It is often a tradeoff between the amount of energy saved for completion of a task and the extra time it may take. The saving is usually achieved on the cost of taking a longer time [6,8].

Despite importance, few dedicated measures are used in MANETs to describe the energy-time tradeoff in a directly quantitative means. Most discussions address the issues of energy saving and time latency separately. The combined effect, however, has not been explicitly addressed. This often causes inconvenience

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 475–486, 2008.
© Springer-Verlag Berlin Heidelberg 2008

in assessment of different protocols or for exploring the insight of a specific approach on energy-time efficiency. Consider a simple example of comparison of two systems. Suppose that system A achieves 'energy saving of 40% with 15% time delay', while system B '35% saving with 5% time delay'. Which one is better in terms of the energy-time efficiency and how much better?

The energy consumption of a real MANET can be far more complicated to determine than this question. Nodes in MANETs consist of various mobile computer devices. They may come from different manufacture in different decades, equipped with different operating systems, under different energy management schemes, yet they need to cooperate to forward packets and to maintain a live network as long as possible. This means that neither the amount of energy saving nor the length of the time taken alone is sufficient for assessment purpose. In real time applications, time factor may play a more important role than the energy, and vice versa in energy constraint applications. In order to address the energy efficiency issues, we need to find a combined measure that takes into consideration of both the energy consumption and the time required, and it should be scalable, genetic, simple, and allow a better understanding of the energy consumption problems in MANETs.

In this paper, we consider the energy consumption problems from a new angle by considering a combined effect, described as *cost*, to capture the tradeoff nature of the amount of energy consumption and the time required to complete a task. Two simple metrics are proposed: one is referred to as *product models* and the other *trade model*, both including entities of the energy and time. We define the models and explain the mathematical and physical meanings in adoption of these measures in Sections 4 and 5. We briefly review Blocking Expanding Ring Search (BERS) and Expanding Ring Search (ERS), two Time to Live (TTL)-based ERS algorithms [8,9,11] for commonly used reactive route discovery protocols for MANETs in Section 2. We discuss the goal and experiment settings in Section 6. We demonstrate the advantages of the new metrics by presenting the analytical results for BERS and ERS in Section 7. The metrics we proposed are in fact not only simple but also genetic and flexible. We describe briefly how to extend the measures to a number of variations which can measure diverse and complex systems in Sections 4.2 and 4.3. We conclude finally that these new metrics can be a useful tool for exploring the energy efficiency issues and for assessment of different protocols of MANETs in Section 8.

2 Background and Related Work

A communication channel in MANETs can be established between two nodes consisting of a *source*, a *destination* and possibly a number of *intermediate nodes* without any fixed base station. Intermediate nodes need to cooperate to forward packets upon requests.

The route discovery process in MANETs, among many necessary activities, has been found to consume significant amount of energy [12]. The process starts from the moment when a source node broadcasts the first RREQ (Route

Request) packet until the source node receives the RREP (Route Reply) or the flooding terminates, whichever the latest.

We adopt two models in the route discovery process, namely *one-to-all* [13] multicast for broadcast and *one-to-one* unicast for transmission RREP, and assume a route cache on each mobile node. A node in MANETs broadcasts a RREQ and its one-hop neighbour nodes within the broadcast range cooperate the route discovery process by checking their own route caches for requested route information and maintaining an updated list of known routes.

An intermediate node who has the requested route information towards the destination is defined as a *route node* in this paper. If the route information is found in its route cache, the route node would stop rebroadcasting the RREQ and sends a RREP to the source node with the complete route information consisting of the cached route in itself and the accumulated route record in the RREQ. In this way, the route may be established more quickly and the total amount of delivery time and energy consumption can be reduced.

2.1 TTL-Based Expanding Ring Search

Reactive routing protocols such as DSR [10] and AODV [11] are often supported by a so-called *ERS* scheme. The goal of the ERS is to find nodes who have the valid route information to the destination node in their route cache by propagating RREQs in a controlled flooding manner.

To control the flooding in MANETs, a TTL [10] sequence is used with ERS. A pre-defined TTL number is issued with a RREQ which defines a maximum radium of a searching area by flooding. Each time when TTL is run out, the source node restarts a second-round flooding process by rebroadcasting the RREQ with an increased TTL number to allow the new RREQ to reach the nodes in further distance. There is no optimal TTL incremental sequence. The common values of the incremental TTL are 1 [14,15], and 2 [16]. For simplicity of discussion, we assume the increment of TTL value is 1, but other increments can be easily applied with similar approach.

ERS wastes energy by rebroadcast RREQs redundantly. Flooding analysis shows that rebroadcast could provide at most 60% additional coverage and only 41% on average over that already covered by the previous attempt [13].

2.2 Blocking Expanding Ring Search

The BERS is a modified ERS which achieves substantial amount of energy saving in the worst case [8]. BERS integrates, instead of TTL sequences, a newly adopted control packet, `stop_instruction` and hop number (H) to reduce the energy consumption during route discovery process.

The basic route discovery structure of BERS is similar to that of conventional TTL-based ERS. The source node, however, passes the right of reissuing RREQs to intermediate nodes on subsequent rings. Since the source node only issues a single RREQ, BERS does not resume its second round flooding from the source node when TTL fails. The new RREQs can be initialised by any appropriate

intermediate nodes in a synchronised fashion. Intermediate nodes that perform a rebroadcast on behalf of the source node or the nodes on previous ring act as an *agent* node. In this way, the energy saving is achieved.

The energy saving gained, however, does not come for free. Like in most energy efficient protocols, the energy saving is achieved on the cost of time delay.

3 Our Contribution

From next section, we introduce our cost and trade models and demonstrate how these models can be used to evaluate the energy-time efficiency of a protocol. As an application, we investigate the performance difference between BERS and ERS using the new metrics. We demonstrate our simulation results and discuss the characteristics of BERS and ERS in terms of energy-time efficiency.

Our main contribution in this paper includes: (i) introducing two simple and abstract metrics, namely cost and trade models for MANETs with justification, (ii) analysing the energy-time efficiency of BERS and ERS applying our models, (iii) demonstrating the behaviours of BERS and ERS under various node distributions.

4 Cost Models

Our initial goal is to describe the tradeoff nature of two independent but related entities, i.e. *energy* and *time* in most route discovery protocols. The new combined measure, described as *cost*, should be simple, abstract and generic. Our first metric is referred to as a *product model*.

4.1 Product Model

A cost function based on the product of the energy and time can be defined as

$$C(n) = E(n)T(n) \tag{1}$$

where n represents the size of the input data, which is task-oriented and can be, for example, the number of expanding rings, or the number of nodes. $E(n)$ represents the energy consumed and $T(n)$ the time it takes to complete a task. The cost function $C(n)$ takes two arguments $E(n)$ and $T(n)$. Obviously, the smaller the $C(n)$ is, the more energy-time efficient is the system.

The intuition of this model comes from the *moment principle of leverage* in physics as demonstrated in Figure 1. The principle of the lever tells us, in order to achieve a load balance state (the *static equilibrium*), with all forces balancing, the *moment*, i.e. the product of the weight and the distance between the load point and the fulcrum (lever pivot), on either side of the fulcrum must be equal in value. Consider an instance that two Approaches A and B are carried out to complete a task involving a number of nodes. Suppose it takes T_1 time for

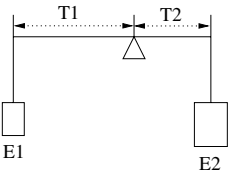


Fig. 1. Product model

Approach A to complete the task and consumes energy E_1, and T_2 time for Approach B and consumes energy E_2.

To quantify the performance of A and B, we imagine the following scenario:

We weight the energy consumptions E_1 and E_2 on a balance scale as shown in Figure 1, where the distances from the hanging positions on the left, and right, to the fulcrum, i.e. the two *moment arms*, correspond to T_1 and T_2 respectively. If the moments E_1T_1 and E_2T_2 are equal in value, the scale will show the balance. This means that the performances of Approach A and B are equally good (or bad) in terms of energy-time efficiency. There may be two possibilities: (a) If $E_1 = E_2$, then $T_1 = T_2$; (b) If $E_1 \neq E_2$, then $T_1 \neq T_2$.

