Ioanis Nikolaidis
Kui Wu (Eds.)

LNCS 6288

Ad-Hoc, Mobile and Wireless Networks

9th International Conference, ADHOC-NOW 2010
Edmonton, AB, Canada, August 2010
Proceedings

Springer

Lecture Notes in Computer Science 6288

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
TU Dortmund University, Germany

Madhu Sudan
Microsoft Research, Cambridge, 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

Ioanis Nikolaidis   Kui Wu (Eds.)

Ad-Hoc, Mobile and Wireless Networks

9th International Conference, ADHOC-NOW 2010
Edmonton, AB, Canada, August 20-22, 2010
Proceedings

Springer

Volume Editors

Ioanis Nikolaidis
University of Alberta
Edmonton, AB, Canada
E-mail: nikolaidis@ualberta.ca

Kui Wu
University of Victoria
Victoria, BC, Canada
E-mail: wkui@cs.uvic.ca

Library of Congress Control Number: 2010931241

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

LNCS Sublibrary: SL 5 – Computer Communication Networks
and Telecommunications

ISSN 0302-9743
ISBN-10 3-642-14784-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14784-5 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.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

For more than a decade, ad hoc wireless networks have galvanized the interest and sparked the imagination of researchers. Such networks consist of a set of nodes equipped with wireless interfaces and they are designed to form self-organizing and spontaneous networks. They are usually multi-hop in nature, collectively forwarding and processing data to accomplish an application task. In addition, their topologies can be highly dynamic due to the potential mobility of the nodes. All these properties, in isolation and combination, pose a plethora of research challenges as well as new application opportunities.

The International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW) serves as one of the premier venues for researchers and industrial practitioners to exchange ideas in this exciting area. Following previous ADHOC-NOW conferences in Murcia, Spain (2009), Sophia Antipolis, France (2008), Morelia, Mexico (2007), Ottawa, Canada (2006), Cancun, Mexico (2005), Vancouver, Canada (2004), Montreal, Canada (2003), and Toronto, Canada (2002), the ninth ADHOC-NOW conference took place at the University of Alberta in Edmonton, Canada, during August 20–22, 2010. As the capital city of the Province of Alberta, Edmonton is a cultural, governmental, and educational center and offers year-round world-class festivals, including the Edmonton International Fringe Theatre Festival. Edmonton is also home to North America's largest indoor shopping mall, the West Edmonton Mall. The summer in Edmonton is particularly joyful and has mild temperature and long, sunny daytime.

The 9th ADHOC-NOW attracted 43 paper submissions, authored by researchers from 21 countries. Of the submitted papers, 16 papers were accepted as full papers after a rigorous peer-review process. The accepted papers cover topics in routing/broadcasting/multicasting protocols, energy efficiency, sensor coverage, scheduling algorithms, localization, mobility modeling, data collection and processing, and vehicular networks. We believe this coverage is broad and representative of the current research interests and activities in this area.

We would like to thank the work of the reviewers and all of the Program Committee members, as well as all the volunteers who helped in putting together an excellent program. We are grateful to the University of Alberta for its hospitality and to MITACS for its continuing support of ADHOC-NOW in Canada.

June 2010

Ioanis Nikolaidis
Kui Wu

Organization

Committees

Program Co-chairs

Ioanis Nikolaidis	University of Alberta, Canada
Kui Wu	University of Victoria, Canada

Proceedings Co-chairs

Qiang Ye	University of Prince Edward Island, Canada
Sajid Hussain	Fisk University, USA

Publicity Co-chairs

Chun Tung Chou	University of New South Wales, Australia
Cheng Li	Memorial University, Canada

Steering Committee

Evangelos Kranakis	Carleton University, Canada
Michel Barbeau	Carleton University, Canada
S. S. Ravi	SUNY Albany, USA
Ioanis Nikolaidis	University of Alberta, Canada
Violet Syrotiuk	Arizona State University, USA
Thomas Kunz	Carleton University, Canada
Ivan Stojmenovic	SITE University of Ottawa, Canada

Technical Program Committee

Nael Abu-Ghazaleh	SUNY Binghamton, USA
Michel Barbeau	Carleton University, Canada
Luciano Bononi	University of Bologna, Italy
Marcello Caleffi	University of Naples "Federico II", Italy
Jiannong Cao	Hong Kong Polytechnic University, Hong Kong
Juan Carlos Cano	Univ. Politecnica de Valencia, Spain
Jean Carle	University of Lille, France
Arnaud Casteigts	SITE University of Ottawa, Canada
Edgar Chavez	Univ. Michoacana San Nicolas de Hidalgo, Mexico
Chun Tung Chou	University of New South Wales, Australia

Costas Constantinou	University of Birmingham, UK
Sajal Das	University Texas at Arlington, USA
Falko Dressler	Univ. of Erlangen, Germany
Rasit Eskicioglu	University of Manitoba, Canada
Hannes Frey	University of Paderborn, Germany
Vasilis Friderikos	King's College London, UK
Jie Gao	Stony Brook University, USA
Xiaohua Jia	City University of Hong Kong, Hong Kong
Holger Karl	University of Paderborn, Germany
Ralf Klasing	CNRS, France
Jerzy Konorski	Gdansk University of Technology, Poland
Evangelos Kranakis	Carleton University, Canada
Thomas Kunz	Carleton University, Canada
Ivan Lequerica	Telefonica I+D, Spain
Xiang-Yang Li	Illinois Institute of Technology, USA
Xu Li	University of Ottawa, Canada
Weifa Liang	The Australian National University, Australia
Hai Liu	Hong Kong Baptist University, Hong Kong SAR
Pietro Manzoni	Univ. Politecnica de Valencia, Spain
Jelena Misic	Ryerson University, Canada
Nathalie Mitton	University of Lille, France
Marc Mosko	Palo Alto Research Center (PARC), USA
Lata Narayanan	Concordia University, Canada
Amiya Nayak	SITE University of Ottawa, Canada
Ioanis Nikolaidis	University of Alberta, Canada
Sotiris Nikoletseas	CTI and University of Patras, Greece
Jaroslav Opatrny	Concordia University, Canada
Marina Papatriantafilou	Chalmers University, Sweden
S. S. Ravi	SUNY Albany, USA
Francisco J. Ros	University of Murcia, Spain
Pedro Ruiz	University of Murcia, Spain
Juan A. Sanchez	University of Murcia, Spain
Ivan Stojmenovic	SITE University of Ottawa, Canada
Violet Syrotiuk	Arizona State University, USA
Jozef Wozniak	Gdansk University of Technology, Poland
Kui Wu	University of Victoria, Canada
Qin Xin	Simula Rserach Lab., Norway

Additional Reviewers

Gary Wong
Zhongming Zheng
Tomasz Radzik
Di Yuan
Nicholas Boers
Evandro De Souza
Israat Haque
Benny Shimony
Ryan Vogt
Marios Angelopoloulos
Krzysztof Gierlowski
Slawomir Kuklinski
Marek Natkaniec
Krzysztof Szczypiorski

Table of Contents

Simulation-Based Comparison of Three Wireless Multicast Routing Protocols: MOST, MOLSR and SMOLR

Amina Meraihi Naimi[1], Cedric Adjih[1], Pascale Minet[1],
and Georgios Rodolakis[2]

[1] Hipercom Project-Team, INRIA Paris-Rocquencourt
[2] Macquarie University

Abstract. In this paper, we study multicast protocols for Mobile Ad-Hoc Networks, and specifically OLSR networks. We describe and compare three multicast protocols, namely MOST (Multicast Overlay Spanning Tree), MOLSR (Multicast OLSR) and SMOLSR (Simple Multicast OLSR), with different approaches and properties with respect to scalability. Our approach is simulation-based: through different scenarios and configurations, we evaluate the performance of each protocol in terms of average packet delivery ratio and average packet retransmissions. We interpret the results in terms of key design properties and applicability of the protocols.