If, however, the moments are different, i.e. $E_1T_1 < E_2T_2$ (or $E_1T_1 > E_2T_2$), it is possible to be one of the following four cases:

1. $E_1 < E_2$ (or $E_1 > E_2$), and $T_1 = T_2$,
   this means that Approach A (or B) is more energy efficient than Approach B (or A).
2. $E_1 = E_2$, and $T_1 < T_2$ (or $T_1 > T_2$),
   this means that Approach A (or B) is more time efficient than Approach B (or A).
3. $E_1 < E_2$ (or $E_1 > E_2$), and $T_1 < T_2$ (or $T_1 > T_2$),
   this means that Approach A (or B) is more efficient in both of energy and time than Approach B (or A).
4. $E_1 < E_2$ ($E_2 < E_1$), and $T_1 > T_2$ (or $T_2 > T_1$),
   this means that Approach A (or B) is more energy efficient but not time efficient than Approach B (or A).

The first case reflects the situation where one side is lighter in weight than the other side but both moment arms remain the same length, the scale will show imbalance and E_1 (or E_2), one side will be moving up.

The second case corresponds to the situation where the loads on both sides are of equal weight, but the moment arm (T_1) of one side is shorter than the other side, the scale will then show the imbalance, and one side (E_1) will be moving up.

The third case represents the situation where both the weight and the moment arm of one side are less than the other side in value, the scale will show the imbalance again, and E_1 side will be moving up.

The fourth case represents the situation where the weight of one side is lighter but the moment arm is longer than the other side, the scale will show balance or

imbalance depending on the moment values on both sides, i.e. whether $E_1T_1 = E_2T_2$ or $E_1T_1 \neq E_2T_2$.

Note that both energy E and time T are a function of the input size n of an algorithm. This is because, in general, the larger the input size n, the more energy will be consumed and it would usually take longer to complete a task.

This energy-time measuring model is simple but efficient, and can be generalised as given in the next section.

4.2 General Product Model

The cost function proposed in Eq.(1) is not flexible for situations where one aspect, either the energy or the time, is considered as more important than the other. For example, there may be situations where the energy is the only criteria to be measured, or time aspect is considered as more significant than the energy. We therefore modify the cost function in Eq.(1) by adding a parameter α as the power of the time factor:

$$C(n) = E(n)T^{\alpha}(n), \text{ where } \alpha \geq 0 \tag{2}$$

Here α is a positive real number. As we can see, the general cost function (2) becomes our linear cost function (1) when $\alpha = 1$. In other words, the linear cost function (1) is a special case of the general cost function (2) when $\alpha = 1$.

Similarly, when $0 \leq \alpha < 1$, the weight of the time factor is reduced, and the energy factor weights more. When $\alpha > 1$, the weight of the time factor is increased, and the energy factor weights less.

4.3 Extended General Product Model

For a more complex system, we can also consider extending E to a vector space $\mathbf{E} = (E_1, E_2, \cdots, E_m)$ and $\mathbf{T} = (T_1^{\alpha}, T_2^{\alpha}, \cdots, T_m^{\alpha})$, where m is the number of subsystems and α is a weight parameter for emphasis on T_j $(j = 1, 2, \cdots, m)$. So the cost function (1) becomes

$$C(\mathbf{E}, \mathbf{T}) = \mathbf{E}^{-1}\mathbf{T} \tag{3}$$

We have the extended general model as follows:

$$C(\mathbf{E}, \mathbf{T}) = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{pmatrix} (T_1^{\alpha}, T_2^{\alpha}, \cdots, T_m^{\alpha}) \tag{4}$$

This is equivalent to the same scenario to that for our general model except where we hang a number of objects at different positions on either side of the equilibrium.

In theory, the cost function can be tailored to suit various situations in measuring tradeoffs and to apply to different layers and different systems. For example, on the Medium Access Control (MAC) layer, energy efficient algorithm design needs to consider a number of attributes such as collision, idle listening, overhearing, radio controlling. With extended cost function (4), the comparisons of the energy efficiency can be made across the layers and different systems.

5 Trade Model

The cost models in the previous section can be used to measure the amount of energy-time tradeoff. In this section, we introduce another simple metric called *trade model* which can indicate the level of energy-time tradeoff.

The trade model captures the amount of energy saving in exchange of unit time latency. This is a normalised measure and is defined as $\Delta E/\Delta T$, where $\Delta T \neq 0$. It describes the amount of energy saving that is traded off by a unit time of latency. Given two protocols with different amounts of energy consumption E_1 and E_2 and the different lengths of time required T_1 and T_2, the amount of energy saving traded by the time latency can be calculated by

$$\frac{\Delta E}{\Delta T} = \frac{E_2 - E_1}{T_1 - T_2}$$

We assume that the first system consumes less energy but takes longer time as the cost, i.e. $E_1 < E_2$ but $T_1 > T_2$. However, as we shall see later, this condition is not essential since the equation can capture other cases such as $E_1 > E_2$ and $T_1 > T_2$.

6 Energy Efficiency of BERS and ERS

In this section, we compare the results in two cases. In the first case, we use the conventional separate measures for the energy saving and time latency. In the second case, we use our cost functions to demonstrate the advantages of the measures.

6.1 Separate Measures

We demonstrate here how the energy consumption and the time latency, despite being dependent on one to another, are measured and discussed separately.

Energy Consumption: The energy consumption can be described in the following mathematical expressions [8]. For convenience of discussion, we adopt the unit of the energy (in *UnitEnergy*) as the amount of energy consumed by a node for broadcasting an RREQ. Similarly, we define the unit of the time (in *UnitTime*) as the the amount of time for each node to wait before

rebroadcasting RREQ. The total energy consumed in BERS route discovery process can be described as:

$$\mathsf{E}_{BERS} = 2\left(1 + \sum_{i=1}^{H_r - 1} \mathsf{n}_i\right) + \mathsf{E}_{RREP} \text{ (UnitEnergy)}$$

where E_{RREP} is the amount of energy consumed for unicasting the RREP.

The total energy consumed in ERS is:

$$\mathsf{E}_{ERS} = \mathsf{H}_r + \sum_{i=1}^{H_r - 1} \sum_{j=1}^{i} \mathsf{n}_j + \mathsf{E}_{RREP} \text{ (UnitEnergy)}$$

The energy saving by using BERS is therefore:

$$\mathsf{E} \quad = \mathsf{E}_{ERS} - \mathsf{E}_{BERS} = \mathsf{H}_r - 2 + \sum_{i=1}^{H_r - 1} \left(\left(\sum_{j=1}^{i} \mathsf{n}_j\right) - 2\mathsf{n}_i\right) \text{ (UnitEnergy)}$$

Time-Delay: Again we adopt the analytical results from [8].
The total time required for BERS is:

$$\mathsf{T}_{BERS} = 3\mathsf{H}_r + 2\sum_{i=1}^{H_r} \mathsf{i} = \mathsf{H}_r^2 + 4\mathsf{H}_r \text{ (UnitTime)}$$

Similarly, the total time for ERS route discovery process is:

$$\mathsf{T}_{ERS} = 2\sum_{i=1}^{H_r} \mathsf{i} = \mathsf{H}_r^2 + \mathsf{H}_r \text{ (UnitTime)}$$

The time latency for BERS is $3H_r$, i.e.

$$\mathsf{T} \quad = \mathsf{T}_{BERS} - \mathsf{T}_{ERS} = 3\mathsf{H}_r \text{ (UnitTime)}$$

where H_r is the hop number of the first node that returns the RREP to the source node.

6.2 Comparing BERS and ERS

We now apply our models developed in previous sections and take the route discovery of the MANETs as an example to demonstrate how our cost models and trade model can be used to explore the energy-time efficiency of two algorithms: one is the conventional TTL-based ERS and the other is the BERS.