1 Introduction

Multicast is a different form of communication from the ubiquitous unicast communications used in client-server protocols (which includes HTTP, i.e. the Web). It consists in sending a stream of data from a *source* to several receiver nodes (*client*) in the network. The source and the client forms a *multicast group*. Audio or video conferencing, push-to-talk, and multimedia content distribution are examples requiring multicast communications, which in turn, require a multicast routing protocol.

In mobile ad hoc networks (MANET), different approaches have been followed for designing multicast protocols for wireless networks and the existing protocols vary depending on the group structure, the nature of transmissions, the reliance (or not) on an underlying unicast routing protocol, etc. A survey of some MANET multicast protocols with a complete classification can be found in [1] and [3]; and [4] for instance for broadcast protocols. These differences are expected to have an impact on protocol performances and the goal of this article is to highlight this impact in case of three multicast MANET protocols proposed as an extension of the OLSR unicast routing. In this article, we evaluate and compare performances of MOST (Multicast Overlay Spanning Tree), MOLSR (Multicast OLSR) and SMOLSR (Simple Multicast OLSR) through NS2 simulations. Our focus is on evidencing the different behaviors of different families of protocols rather than optimizing the performance of one family of

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 1–14, 2010.

© Springer-Verlag Berlin Heidelberg 2010

protocols, and to identify the key properties of multicast algorithms[1]. The rest of the document is organized as follows:

Section 2 presents a brief description of the three multicast protocols. In Section 3, we present a simulation-based comparison of these multicast protocols in various scenarios with different network configurations. Protocols are evaluated in terms of delivery ratio, average number of packet retransmissions for different group sizes, different rates of the multicast sources and different mobility scenarios. Finally, Section 4 discusses the adequacy of features of each family of protocols to different scenarios, taking into account the performance evaluation reported in the previous section, discusses multicast protocol design and concludes this document.

2 Multicast Protocols Description

The three studied multicast protocols are representative of different approaches to multicast in MANETs [3]:

– SMOLSR: optimized broadcast to the entire network
– MOLSR: shortest-path tree from source to every client, using *neighborcast*
– MOST: (overlay) unicast tree joining all group members

We denote *neighborcast*, the action of transmitting the same packet to several neighbors at the same time: it is the usual method for benefiting from the *wireless multicast advantage.*

The table 1 is a summary of the properties of the protocols, and the Figure 1 illustrates the outcome of three protocols on the same sample topology.

Table 1. Main properties of the protocols (OLSR assumed as unicast routing protocol)

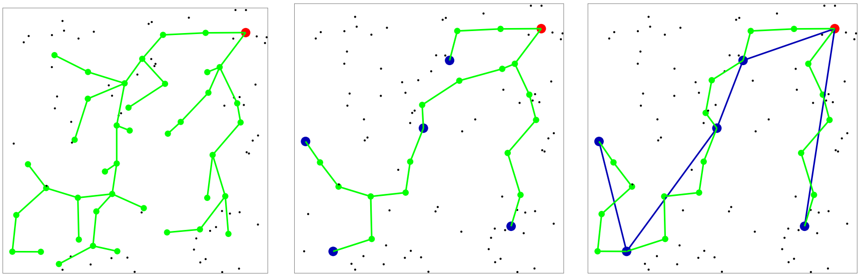
Name	Method	Transmissions	Additional Protocol Overhead
SMOLSR	broadcast	neighborcast	none (already built into OLSR)
MOLSR	source-rooted tree	neighborcast	overhead for tree creation/maintenance
MOST	spanning tree	unicast	group membership announcement

2.1 SMOLSR

SMOLSR (for Simple Multicast OLSR) [9] is a simple multicast forwarding protocol. It is an optimized flooding which uses the MPRs (Multi Point Relay) to disseminate the multicast data to the entire network[2]. Thus, the knowledge of multicast groups and membership is not required. The MPR concept used in OLSR is summarized as follows. A node selects a subset of its 1-hop symmetric neighbors that cover all the nodes that are at two hops from it. This subset

[1] One of our motivations, related to the report [7] for French MoD, is indeed to identify appropriate protocols for different military applications in different scenarios.

[2] In the IETF proposal SMF [10], it is also called “Source-based Multipoint Relay”.



(a) SMOLSR: 41 (neighborcast) transmissions

(b) MOLSR: 18 (neighborcast) transmissions

(c) MOST: 22 (unicast) transmissions - dark lines represent virtual links

Fig. 1. The three multicast protocols on the same topology (1 source and 5 clients)

is referred to as MPR set. Upon *first* receipt of a multicast packet, the node must determine whether it should forward the packet or discard it. If this node belongs to the sender's MPR set, then the packet is forwarded.

Doing so, only a subset of nodes relay the data packets: they form a connected dominating set. It belongs to the family of optimized broadcast protocols (see [4] for a survey). Note that this connected dominating set is not static nor optimal (see [4] for alternatives): it is dynamically formed, and thanks to *dynamic self-pruning*, it offers some resilience against losses.

2.2 MOLSR

MOLSR (for Multicast OLSR) [11], allows each node of a multicast group to receive the information from a multicast source. For this purpose, it maintains a tree per pair (multicast source, multicast group), taking advantage of the topology information provided by the OLSR unicast routing protocol. MOLSR builds a source-dependant tree and ensures that any multicast client is reached by the shortest path. The trees are updated whenever a change in the topology is detected. A multicast tree is built and maintained for any tuple (source, multicast group) in a distributed manner without any central entity.

Tree Building, Maintenance and Detachment. Once a source wants to send data to a specific multicast group, it sends a SOURCE_CLAIM message enabling nodes which are members of this group to detect its presence and to attach themselves to the associated multicast tree. This message is flooded within the ad hoc network using the optimized flooding technique of OLSR. Branches are built hop by hop in a backward manner as follows. When a group member receives a SOURCE_CLAIM message and it is not already a participant of this (source, multicast group) tree, it attaches itself to the tree and designates the next hop to reach the source in a shortest path as its *parent* in the multicast tree and sends a CONFIRM_PARENT message to it. The parent node receiving this message attaches itself to the (source, multicast group) tree, if it is not already a participant to this tree. The trees are periodically refreshed, by means of the SOURCE_CLAIM message and the CONFIRM_PARENT message. Notice that topology changes are still detected by the exchange of topology control messages which is done naturally by OLSR. Thus, trees updates are triggered by the detection of topology changes.

Routing Decision and Encapsulation. For any non-duplicate received packet, the MOLSR node determines whether it should forward the packet or discard it. If this node is attached to the associated tree as a *parent*, then it forwards the packet (with neighborcast), otherwise the packet is not routed.

2.3 MOST

A third multicast protocol, called MOST (Multicast Overlay minimum Spanning Tree) was proposed in [6] and operates on the OLSR protocol. MOST belongs to the category of Application-layer Multicast (ALM) protocols [2] (also called virtual multicast protocols) defined for wired networks and it inherits most of their advantages. For instance, only machines involved in multicast must be equipped with multicast capabilities unlike conventional protocols, where all the machines on the network must integrate the multicast capabilities (implement the protocols). In wireless networks, some overlay multicast protocols were proposed [1], but in contrast to these protocols, MOST was motivated by analytical results on the achievable capacity of multicast communication in ad hoc networks (the theoretical capacity bounds proved in [5]).

The MOST algorithm [5] consists in building a minimum spanning tree connecting all the clients of a given multicast group. Unlike MOLSR, MOST builds a so-called group-shared tree, which implies the existence of one single tree per multicast group, whatever the source. A branch - also called *a logical link* is a unicast path (of one or several hops) between two clients. Tunneling is performed on these links to route packets. Like MOLSR, MOST requires an underlying link state unicast routing protocol to determine unicast paths between any two nodes in the network. To proceed to the computation of the overlay tree, multicast nodes need to have knowledge of the membership of their multicast groups. A message called *most* including the list of multicast groups to which the node belongs is periodically sent to the entire network like OLSRv2 TCs (using MPR optimization).

Tree Computation. MOST operates in a distributed manner, and periodically, it computes the overlay tree for each multicast group based on the group membership and the network topology (by means of a single modified Dijkstra route computation [6]). In order to reduce losses caused by topology changes, a logical neighbor in any tree in the past is retained for a given holding time.

Routing Decision and Encapsulation. for any non-duplicate packet, MOST determines which node(s) the packet must be forwarded to. The destinations are nothing else than the set of the logical neighbors except the one which the packet was received from. The data packet is then encapsulated in a unicast UDP packet and sent in unicast to each logical neighbor.

2.4 Generic Multicast Architecture

We now present the generic architecture that is used by the three multicast protocols, for the real implementation of the protocols. This architecture, given in [7] presents the advantage of separating the topology control and the multicast structure management from the multicast data routing, here called GMF (Generic Multicast Forwarder). The different modules entering in the design of the three protocols are represented in Figure 2 with their interactions.

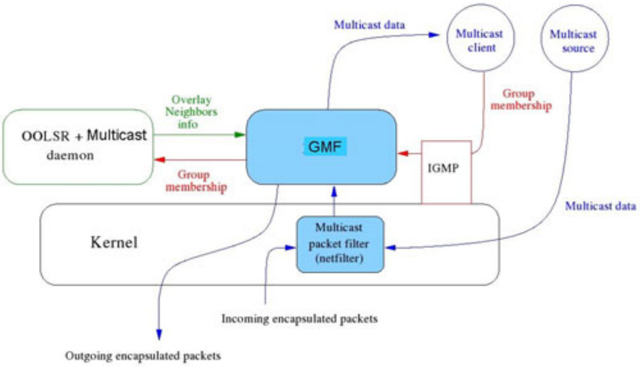


Fig. 2. Generic multicast architecture

- **OOLSR, Object Oriented OLSR**, [15], is INRIA's implementation of the OLSR protocol [13].
- **Multicast:** the module in charge of maintaining the multicast structure needed for multicast routing. This module uses control messages specific to the multicast protocol chosen. In practice, this module represents either the SMOLSR, MOLSR or MOST daemon.
- **GMF, Generic Multicast Forwarder:** this module is responsible for capturing and encapsulating multicast packets to be forwarded according to the multicast protocol chosen. For instance, multicast packets are forwarded to the entire network in case of SMOLSR, and inside a multicast tree in case of MOLSR or MOST. Forwarding rules are specified by the multicast protocol.
- **IGMP, Internet Group Management Protocol**, [16]**:** this module maintains the group membership.

The core functioning of OOLSR and Multicast described above is implemented in a shared library which is also used in NS2 within the OLSR routing agent. Thus for simulations, the architecture is identical, except that the GMF/IGMP/Kernel parts which are replaced by NS2 equivalents.

3 Performance Evaluation of Multicast Protocols

The performance evaluation of the three multicast protocols studied is done by means of simulations with NS2.

3.1 Simulation Parameters and Evaluation Criteria

Simulation parameters For the simulations, we adopt the parameters listed in Table 2:

Table 2. Simulation parameters

Run	Duration	300s
Configuration	Network area	1850m x 1850 m
	Number of nodes	200 or 100
Multicast	Group size	5 or 10 or 20
	Number of groups	1 or 2 or 3 or 4
	Number of sources	1 source per group
Mobility	Model	Random Way-point
	Maximum speed	0 or 1m/s or 5m/s or 10m/s
	Pause time	10 s
Traffic	Type	CBR
	Rate	64 or 150 or 200 or 250 kbps
	Packet size	1200 bytes
OLSR	HELLO interval	1s
	TC interval	5s
MAC	IEEE 802.11b	11Mbps
	Broadcast (neighborcast) rate	2 or 11 Mbps
PHY	Transmission range	250m
	Propagation model	Two-ray ground

Evaluation Criteria. We will evaluate the delivery ratio for the three multicast protocols considered. In order to have an estimate of the overhead generated, we will also evaluate the average number of packet retransmissions, also called average packet forwarding.

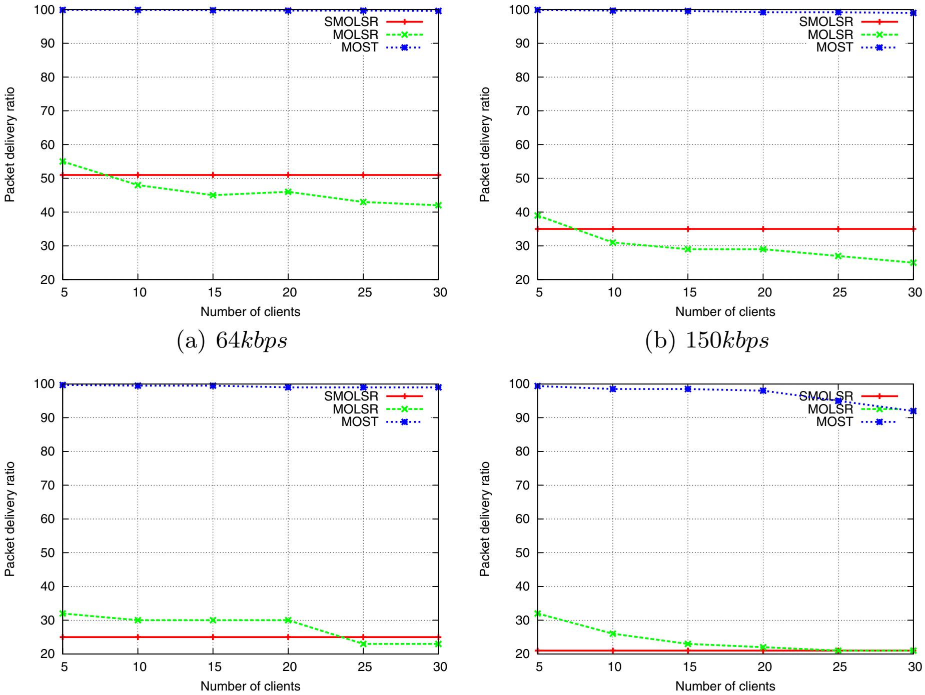
We consider different scenarios and study the impact of the:

- group size,
- number of groups,
- source rate,
- node mobility.

In order to obtain reliable results, simulations are iterated several times (7 on average) and the mean value is computed.

3.2 Packet Delivery Ratio versus Throughput

Simulations are conducted to determine the impact of the source rate on each protocol in term of packet delivery ratio. Since the goal here is to find the saturation point of the network, we consider a static topology. We consider a 200 wireless nodes network in a 1850×1850m^2 area, with one multicast group. We vary the number of clients as well as the source bit rate and evaluate the packet delivery ratio (in short *PDR*). Results are depicted in Figure 3.



(a) 64*kbps*

(b) 150*kbps*

(c) 200*kbps*

(d) 250*kbps*

Fig. 3. Comparison of multicast packet delivery ratio between the 3 protocols

As expected, MOST offers better packet delivery ratio than both MOLSR and SMOLSR in all cases (whatever the source rates or the number of multicast clients). This is explained by the use of unicast transmissions of data packets by MOST whose advantage is double. First, packets are sent at a higher rate (11*Mbps*, versus 2*Mbps* for multicast transmissions) which reduces significantly the channel occupancy. Second, packets are retransmitted when they are lost increasing the packet delivery ratio. For instance, we notice that with MOST the source node can transmit with a rate up to 200*kbps* with a very high delivery ratio. For higher transmission rates (from 250*kbps*), the packet delivery ratio remains good for small groups but decreases for large group sizes.