We have conducted a number of analytical simulation based on the above theoretical results and implemented in IDL 6.0 (Research Systems, Boulder, CO, USA). Our main goal is to investigate the difference between the performance of BERS and ERS in terms of energy-time efficiency. In order to gain the insight of the two algorithms, we investigate their behaviours under various node distributions as follows:

1. Uniform Distribution
   A total of 1000 nodes are placed uniformly in a geographic area covering a region with H_r of 10.
2. Pseudo-Normal Distribution
   A total of 1000 nodes are within a geographic area covering a region with H_r of 10. For each H_r covered area, 100 nodes are placed uniformly. That is: the given area is divided by 10 rings $H_r = 1, 2, \cdots, 10$, there are 100 nodes uniformly distributed within each area, i.e. 100 nodes in $area_{H_r<=1}$, another 100 in $area_{1<H_r<=2}$, ..., another 100 nodes in $area_{9<H_r<=10}$. The node density is gradually reduced from the centre along the radius to the outest ring.

We experiment with the settings for BERS and ERS and make comparison on their performances using our two models. In the general product model $C(E,T) = C(n)T^{\alpha}(n)$, we consider $\alpha = 0.5, 1$ and 2, respectively. We plot the energy-time cost against H_r in the same diagram for both BERS and ERS and try to answer the following questions: (i) Under what conditions is BERS superior than ERS in terms of energy-time saving? (ii) When time is critical, how would the answer to question (i) change? (iii) Similarly, when time is not so critical, how would the answer to question (i) differ?

7 Results

The energy-time efficiency of BERS and ERS are demonstrated in this section based on the simulation results under various node distributions using product and trade models.

7.1 Product Model

We discuss the results for the uniform and pseudo-normal node distributions separately.

Uniform Distribution. This distribution is the simplest case. It is interesting because it corresponds to an ideal situation.

Figure 2 shows the cost measured from the product model when $\alpha = 0.5$, 1 and 2. When $\alpha = 2$, two curves have no significant difference, demonstrating that the performances of two systems are nearly the same under the weight $\alpha = 2$. When $\alpha \leq 1$, the performance of BERS is significantly better than that of ERS when $H_r > 7$. It is demonstrated that low weighting should be used for the assessment of energy-time efficiency.

Pseudo-Normal Distribution. Figure 3 (left) shows the performance of BERS and ERS in the uniform distribution and (right) in the pseudo-normal distribution, both under the product model when $\alpha = 1$.

The two results are different. For the pseudo-normal distribution because there are more nodes in the centre than outside of the centre, there is more energy saving. The good performance for BERS in the pseudo-normal distribution starts at $H_r \geq 6$, while in the uniform distribution which starts at $H_r \geq 7$.

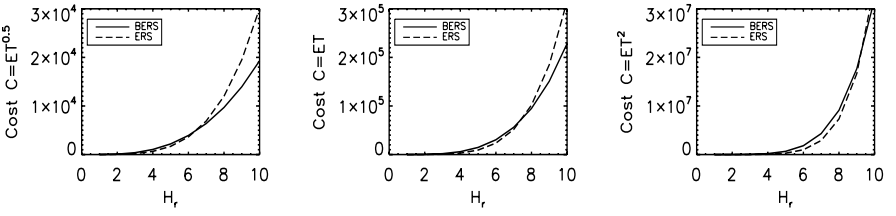


Fig. 2. Energy-time cost from product model when = 0.5, 1 and 2 (from left to right) under uniform distribution

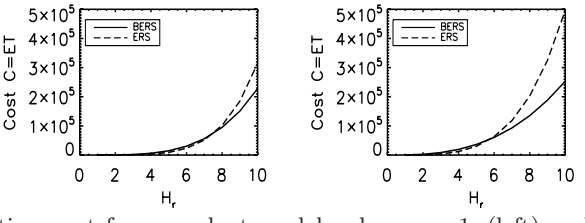


Fig. 3. Energy-time cost from product models when = 1, (left) under uniform distribution and (right) under pseudo-normal distribution

7.2 Trade Model

Figure 4 shows the behaviours of BERS in terms of the energy saving traded off by the unit time delay in comparison to that of ERS under two node distributions, namely, uniform distribution and pseudo-normal distribution.

We plot $\Delta E/\Delta T$ against H_r. As we can see in Figure 4 (left), for example, when $H_r <= 5$, BERS actually consumes more energy instead of saving any for uniform distribution. In Figure 4 (right), the threshold of H_r is approximately 4 for the case of pseudo-normal distribution.

Similarly, $\Delta T/\Delta E$, where $\Delta E \neq 0$ can be used to measure the time delay caused by unit energy saving.

This measure, although convenient, appears to have at least two disadvantages. The first disadvantage is that the measure will be invalid unless the time delay of the two systems under assessment is sufficiently different in value. The denominator ΔT (or ΔE) will approach zero in the equation otherwise. Secondly, it is not flexible enough to be used to describe the tradeoff of the energy-time efficiency.

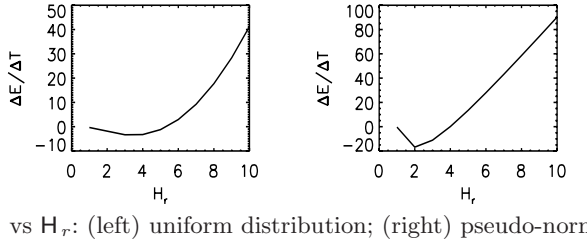


Fig. 4. E/ T vs H_r: (left) uniform distribution; (right) pseudo-normal distribution

The results are consistent to the previous ones. For example, in case of uniform distribution, while the product model shows when ($H_r = 7$) BERS starts to gain from the energy-time trade, the trade model tells us when ($H_r = 5$) the trade process actually begins. Together, the two models tell us: when $5 < H_r < 7$, although BERS starts to save energy, the saving amount is not sufficient to cover the lost of time in comparison to ERS.

8 Conclusions

We have introduced two new metrics for assessment of energy-time tradeoffs in MANETs, namely, *cost models* and *trade model* (or *product models* and *energy saving per unit-time latency model*). The product models measure the cost of combined effect of energy consumption and time delay. The trade model measures the energy saving gained from unit time delay and it tells how much an energy-time tradeoff is worth and when precisely the trade begins. The cost models have been extended and generalised for more complex systems.

We have analysed the behaviours of the BERS and ERS, two TTL-based expending ring search protocols, under different MANET node distributions applying our new measures. The energy-time tradeoffs of the two protocols are compared and evaluated using the proposed models. We found that the new measures are efficient for assessment of the energy-time tradeoffs of different systems.

With these new measures, heuristics of protocols can be further explored and more interesting quantitative questions can be answered about energy-time tradeoffs. For example, given a distribution of nodes for a geometric area, which protocol is more efficient in terms of energy-time efficiency? In contrast, these questions would not have been so easy to answer with conventional separate measures.

Our work is not restricted to measuring energy-time tradeoffs in MANETs. The approaches can be adopted to investigate other energy efficiency problems for wireless networks. In fact, the defined models can be applied for any systems that involve tradeoffs.