MOLSR leads to a maximum PDR equal to 55% which in itself is not satisfactory. This means that the channel has reached saturation although the source rate is low (64*kbps*). Moreover, this rate significantly decreases either when we increase the source rate or the group size. For instance, the PDR falls to 20% for a 250*kbps* rate with 25 clients.

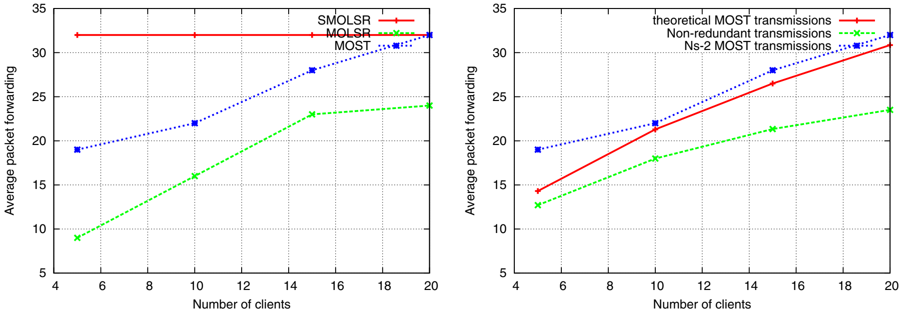
SMOLSR results are the same order of magnitude as MOLSR but some differences can be noted. For small group sizes, we notice a slight advantage for MOLSR. This is because SMOLSR floods the entire network regardless the group size which increases the global load. However, for large groups and small source rates, for example a 64*kbps* rate, SMOLSR gives better packet delivery ratio (around 40% for MOLSR and 50% for SMOLSR).

⟹ The key property (from table 1) is the nature of transmissions: unicast versus neighborcast. Overall, unicast offers smaller channel occupancy and better reliability.

3.3 Comparison of Average Packet Forwarding

By average packet forwarding we denote the average packet retransmissions in the network. The measure of average packet forwarding is useful in evaluating the impact of the multicast traffic on the global network load.

For this purpose, we consider a randomly generated topology of 100 wireless nodes forming an ad hoc network, in a area of $1500m*1500m$. We consider group sizes ranging from 5 to 20 nodes (not including the source). One source sends a traffic of 64*kbps* during 150 seconds of simulated time. We use in turn SMOLSR, MOLSR and MOST. Simulation results are depicted in Figure 4(a).



(a) Average number of retransmissions per multicast packet

(b) Comparison between MOST average number of retransmissions and Non-redundant transmissions

Fig. 4.

As we know, the main drawback of SMOLSR is to flood the entire network. As a result, with small groups, the average packet forwarding is higher than the two other protocols.

The average packet forwarding relies on three factors:

1. The number of forwarders (nodes on the tree that relay packets to the clients).
2. The ratio of received packets on each forwarder. This in turn depends on the loss probability on each branch on the tree.
3. The transmission nature i.e Unicast (like with MOST) or Multicast (like with MOLSR).

With MOST, the cost includes the redundant unicast transmissions and increases with the number of clients. On the other hand, MOLSR takes advantage of neighborcast transmissions which cost one transmission for the entire neighborhood. As a result, MOLSR provides the best packet forwarding whatever the group size. To highlight this idea, we reproduce the same scenario on a graph simulator with MOST protocol. As the packet delivery ratio is equal to almost 100% with MOST, the average packet forwarding is equal to the average number of forwarders in the MOST tree. This is depicted in Figure 4(b). First, we can notice that the average packet forwarding obtained via NS-2 and the one obtained theoretically are close. Second, we subtract the cost of redundancy due to unicast transmissions to consider its impact on the average packet forwarding for MOST. By comparing Figure 4(b) and Figure 4(a), we notice that the curve "Non-redundant transmissions" is close to MOLSR average packet forwarding.

However, the neighborcast transmission is not the only reason for reducing the packet average forwarding for MOLSR. In fact, packet loss itself reduces the number of retransmissions since not all packets reach all forwarders on the path in order to be routed. This is why, both packet delivery ratio and average packet forwarding must be considered as performance criteria.

In our scenario, MOLSR offers better average packet forwarding but with more losses, while MOST gives better packet delivery ratio, as summarized in Table 3.

Table 3. Packet delivery ratio in a 100 nodes network, and a $64kbps$ source rate

Number of clients	PDR(%) MOLSR	PDR(%) MOST
5	91	99.9
10	88	99.9
15	83	99.9
20	83	99.9

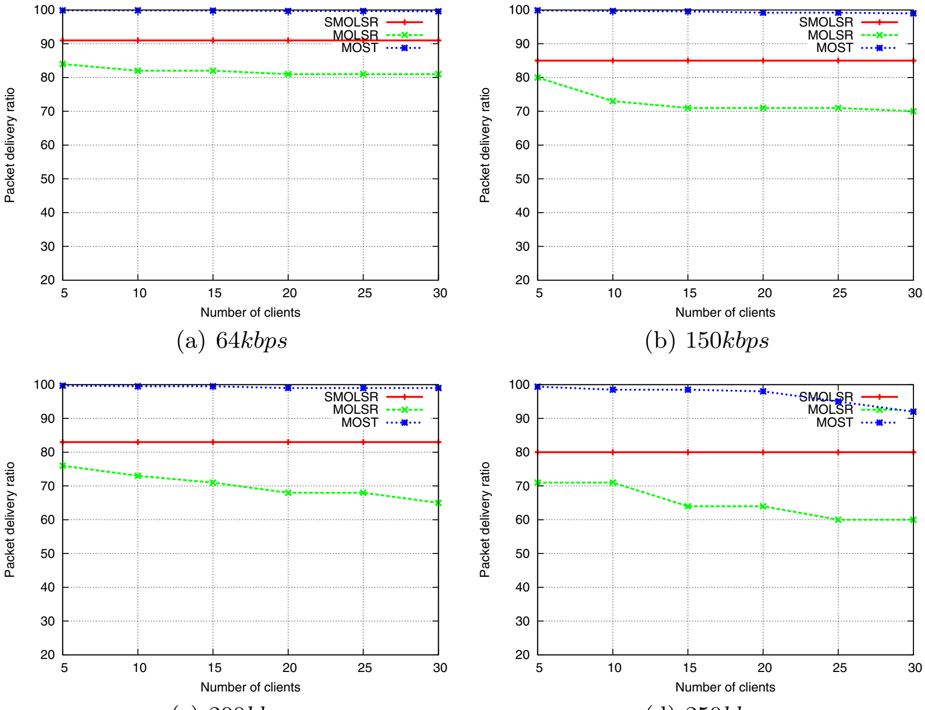
$\Longrightarrow$ **Here, the key properties (from table 1) for the number of forwarders are the nature of the transmissions (unicast vs. neighborcast) and the method (broadcast vs. multicast). It is minimized with neighborcast and multicasting.**

$\Longrightarrow$ **The key property for reliability is still the nature of transmissions: unicast is more reliable.**

3.4 Setting Neighborcast Basic Rate to 11Mbps

As we saw in previous section, MOST protocol supports higher throughputs than MOLSR and SMOLSR due to the use of unicast transmissions sent at $11Mbps$. Recall that the reason of setting the default 802.11 broadcast (neighborcast) rate to $2Mbp$s is to ensure reliability. In fact, it is important to maximize the chance of packets to be received at once since no retransmission is allowed. However, only low modulation rates are able to ensure that goal. Indeed, a high modulation rate requires an excellent signal quality otherwise it generates a lot of losses. That said, it seems to us important to compare the three protocols in the same context, i.e using the same modulation of $11Mbps$. Of course, the context of such a configuration would be an indoor network with good links quality. We set the multicast rate to $11Mbps$, and run simulations, the results are shown in Figure 5.

For MOLSR, the best packet delivery ratio is around 85% when it was only 50% with the classical $2Mbps$ broadcast (neighborcast) rate). The same impact is observed using SMOLSR with a PDR up to 90%. As we can notice, MOST still takes advantage on MOLSR and SMOLSR because although multicast packets are now transmitted with a higher rate, retransmissions are not performed like



(a) $64kbps$

(b) $150kbps$

(c) $200kbps$

(d) $250kbps$

Fig. 5. Comparison of PDRs with 11Mbps neighborcast rate

with unicast packets which has necessarily an impact on the packet delivery ratio.

=⇒ **Even when factoring out the higher data rate of unicast, the nature of transmissions was found to be decisive with better reliability of unicast even if it incurs more transmissions.**

3.5 Protocols Performance versus Number of Groups

We now run simulations by fixing the number of clients to 10 and varying the number of groups. In each group, a source is transmitting a CBR traffic with a $64kbps$ rate. We evaluate the impact of the number of groups on the packet delivery ratio. Results are illustrated in Figure 6. For MOST, a very high PDR is recorded until the number of groups reaches 8. For MOLSR, PDR decreases from 45% to 22% whereas it decreases from 51% to 14% for SMOLSR. As expected, when we increase the rate, MOLSR behaves better than SMOLSR.

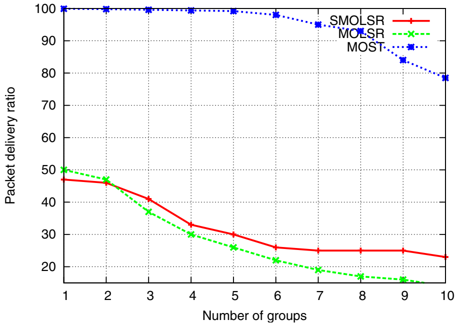
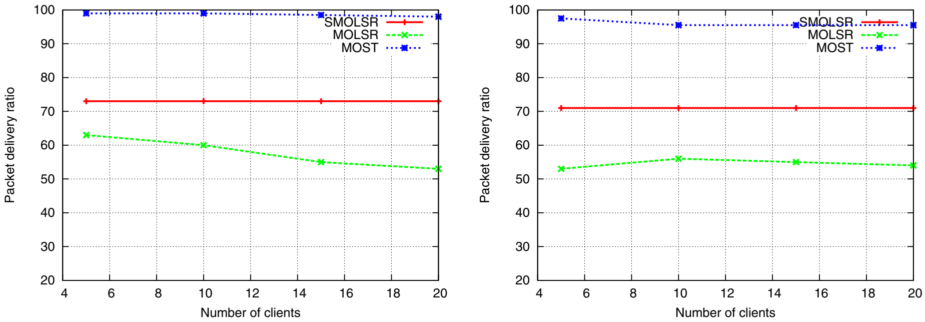


Fig. 6. PDR versus number of groups

=⇒ **Here, the key property for PDR is indirectly the method of transmission (broadcast vs. multicast). With multicast, congestion occurs later than with broadcast, and thus offers better PDR.**

3.6 Protocols Performance with Mobility

In order to evaluate each protocol performance with mobility, we consider the same scenario in which an arbitrary source node sends a CBR traffic of $64kbps$ for 300 seconds. The number of clients ranges from 5 to 20 nodes and the maximum mobility speed varies from $1m/s$ to $10m/s$. The mobility model is the random way-point with a pause time of $10s$: nodes choose a random point in the network area and move to it with a constant speed chosen at random between $1m/s$ and the maximum defined value; after they have reached their destination, they remain static for a period equal to the pause interval and then the same procedure is repeated.



(a) $1m/s$. (b) $10m/s$.

Fig. 7. Comparison of MOST/MOLSR/SMOLR pdr in a mobile network

We first evaluate the impact of mobility on the packet delivery ratio. As we can see, MOST delivery ratio remains very high whatever the speed. However, as the speed increases, traffic load due to duplicate packets increases to reach more than 10%. A duplicate packet means that there was a transient loop. This transient loop is due to link breakage followed by a rebuilding of the multicast tree caused by mobility.

With SMOLSR, packet delivery ratio is not affected by mobility for reasonable speeds (up to $10m/s$). With MOLSR, performance decreases when we increase mobility with small groups, but PDR is no more affected by mobility for bigger groups with reasonable speeds (up to $10m/s$). SMOLSR delivery ratio is better than MOLSR in case of mobility.

Arguably, the complexity of the protocol exchanges is a factor: SMOLSR is the simplest (immediately available with OLSR), MOST is simple since it consists in a declaration of membership (independent from the topology), MOLSR is the most complex since the source tree must be updated and maintained.

$\Longrightarrow$ **The key property for resilience to mobility is the protocol: results suggest that it is related to the complexity of protocol exchanges in reaction to topology changes**

4 Conclusion

The performance evaluation of the three multicast protocols, allows us to draw some conclusions for the design and the choice of a multicast routing protocol. The absolute value for results obtained from MOLSR and SMOLSR is around 50% PDR, for our arguably reasonable scenarios and simulation parameters with a single $64kbps$ source: it illustrates the fact that wireless channel congestion cannot be considered as a non-issue for multicast.

The first observation is that the key performance parameter in several simulations was found to be the choice between unicast and neighborcast. In theory, neighborcast allows for less transmissions by benefiting from the wireless multicast advantage. On the other hand, in practice, in many wireless technologies (including 802.11b,g,...): unicast benefits from higher data rates, from automatic rate adaptation and from reliability features (such as acknowledgments and repetitions), whereas such sophisticated features are not available off-the-shelf for neighborcast. This was reflected in our scenarios and our simulations, with overall higher reliability and lower congestion for the protocol using unicast transmissions, MOST[3]. Hence, these features, actually MAC and physical layer features rather than multicast protocol features, were found to tilt the balance in favor of unicast, and are a crucial design factor.

Note also that even with some cross-layer improvements for neighborcast (such as for instance [17]), in sparse networks or in networks where the density of the multicast group is moderate[4], there are not many opportunities for benefiting from the wireless multicast advantage, and unicast overlay approaches remain excellent contenders.

A second observation is that some multicast protocols (SMOLSR and MOST) are almost built into the underlying routing protocol (OLSR)[5]: we hypothesized that this contributed greatly to their better behavior with respect to external changes in the network (mobility).

Overall, for applications with moderate density of group members, we conclude that approaches such as MOST are overall an excellent choice for multicast, as much for technological and practical reasons as for algorithmic ones.

For high density of group members, the studied protocols and our simulations do not point towards a definite protocol family or conclusion. Still we observe that, by itself, the network-wide broadcast exemplified by SMOLSR is costly when members are concentrated in some areas (with results worse than MOLSR). In such scenarios, a localized broadcast such as geocasting, or the approach of broadcasting in a limited area from source to destinations (for SMOLSR: [18]) should be considered.

In general, for high member density, simulations illustrated the fact that neighborcast is beneficial when the number of transmissions is considered, (which could be for instance even improved in MOLSR, with better tree construction), but the issues remain low data rate (by default) and reliability. These could be overcome by methods accepting a high data rate neighborcast at the price of lower reliability. The trade-off would be worthwhile when loss recovery/compensation has low cost in the broadcast method. One possibility is broadcast with network coding (see DRAGONCAST [19] for an example).