References

1. Garcia, J.E., Kallel, A., Kyamakya, K., Jobmann, K., Cano, J., Manzoni, P.: A novel dsr-based energy-efficient routing algorithm for mobile ad-hoc networks. In: IEEE Vehicular Technology Conference, Orlando, Florida, USA (2003)
2. Ghanem, N., Boumerdassi, S., Renault, E.: New energy saving mechanisms for mobile ad-hoc networks using olsr. In: Proc. PE-WASUN 2005, Montreal, Quebec, Canada, pp. 273–274 (October 2005)
3. van Dam, T., Langendoen, K.: An adaptive energy-efficient mac protocol for wireless sensor networks. ACM SenSys 2003 (November 2003)
4. Chen, J., Sivalingam, K., Agrawal, P.: Performance comparison of battery power consumption in wireless multiple access protocols. ACM/Baltzer Wireless Network 5(6), 445–460 (1999)

5. Hassan, J., Jha, S.: On the optimisation trade-offs of expanding ring search. In: Sen, A., Das, N., Das, S.K., Sinha, B.P. (eds.) IWDC 2004. LNCS, vol. 3326, pp. 489–494. Springer, Heidelberg (2004)
6. Sanchez-Miquel, L., Vesselinova-Vassileva, N., Barcelo, F.: Energy and delay-constrained routing in mobile ad hoc networks: an initial approach. In: Proc. PE-WASUN 2005, Montreal, Quebec, Canada, pp. 262–263 (October 2005)
7. Stemm, M., Katz, R.H.: Measuring and reducing energy consumption of network interfaces in hand-held devices. EICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications E80-B(8), 1125–1131 (1997)
8. Park, I., Kim, J., Pu, I.: Blocking expanding ring search algorithm for efficient energy consumption in mobile ad hoc networks. In: Proc. WONS 2006, Paris (2006)
9. Johnson, D.: Routing in ad hoc networks of mobile hosts. In: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1994), pp. 158–163 (December 1994)
10. Johnson, D., Maltz, D.: Dynamic source routing in ad hoc wireless networks. In: Imlellnski, T., Korth, H. (eds.) Mobile Computing, pp. 153–181. Kluwer, Dordrecht (1996)
11. Perkins, C., Royer, E.: Ad -hoc on-demand distance vector routing. In: Proceedings of The 2nd Annual IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999), New Orleans, LA, pp. 90–100 (February 1999)
12. Meghanathan, N.: Energy consumption analysis of the stable path and minimum hop path routing strategies for mobile ad hoc networks. International Journal of Computer Science and Network Security 7(10), 30–39 (2007)
13. Ni, S., Tseng, Y., Chen, Y., Shen, J.: The broadcast storm problem in a mobile ad hoc network. Wireless Networks 8(2), 153–167 (2002)
14. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceeding of the ACM Sigmetrics Conference, Mariana del Rey,CA (June 2002)
15. Royer, E.: Routing in ad-hoc mobile networks: On-demand and hierarchical strategies. Ph.D. dissertation, University of California at Santa Barbara (December 2000)
16. Perkins, C., Royer, E., Das, S.: Ad hoc on-demand distance vector (aodv) routing. IETF Request for Comment (July 2003), www.ietf.org/rfc/rfc3561.txt

A Framework for Joint Cross-Layer and Node Location Optimization in Mobile Sensor Networks

Vladimir Marbukh and Kamran Sayrafian-Pour

Information Technology Laboratory
National Institute of Standards and Technology
{marbukh,ksayrafian}@nist.gov

Abstract. This paper proposes an extension to Network Utility Maximization (NUM) framework, referred to as L-NUM (Location-aware NUM). This framework is intended to characterize both the amount of the received sensor information and the network ability to deliver the information to the intended recipient(s). The sensor location is controlled by maximization of the system utility production, which accounts for both rate of the network utility increase and the negative effects of the energy consumption as a result of sensor motion. Definition of sensor utility in L-NUM incorporates the value of sensor information which is affected by sensor locations. Once model-specific network utility and system utility production are defined, L-NUM provides intuitively appealing and tractable framework for mobile sensor network optimization.

1 Introduction

Mobile sensor networks are envisioned for detecting and tracking potential targets and events for civilian as well as military purposes. Locations of sensors in a mobile sensor networks affect both the network ability to detect and track the identified targets and events as well as the ability to communicate the relevant information to the intended recipients. The communication ability can be improved if sensors are capable of optimally self-organizing into a multihop mobile network where sensors cooperate in relaying each other information in addition to transmitting their own information. Since the detecting and tracking needs could potentially compete with the communication needs, optimal realization of mobile sensor networks requires node ability to balance these competing requirements using local, and typically incomplete, information. Energy conservation requirements could also be a major factor in controlling sensor position due to their possible impact on the sensor lifespan and in turn on the rest of the network performance. Developing self-organized mobile sensor networks capable of adjusting to target movement and/or other environmental changes requires developing sophisticated algorithms capable of balancing numerous inherent trade-offs. This paper proposes a tractable extension to Network Utility Maximization (NUM) framework aimed at addressing some of these algorithmic challenges.

NUM framework for fair bandwidth allocation in a wire-line network has been proposed in [1]. This framework assumes elastic users, sources or applications whose

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 487–496, 2008.
© Springer-Verlag Berlin Heidelberg 2008

satisfaction can be quantified by a utility function of the corresponding end-to-end bandwidth. Framework [1] assumed that elastic users/sources are capable of adjusting their bandwidth requirements in response to the network congestion. This framework has been extended to include cross-layer optimization of wire-line as well as wireless networks [2]-[4]. The extended NUM jointly optimizes flow control, routing, scheduling and power control. The optimization is achieved by a decentralized, adaptive closed-loop algorithm with feedback signals which can be interpreted as resource congestion prices.

In [5], a combination of utility-based flow model with potential field based approach to control the sensor positions has been proposed. This hybrid framework accounts for the effect of sensor locations on their ability to transmit sensor information to the intended recipient(s). This is achieved through link capacity constraints in the mobile ad-hoc network formed by the sensors. The corresponding optimal, location-dependent network utility, viewed as a potential field, defines potential forces guiding the sensors motion. However, the network utility only quantifies communication abilities of the sensor network and does not take into account the effect of sensors locations on the sensor ability to get valuable information on the target(s). Instead, the value of sensor information is incorporated through phenomenologically defined forces, e.g., "attractive forces towards goals". Also, sensor motion in [5] is guided by a mass-damper model driven by the sum of the potential and phenomenological forces with damping coefficients that represent the energy expended on the sensor motion.

Here, we propose another utility-based framework for mobile sensor network optimization, referred to as Location-aware Network Utility Maximization (L-NUM). L-NUM assumes that the aggregate sensor utility quantifies the value of the sensor information, which is a function of both sensor information rates and sensor physical locations. Given sensor locations, aggregate sensor utility maximization subject to the communication constraints yields the optimal cross-layer network design. The corresponding optimal sensor network utility quantifies both value of sensor information and ability of the network to deliver this information to the intended recipient(s). The maximum of this aggregate network utility yields the optimal sensor locations. In practice, reaching these optimal locations by mobile sensors may be infeasible due to the un-accessible terrain and/or limitations on the node energy supply. L-NUM proposes to account for these factors through utility production, which is a function of both, sensors locations and speeds. The sensor speeds are selected to maximize the utility production, given the sensor locations.

Once the model-specific network utility and system utility production are defined, L-NUM provides intuitively appealing, self-contained and tractable framework for mobile sensor network optimization. This paper describes L-NUM framework with a very brief discussion of some of the numerous methodological, computational and implementation issues. The main methodological issues include quantifying the location-dependent value of sensor information, i.e., sensor utility, as well as utility production. Computational and implementation issues include decentralized optimization based on local and typically incomplete information.

The rest of this paper is organized as follows. Section II summarizes the conventional NUM framework. Section III describes L-NUM as a natural extension of the conventional NUM by incorporating spatial effects into cross-layer

optimization and energy conservation considerations into position control. Section IV briefly illustrates L-NUM on an example of single mobile sensor transmitting information to a single receiver. Finally, Conclusion summarizes the proposed approach and outlines direction for future research.

2 Network Utility Maximization

Consider a network comprised of a set of sources S and a set of resources (i.e. links) $l \in L$ with capacities c_l. Each source $s \in S$ identifies a unique source-destination pair and a set of feasible routes R_s. Each route $r \in R_s$ is a collection of resources $l \in r$. Source s satisfaction of having end-to-end bandwidth λ is characterized by the utility function $u_s(\lambda),\ s \in S$ which is assumed to be monotonically increasing and concave in $\lambda \geq 0$. For example, widely used weighted (α, w) - fair rate allocation [6] is based on utilities

$$u_s(\lambda) = \begin{cases} w_s \dfrac{\lambda^{1-\alpha}}{1-\alpha} & if \quad \alpha \neq 1 \\ w_s \ln \lambda & if \quad \alpha = 1 \end{cases}, \tag{1}$$

where $\alpha, w_s > 0$ are parameters. When $w_s = 1$, the cases $\alpha \to 0$, $\alpha \to 1$ and $\alpha \to \infty$ correspond respectively to an allocation which achieves maximum throughput, and is proportionally fair or max-min fair.