[3] Notice that although neighborcast can be simulated by several unicast transmissions for MOLSR, SMOLSR or any protocol, the gain upon MOST is uncertain since MOST builds a minimum unicast spanning tree anyway (see also [5]).

[4] According to the conducted simulations, the performances of MOST remain good as long as the group size does not exceed 20% of the network size.

[5] MOST requires only the additional knowledge of the group membership (performed through proper proactive advertisements).

References

1. Ali, M.A., El-Sayed, A., Morsi, I.Z.: A Survey of Multicast Routing Protocols for Ad-Hoc Wireless Networks. Minufiya Journal of Electronic Engineering Research (MJEER) 17(2) (July 2007)
2. Hosseini, M., Ahmed, D.T., Shirmohammadi, S., Geoganas, N.D.: Survey of Application-Layer Multicast Protocols. IEEE Comm. Surveys and Tutorials 9(3), 58–74 (2007)
3. Badameh, O., Kadoch, M.: Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy. EURASIP Journal on Wireless Communications and Networking (January 2009)
4. Stojmenovic, I., Wu, J.: Broadcasting and activity scheduling in ad hoc networks. In: Basagni, S., Conti, M., Giordano, S., Stojmenovic, I. (eds.) Mobile Ad Hoc Networking, pp. 205–229. IEEE/Wiley (2004)
5. Jacquet, P., Rodolakis, G.: Multicast Scaling Properties in Massively Dense Ad Hoc Networks. In: ICPADS (2), Fukuoka, Japan, pp. 93–99
6. Rodolakis, G., Meraihi Naimi, A., Laouiti, A.: Multicast Overlay Spanning Tree for Ad Hoc Networks. In: WWIC, Coimbra, Portugal (2007)
7. Hipercom Project-Team, Analysis of three MANET Multicast Protocols: SMOLSR, MOLSR and MOST, Contract CELAR/INRIA Number 06 42 215 (August 2008)
8. Minet, P., Laouiti, A.: Multicasting in Mobile Ad Hoc Networks. In: Handbook of Wireless Ad Hoc and Sensor Networks. Springer, Heidelberg (2008)
9. Hipercom Project-Team, Simple Multicast OLSR (SMOLSR), http://hipercom.inria.fr/SMOLSR-MOLSR/
10. Macker, J. (ed.), the SMF Design Team: Simplified Multicast Forwarding, IETF MANET WG (March 2010), draft-ietf-manet-smf-10 (work in progress)
11. Laouiti, A., Jacquet, P., Minet, P., Viennot, L., Clausen, T., Adjih, C.: Multicast Optimized Link State Routing, INRIA research report RR-4721 (2003)
12. Rodolakis, G., Meraihi Naimi, A., Laouiti, A.: Multicast Overlay Spanning Tree Protocol for Ad Hoc Networks. In: WWIC 2007, Coimbra, Portugal (2007)
13. Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., Viennot, L. (eds.): Optimized Link State Routing Protocol, IETF, RFC 3626 (October 2003)
14. Clausen, T., Dearlove, C., Jacquet, P., the OLSRv2 Design Team: The Optimized Link State Routing Protocol version 2, MANET working group, IETF (April 2010), draft-ietf-manet-olsrv2-11 (work in progress)
15. Hipercom project, Object oriented OLSR (OOLSR), http://hipercom.inria.fr/oolsr/
16. Fenner, W.: Internet Group Management Protocol, version 2, IETF, RFC 2236 (1997)
17. Sun, M.-T., Huang, L., Arora, A., Lai, T.-H.: Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks. In: ICPP 2002 (2002)
18. Cho, S.Y., Adjih, C.: Optimized multicast based on multipoint relaying. In: First International conference on Wireless Internet, WICON 2005, Budapest, Hungary (July 2005)
19. Cho, S.Y., Adjih, C.: Wireless Broadcast with Network Coding: Dynamic Rate Selection. In: Med. Hoc. Net. 2008, Spain (June 2008)

An Energy Efficient Power Control Protocol for Ad Hoc Networks Using Directional Antennas

Carlos Quiroz-Perez and T. Aaron Gulliver

Department of Electrical and Computer Engineering, University of Victoria
PO Box 3055, STN CSC, Victoria BC V8W 3P6, Canada
cquiroz@uvic.ca, agullive@ece.uvic.ca

Abstract. A wireless ad hoc network is a collection of mobile nodes that can communicate with each other. Typically, nodes employ omnidirectional antennas. The use of directional antennas can increase spatial reuse, reduce the number of hops to a destination, reduce interference, and increase the transmission range in a specific direction. This is because omnidirectional antennas radiate equally in all directions, limiting the transmission range.

Because most mobile nodes operate using batteries, protocols which conserve energy are of great interest. In this paper, we introduce the Dynamic Directional Power Control (DDPC) protocol. This protocol dynamically varies the energy used in directional transmission to increase battery life without sacrificing connectivity. DDPC takes into account the remaining battery power in determining the node transmission power. It can achieve a higher network lifetime when compared to a network where nodes use a fixed transmit power level.

1 Introduction

Wireless devices commonly use omnidirectional antennas [8]. These antennas radiate signals in all directions resulting in a circular transmission/reception pattern, so the signal is received by all nodes within range. Since this signal is typically intended for a specific receiver, it is not necessary for all neighboring nodes to receive it. As a consequence, the wireless channel is not efficiently used and the receiver gets only a small part of the energy, with much of the transmitted energy wasted. With directional antennas, a transmitter can concentrate most of its power towards the destination, hence, it is able to reach nodes further from the sender. However, if the receiver is close to the sender, energy from the transmitter may be wasted. Therefore, transmit power control is needed to provide sufficient signal strength to reach the destination without causing too much interference to neighboring nodes. Power control can also increase battery life [3], which in turn increases the lifetime of the network [4].

Improving energy use through software mechanisms or protocols is a good approach since it is harder and more complex to build batteries with better capacity. In [5], it is stated that the optimal transmission power level in wireless ad-hoc networks depends on network conditions such as the number of nodes, the network configuration and the traffic load. Two transmission power mechanisms are proposed, Common Power Control (CPC) and Independent Power Control (IPC). These algorithms adapt the transmission power according to the network conditions in order to improve throughput.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 15–28, 2010.
© Springer-Verlag Berlin Heidelberg 2010

With CPC, all nodes use the same transmission power, while with IPC, nodes use independent transmission power. Contention time (the time taken to successfully send a packet), thresholds are used to determine the transmission power. These algorithms force the nodes to increase or decrease their transmission power when the contention time reaches upper or lower thresholds, respectively. However, they do not take into account the remaining battery power. As a consequence, nodes may run out of power sooner than with a power control strategy that considers this amount.

The authors in [1] propose a power control (P-CON) protocol which takes into account the remaining battery power of the node. The idea is to vary the transmit power to increase network lifetime (when the first node runs out of energy), and to reduce end-to-end delay in wireless ad hoc networks. The source node starts transmitting with a maximum (initial) transmission range, and invokes the P-CON algorithm periodically based on a selected time interval. When P-CON is invoked, the source node reduces its transmit power gradually using a power control tuning parameter, α. If α is smaller than unity, the decreasing transmit power is less sensitive to node battery power changes, while if α is greater than 1, this power is very sensitive to the battery power level. The source node gradually reduces its power until it reaches a fixed minimum power level or *minimum transmission range*. The node continues operating at this minimum until communication with the destination node is completed. The transmit power cannot go below this minimum, which is determined based on network size, number of nodes and node mobility. In [1], P-CON has a nominal minimum transmission range of 175m, with the power control tuning parameter α set to 0.4 for low load traffic and 0.7 for high load traffic. This allows for a gradual reduction in operating range. The problem with P-CON is that it assumes a fixed minimum distance between the source and destination (175m). In addition, P-CON requires knowledge of network parameters such as the traffic load, the number of nodes in the network (and mobility if applicable). Therefore, P-CON is not suitable for dynamic networks. In this paper. we propose an efficient and adaptive protocol, Dynamic Directional Power Control (DDPC), for ad hoc networks. This protocol increases battery life while reducing interference to other nodes.