In a link-centric formulation each source $s \in S$ with end-to-end rate λ_s is split into rates λ_r over feasible routes $r \in R_s$:

$$\lambda_s = \sum_{r \in R_s} \lambda_r \tag{2}$$

This results in the aggregate load μ_l on link $l \in L$ where

$$\mu_l = \sum_s \sum_{r: l \in r \subseteq R_s} \lambda_r \tag{3}$$

The link-centric utility maximization framework selects vector of flow rates $\Lambda = (\lambda_r, r \in R_s, s = 1,..,S)$ which maximizes the aggregate user utility

$$U_\Sigma(\lambda) = \sum_s u_s(\lambda_s). \tag{4}$$

where $\lambda = (\lambda_s, s = 1,..,S)$. This maximization is subject to the link capacity constraints $\mu_l \leq c_l$, (2) and (3).

One can also account for capacity constraints $\mu_l \leq c_l$ through congestion penalty

$$F_\Sigma(\mu, c) = \sum_l f_s(\mu_l, c_l), \tag{5}$$

where $\mu = (\mu_l, l \in L)$, $c = (c_l, l \in L)$. Penalty function $f_l(\mu_l, c_l)$ quantifies losses in terms of delays or packet loss due to buffer overflows as the link l utilization μ_l approaches link capacity c_l. Functions $f_l(\mu_l, c_l)$ are assumed to be monotonically increasing and convex in $\mu_l > 0$. A steep function $f_l(\mu_l, c_l)$ increase as μ_l approaches c_l prevents violation of the capacity constraints. NUM with capacity constraints incorporated through the congestion penalty can be expressed as follows

$$U^* = \max_{\Lambda \geq 0} \{U_\Sigma(\lambda) - F_\Sigma(\mu, c)\} \tag{6}$$

where maximization is subject to constraints (2) and (3). Such formulization and its distributed price-based solution have been proposed in [1].

While in a wire-line network link capacities c_l are typically assumed fixed, in a wireless, interference-limited network link capacities are functions of the transmission powers on neighboring links and channel conditions affecting transmission on link l as well as interference from transmissions on other links. A large number of cross-layer optimization frameworks accounting for these interactions have been proposed, e.g., see [2]-[4]. These frameworks often assume that "elastic" link capacities are given functions of the vector of average transmission powers on all links ($p = (p_l, l \in L)$) i.e.:

$$c_l = c_l(p) \tag{7}$$

For example, [4] assumes

$$c_l(p) = k_1 \log[1 + k_2 SIR_l(p)], \tag{8}$$

where k_1, k_2 are constant coefficients, and the Signal-to-Interference Ratio on link $l = (i, j)$ is

$$SIR_{ij} = \frac{p_{ij}\xi_{ij}}{\eta_j + \sum_{(n,k) \neq (i,j), n \neq i, j} p_{nk}\xi_{nj}} \tag{9}$$

In (9) ξ_{ij} is the path loss on link (i, j), and η_j is the noise power at the receiver of node j.

Elasticity of the link capacities in a wireless network naturally lead to the following NUM formulation:

$$U^* = \max_{\Lambda \geq 0} \{U_\Sigma(\lambda) - F_\Sigma(\mu, c)\} \tag{10}$$

with maximization to be subject to capacity constraints (2)-(3), wireless channel model (7), and possibly power constraints

$$p \in P \tag{11}$$

where P is the feasible power region. Much more sophisticated versions of NUM could also include optimization over packet scheduling on different links [2]-[4].

3 Location-Aware NUM

Joint Cross-Layer and Node Location Framework Model

Here, we propose a location-aware extension of NUM for mobile sensor networks by assuming that the aggregate value of the information gathered by S sensors can be quantified by the utility function $U_\Sigma(\lambda, x)$, where vectors $\lambda = (\lambda_1,..,\lambda_S)$, $x = (x_1,..,x_S)$ describe information collection rates λ_s and physical locations (coordinates) x_s of all sensors $s = 1,..,S$.

We assume that the aggregate utility $U_\Sigma(\lambda, x)$ is additive:

$$U_\Sigma(\lambda, x) = \sum_s U_s(\lambda_s, x) \tag{12}$$

where utility (i.e. information value) of each sensor $s = 1,..,S$ information is the following product

$$U_s(\lambda_s, x) = u_s(\lambda_s) v_s(x). \tag{13}$$

The first multiplier $u_s(\lambda_s)$ is an increasing and concave function of the information collection rate λ_s, e.g., of form (1). The second multiplier $v_s(x)$ quantifies the effects of the physical locations of all S sensors $x = (x_1,..,x_S)$ on the value of information captured by the sensor s. The dependence of $v_s(x)$ on the physical locations of all S sensors $x = (x_1,..,x_S)$ can be explained as follows. While the value of the information collected by a single sensor s from the intended target(s) depends on the sensor physical location x_s relative to the target(s), this value can be reduced if other sensors are located close to sensor s due to redundancy of the obtained information. In a situation when all S sensors $s = 1,..,S$ gather information from a single target, it is natural to assume that utilities $v_s(x)$ depend on the target location x_T: $v_s(x) = v_s(x|x_T)$

Physical location of mobile sensors $x = (x_1,..,x_S)$ also affects the quality of wireless channels between different sensors and between sensors and the intended recipient(s) of the sensor information. This is modeled by considering that capacity c_{ij} of the wireless link $l=(i,j)$ depends on the locations of sensors i, j (i.e. x_i, x_j):

$$c_{ij} = c_{ij}(p_{ij}, x_i, x_j) \tag{14}$$

In particular, the path loss component in (9) is a function of the locations of the two communicating sensors i.e.

$$\xi_{ij} = \xi_{ij}(x_i, x_j) \tag{15}$$

For example, in case of free-space propagation [7]:

$$\xi_{ij} = \chi_{ij}\rho_{ij}^{-\gamma} \tag{16}$$

where χ_{ij} and γ are positive constants, and $\rho_{ij} = \rho(x_i, x_j)$ is the physical distance between sensors i and j with physical coordinates x_i and x_j respectively.

As a result of these spatial effects, the optimal network utility (10) is a function of the vector of sensor locations $x = (x_1,..,x_S)$

$$U^*(x) = \max_{\Lambda, P\geq 0}\{U_\Sigma(\lambda, x) - F_\Sigma[\mu, c(p,x)]\} \tag{17}$$

where the maximization is subject to capacity constraints (2)-(3) and power constraints (11). For the case of a single target and destination, optimal utility (17) depends on the target and destination locations x_T and x_D respectively, i.e. $U^*(x) = U^*(x|x_T, x_D)$.

Location Optimization

For given sensor locations $x = (x_1,..,x_S)$, the cross-layer optimized utility can be obtained by solving equation 17. The optimal sensor locations $x^{opt} = (x_1^{opt},..,x_S^{opt})$ maximize this cross-layer optimal utility by:

$$x^{opt} = \arg\max_{x_s \in A_s} U^*(x) \tag{18}$$

where A_s is the allowable (or feasible) area for sensor s. Terrain information including unreachable or undesirable locations can be incorporated here.

Sensor motion (i.e. trajectory) should also take into account the corresponding energy consumption. To account for the "cost" of sensor s motion, we introduce a dissipative function $\varphi_s(x_s, \dot{x}_s)$ which quantifies negative effect of energy supply depletion as a result of sensor s motion with speed $\dot{x}_s$ at location x_s. Functions $\varphi_s(x_s, \dot{x}_s)$ are assumed to be monotonically increasing and convex in $\dot{x}_s$. Also, $\varphi_s(x, \dot{x}_s) > 0$ if $\dot{x}_s \neq 0$ and $\varphi_s(x_s, \dot{x}_s) = 0$ if $\dot{x}_s = 0$.