The rest of the paper is organized as follows. In Section 2, we review the characteristics and advantages of directional antennas and its effects on the transmit power. Section 3 introduces the DDPC protocol while Section 4 presents some performance results. Finally, some conclusions are given in Section 5.

2 Directional Antennas in Ad Hoc Networks

In ad hoc networks, omnidirectional antennas are typically assumed for all nodes. A major drawback with these antennas is that communication between two nodes requires all other nodes in the vicinity to stay silent. In addition, the lower antenna gain with omnidirectional antennas may increase the number of hops a sender needs to reach a distant destination. These issues can often be solved using directional antennas [7]. Directional antennas concentrate the power in a specific direction, instead of spreading it uniformly as with omnidirectional antennas.

A node equipped with N directional antennas can have N beams. The main lobe of each beam spans an angle of $2\pi/N$ radians. For instance, if a wireless node has four directional antennas, the conical radiation pattern of one of its beams will span an angle of $\frac{\pi}{2}$ radians (90°). This angle is referred to as the beamwidth. A node using directional antennas can select only one of its beams with a main lobe gain of G_d. The narrower the beamwidth, the higher the gain, so these antennas offer greater transmission range, but with a reduced coverage angle. Antenna gain is given in units of dBi, dB gain with respect to an isotropic source [2]. Based on [11], the antenna gain is given by

$$G = \frac{2}{1 - \cos\left(\frac{\pi}{180} \times \frac{\text{beamwidth}}{2}\right)} \tag{1}$$

For example, suppose the gain with an omnidirectional antenna is $G_{0_{360^\circ}} = 0$ dB. Using a directional antenna with a beamwidth of 90° gives $G_{d_{90^\circ}} = 8.34$ dBi, and a beamwidth of 60° gives $G_{d_{60^\circ}} = 11.74$ dBi.

The minimum required transmit power, P_T, for correct reception at a distance d can be expressed as

$$P_t = G_T^{-1} G_R^{-1} d^2 \tag{2}$$

where G_T and G_R are the transmit and receive antennas gains, respectively. Note that for simplicity in comparing power control techniques, we do not consider performance as a function of the signal to interference ratio (SINR). The effective communication distance between two nodes is a function of the product of their antenna gains, consequently directional antennas provide range extension. Directional antennas can also provide higher spatial reuse, greater connectivity, and reduced interference to neighboring nodes. Figure 1 shows four nodes transmitting simultaneously in the same neighborhood. This is possible because directional antennas are used.

In this paper, directional transmission and omnidirectional reception is denoted as *Dtx-Orx communication*, and both directional transmission and reception is denoted as *Dtx-Drx communication*. Dtx-Drx communication has a longer link-length than Dtx-Orx communication. For example, suppose a directional antenna with a beamwidth of 90° transmitting to a receiver with an omnidirectional antenna can communicate up to

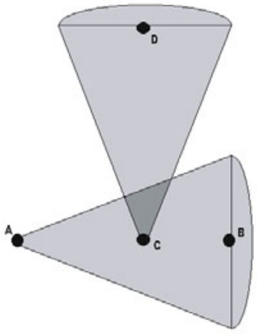


Fig. 1. Four nodes transmitting simultaneously in the same neighborhood using directional antennas

a distance of 1.2 kms. If the beamwidth is reduced to 60°, this distance increases to approximately 2 kms. This is referred to as the *transmission range* [1].

3 The Dynamic Directional Power Control (DDPC) Protocol

Figure 2 shows Dtx-Orx communication between nodes A and B. If the transmission range is 1.2 kms, and node B is located 1.0 km from A, there is a wasted transmission range of 200 m. We refer to this as the *inefficient transmission range*. The goal of transmit power control is to reduce or minimize this range.

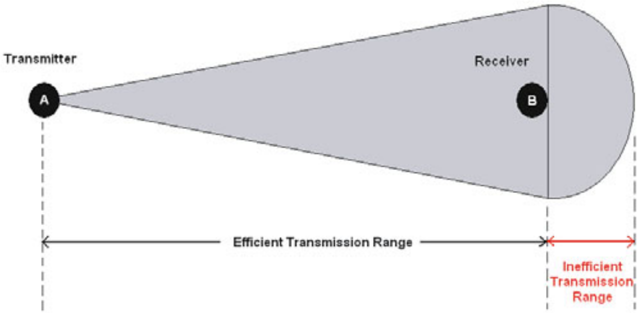


Fig. 2. Efficient and inefficient transmission ranges of a node when using a directional antenna

In this paper we propose a power control solution called *Dynamic Directional Power Control* (DDPC). DDPC shares some characteristics of the P-CON protocol [1]. However, DDPC provides improvements to make the power control more dynamic and efficient than P-CON. Figure 3 shows a block diagram of the DDPC algorithm in the IEEE 802.11 structure and its relationship with the MAC and PHY layers. In order to vary the transmission range, it is necessary to change the transmission power at the PHY layer, so DDPC is implemented at this layer.

DDPC borrows and extends the functionality of the *power control tuning parameter*, α, from P-CON. In P-CON, α determines the sensitivity of the power control strategy to changes in the remaining battery energy at a node [1]. In DDPC, α determines not only the power control sensitivity, but also whether the transmit power should be increased or decreased.

In order to maintain an efficient transmission range, DDPC increases/reduces the transmit power accordingly. Therefore, DDPC must know when the transmitting node loses connectivity with the receiver. When a transmitter does not receive an ACK control packet from the destination confirming that a packet has been received, the MAC layer (at the transmitter) notes the absence of an ACK and retransmits the data packet. DDPC uses the absence/presence of an ACK at the MAC layer to increase or reduce the transmit power.

During transmission, a source node receives an ACK for every data packet that is sent, and the MAC layer notifies ($TxPower = 1$) DDPC that it is receiving ACK

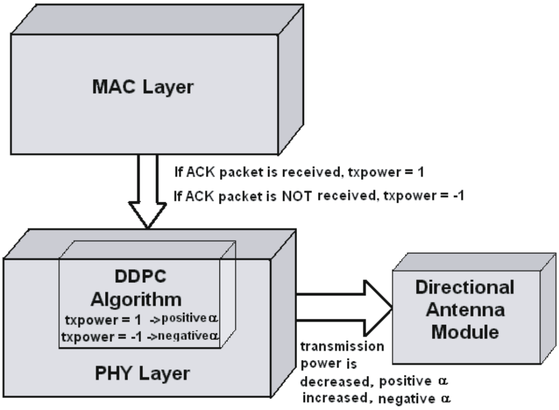


Fig. 3. The Dynamic Directional Power Control (DDPC) algorithm layer implementation with the IEEE 802.11 structure

packets. As a consequence, DDPC uses a *positive* α, which reduces transmission power. On the other hand, when ACK packets stop being received due to lost connectivity, the MAC layer informs ($TxPower = -1$) DDPC that the connection has been lost. As a result, DDPC uses a *negative* α, which increases transmission power to recover connectivity with the destination. The transmit power level ata node is then given by

$$CurrTx = MaxTx * \left(\frac{B_{rem}}{B_{init}}\right)^{\alpha} \tag{3}$$

where B_{rem} is the remaining (available) battery energy, B_{init} is the initial battery energy, $MaxTx$ is the maximum transmission power (set to $MaxTx$ = 0.28 W or $\approx$ 24 dbm [12]), $CurrTx$ is the current transmission power (at the beginning of a transmission ($t = 0$), $CurrTx$ = $MaxTx$), and α is the power control tuning parameter. If α is *positive*, $CurrTx$ decreases. If α is *negative*, $CurrTx$ increases. Proper selection of α is crucial for efficient transmission. Using the MAC layer to monitor connectivity gives DDPC more accuracy than P-CON in varying the transmit power to maintain connectivity.