We assume that the total "cost" of sensor motion is additive:

$$\Phi(x, \dot{x}) = \sum_s \varphi_s(x_s, \dot{x}_s) \tag{19}$$

where $\dot{x} = (\dot{x}_1,..,\dot{x}_S)$ is the vector of sensors velocities. Now consider the effect of sensor s motion on the system performance. The rate of network utility change due to the sensor motion is

$$\dot{U} = \dot{x}\nabla_x U^*(x) = \dot{x}\left(\sum_s \nabla_x U_s^*(x) - \sum_l \nabla_x f_l^*\right) \tag{20}$$

where $\nabla_x = (\partial/\partial x_1,..,\partial/\partial x_n)^T$. Functions $U_s^*(x) = U_s[\lambda^*(x), x]$ and $f_l^*(x) = f_s[\mu^*(x), x]$ in (2) are calculated at the optimum (17). Expression (20) implies that cross-layer optimization (17) is performed at much faster time scale than sensors change their locations.

Now, define system utility production $W(x,\dot{x})$ as

$$W(x,\dot{x}) = \dot{U}(x) - \Phi(x,\dot{x}) \tag{21}$$

where network utility production $\dot{U}$ is given by (20) and dissipative function $\Phi(x,\dot{x})$ is given by (19). We propose to control sensor position by selecting sensor velocity vector $\dot{x}$, which maximizes the utility production (21):

$$\dot{x} = \arg\max_{\dot{x}} W(x,\dot{x}) \tag{22}$$

Interpreting (22) as a dynamic system, one can realize that since $\Phi(x,\dot{x})\big|_{\dot{x}=0} \equiv 0$, the optimal sensor location x^{opt} is an equilibrium point of this dynamic system. It is clear from (21)-(22) that the optimal sensor motion depends on both, potential $U(x)$ and the nature of the friction affecting the dissipative function $\Phi(x,\dot{x})$. For brevity, we only consider two particular cases of static and viscous friction. We assume that $x_s = (x_{sm})$ are Cartesian coordinates of sensor s with components x_{sm}.

In the case of static friction, sensor s dissipative function is

$$\varphi_s(x,\dot{x}_s) = \sum_m a_{sm}(x)\left|\dot{x}_{sm}\right| \tag{23}$$

and in the case of viscous friction, sensor s dissipative function is

$$\varphi_{sm}(x,\dot{x}_{sm}) = (1/2)a_{sm}(x_s)(\dot{x}_{sm})^2 \tag{24}$$

where positive functions $a_{sm}(x) > 0$ characterizes the "difficulty" of moving sensor s at the direction of the dimension m at the point $x_s = (x_{sm})$. It is easy to see that for static friction (23), sensor s either holds its position x_s if the static friction is sufficiently strong or moves at the highest allowable speed otherwise.

In the case of viscous friction (24), the dynamic system (22) takes the following form:

$$\dot{x}_{sm} = \frac{1}{a_{sm}(x_s)} \nabla_{x_{sm}} (\textstyle\sum_s U_s^*(x) - \sum_l f_l^*(x)) \tag{25}$$

Sensor motion (25) balances change in the value of sensor information, represented by the term $\nabla_{x_{sm}} \sum_s U_s^*(x)$, with the change in the sensor ability to deliver this information to the intended recipient, represented by $\nabla_{x_{sm}} \sum_l f_l^*(x)$. Increase in the

friction force represented by friction coefficient $a_{sm}(x)$ causes sensor to slow down. In practical situations one may expect a combination of static and viscous friction effects.

4 Example: A Single Mobile Sensor

Consider a single mobile sensor collecting information from a single target and transmitting this information to a single destination. In this case the network utility takes the following form:

$$U(\lambda, p, x) = u(\lambda)v(x) - f[\lambda, c(p, x)] \tag{26}$$

We assume that power constraints (18) impose upper bound on the average transmission power p. It is easy to see that under natural assumptions utility is maximized for the maximum allowable power p. Therefore, power p can be assumed to be fixed in (26). Formal differentiation of the joint utility function (26) with respect to λ yields the following first-order cross-layer optimality conditions:

$$v(x)u'_{\lambda}(\lambda) = f'_{\lambda}(\lambda, c) \tag{27}$$

We consider weighted (α, w) fair rate allocation utility (1) for which

$$u'(\lambda) = w\lambda^{-\alpha}; \tag{28}$$

We also consider the following penalty function associated with the communication capacity constraints:

$$f(\lambda) = \frac{\theta^{-1}}{c - \lambda} \tag{29}$$

Parameter θ represents the maximum tolerable communication delay. Equation (29) naturally arises from expression $1/(c-\lambda)$ for the average delay in $M/M/1$ queuing system [8]. Combining equations (27)-(29), we obtain the following first-order cross-layer optimality conditions:

$$\lambda^{\alpha} / [c(x, p) - \lambda]^2 = w\theta v(x) \tag{30}$$

Equation (30) has a single solution

$$\lambda = \lambda^*(x, p) \tag{31}$$

which is a function of both, sensor location x and transmission power p. Since sensor utility $U(\lambda, x)$ is an increasing function of λ, the optimal sensor location

$$x^{opt}(p) = \arg\max_{x \in A} \lambda^*(x, p) \tag{32}$$

which maximizes sensor information rate also maximizes the utility. In (32), A is the feasible region for the mobile sensors. The optimal sensor motion is characterized by the equation (22).

In some cases, function (32) can be explicitly identified. For example, in the case of α=0 :

$$\lambda = c(x, p) - [w\theta v(x)]^{-1/2} \tag{33}$$

and, in the case of α=2 :

$$\lambda = \frac{\sqrt{w\theta v(x)}}{1+\sqrt{w\theta v(x)}} c(x, p) \tag{34}$$

Now, consider the situation of low power transmissions: $p \to 0$, when

$$c(x, p) = c_0(x)p + o(p) \;\; as \;\; p \to 0 \tag{35}$$

where $c_0(x) > 0$. For example, with the channel capacity expressions in (8)-(9):

$$c_0(x) = k\,\xi(x, x_D)/\eta\,, \tag{36}$$

where k is a constant coefficient, $\xi(x, x_D)$ is the path loss from the sensor location x to the destination location x_D , and η is the noise power at the destination location x_D . We also assume that the sensor tracks a single target with location x_T , and the spatial component of sensor information value depends on both, sensor and target locations: $v(x) = v(x, x_T)$. Under these assumptions equations (33) and (34) take the following forms respectively:

$$\lambda = (k/\eta)p\xi(x, x_D) - [\theta w v(x, x_T)]^{-1/2} \tag{37}$$

$$\lambda = (k/\eta)p\xi(x, x_D)\frac{\sqrt{\theta w v(x, x_T)}}{1+\sqrt{\theta w v(x, x_T)}} \tag{38}$$

It is reasonable to assume that the spatial component of the sensor utility $v(x, x_T)$ is qualitatively similar to the path loss from the target to the sensor $\xi(x_T, x)$. Also, for simplicity, we assume

$$v(x, x_T) = \beta\xi(x_T, x) \tag{39}$$

where $\beta > 0$ is some coefficient, then equations (37) and (38) take the following forms respectively:

$$\lambda = (k/\eta)p\xi(x, x_D) - [\beta\theta w\xi(x_T, x)]^{-1/2} \tag{40}$$

$$\lambda = (k/\eta)p\xi(x, x_D)\frac{\sqrt{\beta\theta w\xi(x_T, x)}}{1+\sqrt{\beta\theta w\xi(x_T, x)}} \tag{41}$$

Considering equations (40) and (41), the following qualitative conclusions can be driven. The optimal sensor location $x = x^{opt}$, which maximizes (40) or (41), is determined by the trade-off between path loss from the target to the sensor $\xi(x_T, x)$ and from the sensor to the destination $\xi(x, x_D)$. The optimal sensor location $x = x^{opt}$ depends on the terrain through the path loss. Increase in the transmission power p moves the optimal sensor location $x = x^{opt}$ "closer" to the target and "farther" from the destination since power increase enhances communication and allows sensor to concentrate on obtaining information from the target.

5 Conclusion and Future Research

This paper has proposed a framework for self-organization of mobile sensor networks, which includes cross-layer network optimization as well as controlling sensors position. Given sensor locations, cross-layer network optimization allocates resources and configures protocols to ensure delivering the highest utility of the sensor information to the intended recipient(s). Controlling sensor location further enhances this utility.

Future efforts should address numerous research and implementation challenges, including quantification of sensor utility and utility production. Also, a simulation platform is currently under construction to further evaluate the performance of such networks in case of large number of nodes.

References

1. Kelly, F.P., Maulloo, A.K., Tan, D.H.K.: The rate control for communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society 409, 237–252 (1998)
2. Chiang, M.: Balancing transport and physical layers in wireless multihop networks:jointly optimal congestion control and power control. IEEE J. Sel. Areas Comm. 23, 104–116 (2005)
3. Lin, X., Shroff, N.B., Srikant, R.: A tutorial on cross-layer optimization in wireless networks. IEEE J. Sel. Areas Comm. 24(8), 1452–1463 (2006)
4. Chiang, M., Low, S.H., Calderbank, A.R., Doyle, J.C.: Layering as optimization ecomposition: A mathematical theory of network architectures. Proceedings of the IEEE 95(1), 255–312 (2007)
5. Popa, D.O., Stephanou, H.E., Helm, C., Sanderson, A.C.: Robotic Deployment of Sensor Networks Using Potential Fields. In: ICRA 2004, pp. 642–647 (2004)
6. Mo, J., Walrand, J.: Fair end-to-end windo-based admission control. IEEE/ACM Trans. on Networking 8, 556–567 (2000)
7. Rappaport, T.S.: Wireless Communications: Principles & Practice. Prentice Hall, Inc., Englewood Cliffs (1996)
8. Kleinrock, L.: Queueing Systems. Computer Applications, vol. II. John Willey (1976)

Author Index

Abu-Ghazaleh, Nael B. 58, 251
Acampora, Anthony S. 230
Aïache, Hervé 391
Al Hamra, Anwar 189
Altan, Nicola 15
Altman, Eitan 122
Aron, Felix 357

Barakat, Chadi 189
Benenson, Zinaida 279
Bermond, Jean-Claude 204
Bernhard, Pierre 122
Bestehorn, Markus 279
Blondia, Chris 94
Braem, Bart 94
Braun, Torsten 72
Bryce, Ciarán 439
Buchmann, Erik 279

Cavalli, Ana 345
Chan, H. Anthony 321
Chelius, Guillaume 29
Chen, Yung-chih 463
Choi, Jaeyoung 189
Conan, Vania 391

Darties, Benoît 451
De Cleyn, Peter 94
De Soete, Marijke 94
Derhab, Abdelouahid 401
Durand, Sylvain 451
Dziong, Zbigniew 293

Elkin, Michael 425
Endler, Markus 29

Freiling, Felix C. 279

Gagnon, François 293
Gansterer, Wilfried N. 369
Gołebiewski, Zbigniew 241
Gomes, Antônio Tadeu A. 29

Hamam, Yskandar 357
Han, SeonYeong 251
Haring, Günter 369
Heurtefeux, Karel 218
Hurni, Philipp 72

Ip, Louisa Pui Sum 230

Jawurek, Marek 279

Kareem, Tope R. 321
Kastrinogiannis, Timotheos 307
Khan, Majid I. 369
Kik, Marcin 333
Kim, Jinguk 475
Kosek, Katarzyna 380
Kouraogo, Pegdwindé Justin 293
Kranakis, Evangelos 1, 108
Kunz, Thomas 43

Lando, Yuval 425
Latré, Benoît 94
Lebrun, Laure 391
Leone, Pierre 148
Leung, Ian 463
Lima, Luciana S. 29
Liò, Pietro 463
López Villafuerte, Freddy 162
Lu, Xiaofeng 463

Majcher, Krzysztof 241
Mallouli, Wissam 345
Marbukh, Vladimir 487
Matthee, Karel 321
Mbarushimana, C. 265
Moerman, Ingrid 94
Moraru, Luminita 148

Natkaniec, Marek 176, 380
Nikoletseas, Sotiris 148
Ntlatlapa, Ntsibane 321, 357
Nutov, Zeev 86, 425

Odhiambo, Marcel 357
Olwal, Thomas 357

Pach, Andrzej R. 176, 380
Palaysi, Jérôme 451

Papavassiliou, Symeon 307
Paquette, Michel 108
Peeters, Michael 94
Pelc, Andrzej 108
Peleg, David 135
Preneel, Bart 94
Pu, Ida 475

Qin, Liang 43

Rathgeb, Erwin P. 15
Razak, Saquib 58
Roditty, Liam 135
Rolim, Jose 148
Rousseau, Stéphane 391

Sayrafian-Pour, Kamran 487
Sbai, Mohamed Karim 189
Schiller, Jochen 162
Segal, Michael 425
Shahrabi, A. 265
Shen, Yuji 475
Shpungin, Hanan 425
Silva, Alonso 122
Singelée, Dave 94
Sun, Jun-Zhao 413
Szott, Szymon 176

Tsiropoulou, Eirini-Eleni 307
Turletti, Thierry 189

Valois, Fabrice 218
van Wyk, Barend J. 357

Wehbi, Bachar 345
Wiese, Andreas 1

Xiong, Zhang 463

Yu, Min-Li 204

Zagórski, Filip 241
Ziviani, Artur 29

Lecture Notes in Computer Science

Sublibrary 5:
Computer Communication Networks and Telecommunications

Vol. 5198: D. Coudert, D. Simplot-Ryl, I. Stojmenovic (Eds.), Ad-hoc, Mobile and Wireless Networks. XII, 498 pages. 2008.

Vol. 5127: D. Hausheer, J. Schönwälder (Eds.), Resilient Networks and Services. XII, 217 pages. 2008.

Vol. 5067: S.E. Nikoletseas, B.S. Chlebus, D.B. Johnson, B. Krishnamachari (Eds.), Distributed Computing in Sensor Systems. XVIII, 552 pages. 2008.

Vol. 5031: J. Harju, G. Heijenk, P. Langendörfer, V.A. Siris (Eds.), Wired/Wireless Internet Communications. XII, 225 pages. 2008.

Vol. 4982: A. Das, H.K. Pung, F.B.S. Lee, L.W.C. Wong (Eds.), NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. XXII, 945 pages. 2008.

Vol. 4979: M. Claypool, S. Uhlig (Eds.), Passive and Active Network Measurement. XI, 234 pages. 2008.

Vol. 4913: R. Verdone (Ed.), Wireless Sensor Networks. XIII, 388 pages. 2008.

Vol. 4866: S. Fdida, K. Sugiura (Eds.), Sustainable Internet. XIII, 257 pages. 2007.

Vol. 4864: H. Zhang, S. Olariu, J. Cao, D.B. Johnson (Eds.), Mobile Ad-Hoc and Sensor Networks. XVII, 869 pages. 2007.

Vol. 4852: J. Janssen, P. Prałat (Eds.), Combinatorial and Algorithmic Aspects of Networking. VIII, 149 pages. 2007.

Vol. 4837: M. Kutyłowski, J. Cichoń, P. Kubiak (Eds.), Algorithmic Aspects of Wireless Sensor Networks. X, 163 pages. 2008.

Vol. 4793: G. Kortuem, J. Finney, R. Lea, V. Sundramoorthy (Eds.), Smart Sensing and Context. X, 301 pages. 2007.

Vol. 4787: D. Krishnaswamy, T. Pfeifer, D. Raz (Eds.), Real-Time Mobile Multimedia Services. XII, 197 pages. 2007.

Vol. 4786: D. Medhi, J.M. Nogueira, T. Pfeifer, S.F. Wu (Eds.), IP Operations and Management. XII, 201 pages. 2007.

Vol. 4785: A. Clemm, L.Z. Granville, R. Stadler (Eds.), Managing Virtualization of Networks and Services. XIII, 269 pages. 2007.

Vol. 4773: S. Ata, C.S. Hong (Eds.), Managing Next Generation Networks and Services. XIX, 619 pages. 2007.

Vol. 4745: E. Gaudin, E. Najm, R. Reed (Eds.), SDL 2007: Design for Dependable Systems. XII, 289 pages. 2007.

Vol. 4725: D. Hutchison, R.H. Katz (Eds.), Self-Organizing Systems. XI, 295 pages. 2007.

Vol. 4712: Y. Koucheryavy, J. Harju, A. Sayenko (Eds.), Next Generation Teletraffic and Wired/Wireless Advanced Networking. XV, 482 pages. 2007.

Vol. 4686: E. Kranakis, J. Opatrny (Eds.), Ad-Hoc, Mobile, and Wireless Networks. X, 285 pages. 2007.

Vol. 4685: D.J. Veit, J. Altmann (Eds.), Grid Economics and Business Models. XII, 201 pages. 2007.

Vol. 4581: A. Petrenko, M. Veanes, J. Tretmans, W. Grieskamp (Eds.), Testing of Software and Communicating Systems. XII, 379 pages. 2007.

Vol. 4572: F. Stajano, C. Meadows, S. Capkun, T. Moore (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. X, 247 pages. 2007.

Vol. 4549: J. Aspnes, C. Scheideler, A. Arora, S. Madden (Eds.), Distributed Computing in Sensor Systems. XIII, 417 pages. 2007.

Vol. 4543: A.K. Bandara, M. Burgess (Eds.), Inter-Domain Management. XII, 237 pages. 2007.

Vol. 4534: I. Tomkos, F. Neri, J. Solé Pareta, X. Masip Bruin, S. Sánchez Lopez (Eds.), Optical Network Design and Modeling. XI, 460 pages. 2007.

Vol. 4517: F. Boavida, E. Monteiro, S. Mascolo, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 382 pages. 2007.

Vol. 4516: L.G. Mason, T. Drwiega, J. Yan (Eds.), Managing Traffic Performance in Converged Networks. XXIII, 1191 pages. 2007.

Vol. 4503: E.M. Airoldi, D.M. Blei, S.E. Fienberg, A. Goldenberg, E.P. Xing, A.X. Zheng (Eds.), Statistical Network Analysis: Models, Issues, and New Directions. VIII, 197 pages. 2007.

Vol. 4479: I.F. Akyildiz, R. Sivakumar, E. Ekici, J.C.d. Oliveira, J. McNair (Eds.), NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. XXVII, 1252 pages. 2007.

Vol. 4465: T. Chahed, B. Tuffin (Eds.), Network Control and Optimization. XIII, 305 pages. 2007.

Vol. 4458: J. Löffler, M. Klann (Eds.), Mobile Response. X, 163 pages. 2007.

Vol. 4427: S. Uhlig, K. Papagiannaki, O. Bonaventure (Eds.), Passive and Active Network Measurement. XI, 274 pages. 2007.

Vol. 4396: J. García-Vidal, L. Cerdà-Alabern (Eds.), Wireless Systems and Mobility in Next Generation Internet. IX, 271 pages. 2007.

Vol. 4373: K.G. Langendoen, T. Voigt (Eds.), Wireless Sensor Networks. XIII, 358 pages. 2007.

Vol. 4357: L. Buttyán, V.D. Gligor, D. Westhoff (Eds.), Security and Privacy in Ad-Hoc and Sensor Networks. X, 193 pages. 2006.

Vol. 4347: J. López (Ed.), Critical Information Infrastructures Security. X, 286 pages. 2006.

Vol. 4325: J. Cao, I. Stojmenovic, X. Jia, S.K. Das (Eds.), Mobile Ad-hoc and Sensor Networks. XIX, 887 pages. 2006.

Vol. 4320: R. Gotzhein, R. Reed (Eds.), System Analysis and Modeling: Language Profiles. X, 229 pages. 2006.

Vol. 4311: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks II. XI, 253 pages. 2006.

Vol. 4272: P. Havinga, M. Lijding, N. Meratnia, M. Wegdam (Eds.), Smart Sensing and Context. XI, 267 pages. 2006.

Vol. 4269: R. State, S. van der Meer, D. O'Sullivan, T. Pfeifer (Eds.), Large Scale Management of Distributed Systems. XIII, 282 pages. 2006.

Vol. 4268: G. Parr, D. Malone, M. Ó Foghlú (Eds.), Autonomic Principles of IP Operations and Management. XIII, 237 pages. 2006.

Vol. 4267: A. Helmy, B. Jennings, L. Murphy, T. Pfeifer (Eds.), Autonomic Management of Mobile Multimedia Services. XIII, 257 pages. 2006.

Vol. 4240: S.E. Nikoletseas, J.D.P. Rolim (Eds.), Algorithmic Aspects of Wireless Sensor Networks. X, 217 pages. 2006.

Vol. 4238: Y.-T. Kim, M. Takano (Eds.), Management of Convergence Networks and Services. XVIII, 605 pages. 2006.

Vol. 4235: T. Erlebach (Ed.), Combinatorial and Algorithmic Aspects of Networking. VIII, 135 pages. 2006.

Vol. 4217: P. Cuenca, L. Orozco-Barbosa (Eds.), Personal Wireless Communications. XV, 532 pages. 2006.

Vol. 4195: D. Gaiti, G. Pujolle, E.S. Al-Shaer, K.L. Calvert, S. Dobson, G. Leduc, O. Martikainen (Eds.), Autonomic Networking. IX, 316 pages. 2006.

Vol. 4124: H. de Meer, J.P.G. Sterbenz (Eds.), Self-Organizing Systems. XIV, 261 pages. 2006.

Vol. 4104: T. Kunz, S.S. Ravi (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XII, 474 pages. 2006.

Vol. 4074: M. Burmester, A. Yasinsac (Eds.), Secure Mobile Ad-hoc Networks and Sensors. X, 193 pages. 2006.

Vol. 4033: B. Stiller, P. Reichl, B. Tuffin (Eds.), Performability Has its Price. X, 103 pages. 2006.

Vol. 4026: P.B. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), Distributed Computing in Sensor Systems. XIV, 566 pages. 2006.

Vol. 4003: Y. Koucheryavy, J. Harju, V.B. Iversen (Eds.), Next Generation Teletraffic and Wired/Wireless Advanced Networking. XVI, 582 pages. 2006.

Vol. 3996: A. Keller, J.-P. Martin-Flatin (Eds.), Self-Managed Networks, Systems, and Services. X, 185 pages. 2006.

Vol. 3976: F. Boavida, T. Plagemann, B. Stiller, C. Westphal, E. Monteiro (Eds.), NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. XXVI, 1276 pages. 2006.

Vol. 3970: T. Braun, G. Carle, S. Fahmy, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 350 pages. 2006.

Vol. 3964: M.Ü. Uyar, A.Y. Duale, M.A. Fecko (Eds.), Testing of Communicating Systems. XI, 373 pages. 2006.

Vol. 3961: I. Chong, K. Kawahara (Eds.), Information Networking. XV, 998 pages. 2006.

Vol. 3912: G.J. Minden, K.L. Calvert, M. Solarski, M. Yamamoto (Eds.), Active Networks. VIII, 217 pages. 2007.

Vol. 3883: M. Cesana, L. Fratta (Eds.), Wireless Systems and Network Architectures in Next Generation Internet. IX, 281 pages. 2006.

Vol. 3868: K. Römer, H. Karl, F. Mattern (Eds.), Wireless Sensor Networks. XI, 342 pages. 2006.

Vol. 3854: I. Stavrakakis, M. Smirnov (Eds.), Autonomic Communication. XIII, 303 pages. 2006.

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3462: R. Boutaba, K.C. Almeroth, R. Puigjaner, S. Shen, J.P. Black (Eds.), NETWORKING 2005. XXX, 1483 pages. 2005.