Three mechanisms have been developed (Approach $\#1$, Approach $\#2$, and Approach $\#3$), to illustrate the performance and flexibility of DDPC. Figure 4 shows the transmit power variation with P-CON and the three DDPC approaches. With P-CON, as shown in Figure 4a), a node starts transmitting with a power of $MaxTx$. Then the transmit power is reduced until it reaches the *minimum transmission range* with power $MinTx$. This might not be the lowest transmit power before losing connectivity. Once reaching $MinTx$, the node continues to operate at $MinTx$. DDPC Approach $\#1$, is shown in Figure 4b). As with P-CON, the transmission power is decreased from a level of $MaxTx$. When the transmitter detects an absence of ACK (NACK), it starts increasing the transmit power until it reaches $MaxTx$. This process is repeated until the end of the transmission. DDPC Approach $\#2$ is illustrated in Figure 4c). This is similar to P-CON, but instead of using $MinTx$, it uses an absence of ACK (NACK) to determine the

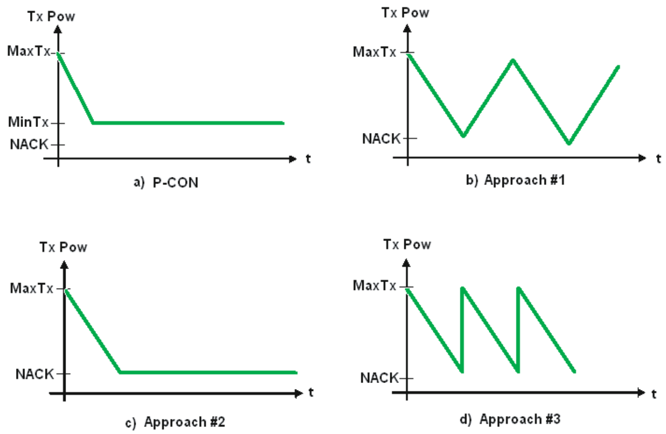


Fig. 4. The transmit power variation under different power control mechanisms: a) P-CON, b) DDPC Approach #1, c) DDPC Approach #2, d) DDPC Approach #3

minimum power level. Once a NACK is detected, DDPC increases the transmit power to a level just high enough to recover connectivity. Then DDPC continues transmitting at this level, which corresponds to the *efficient transmission range*. DDPC Approach #3 is depicted in Figure 4d). This is similar to Approach #1 with the difference that instead of gradually increasing power to $MaxTx$ once a NACK is received, Approach #3 immediately increases the transmit power to $MaxTx$. This process is repeated until the end of the transmission. Since each approach in DDPC represents a different power control strategy, each approach uses different α values in their corresponding algorithms.

3.1 Simulation Environment and Performance Results

To evaluate our protocol, we employed *Network Simulator*, NS-2 [10]. NS-2 is a discrete event simulator widely used in the research community [9]. In order to implement directional antennas in NS-2, we use *The Enhanced Network Simulator* (TENS) [11]. TENS is an extension of NS-2 which provides additional features such as directional antennas. We modified NS-2 to implement the DDPC protocol in the PHY layer and added a link between DDPC and the MAC layer.

We focus only on energy consumption in transmission, and do not consider energy consumed in the idle state and during reception. Since most power is used in transmission, and we are considering different transmit power algorithms, this allows for a clear and fair comparison. Table 1 summarizes the simulation parameters employed. Dtx-Orx communication links use directional antennas with 4 and 6 beams for transmission, while Dtx-Drx communication links have directional antennas for both transmission and reception. For P-CON, we use $\alpha = 0.7$ and a $MinTx$ corresponding to 175 m, as in [1].

Table 1. Simulation Parameters

	Network area	1500 × 1500
	Simulation Time	10000 seconds
Directional Antennas	Beam Width Angle	Number of Beams
	90°	4 beams
	60°	6 beams
DDPC Parameters	$MaxTx$	0.2818 W ≈ 24.5 dBm
	B_{init}	10 J (static), 100 J (mobility)
	α	20, 100, 10
PHY Layer	Signal Propagation Model	Two-ray ground
MAC Layer	IEEE 802.11	
	Link Bandwidth	2 Mbps
	Interface Queue Length	FIFO, size 50
Routing Protocol	AODV	
Traffic Model	Constant Bit Rate (CBR), UDP	
	Data Packet Size	1000 Bytes
	Data Rate	740.0 kbps
Static Model	2 nodes	Distance = 250 m / 600 m
	4 nodes	Distance = 250 m / 600 m
Mobility Model	Type of Communication	Dtx-Drx
	2 nodes	Distance = 50 m to 1000 m
	Beam Width Angle	60°
	Node Speeds	[1, 2, 5, 10, 20, 50] m/s

3.2 Static Network: Single Transmission

We denote 802.11 Directional and P-CON Directional with N antenna beams as 802.11 N-Directional and P-CON N-Directional, respectively. Similarly, we refer to DDPC with N antenna beams as N-DDPC.

Figure 5 shows the energy consumption using a Dtx-Orx communication link with 4 directional antenna beams. The distance between the transmitter and receiver is 600 m. Approach #2 has the longest transmission time, while 802.11 and P-CON have the shortest times. Figure 6 shows the energy consumption using a Dtx-Orx communication link with 6 directional antenna beams. The relative performance is the same as with 4 beams, with Approach #2 still providing the longest transmission time, but it is now ≈3500 seconds as opposed to ≈1600 seconds. P-CON finishes its available battery energy after only ≈200 seconds in both cases. P-CON has poor performance because it was developed with static parameters, and so is not dynamic.

3.3 Static Network: Simultaneous Transmissions

We now evaluate the performance when there are simultaneous transmissions in the same neighborhood. From Figure 1, we refer to the transmission from A to B as *flow 1* and the transmission from C to D as *flow 2*. The results in this section were obtained using directional antennas with 4 beams at both the transmitters and receivers. Figure 7 shows the energy consumed in flow 2 with a separation distance of 250 m. Approach

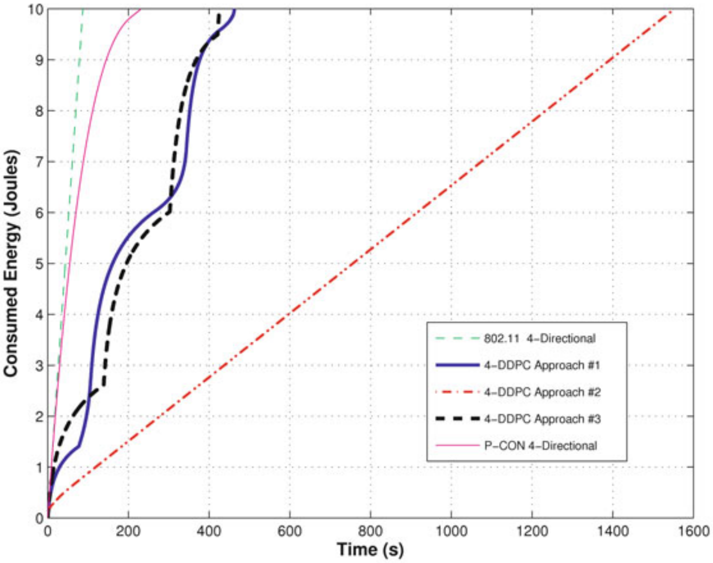


Fig. 5. Energy consumption with a Dtx-Orx communication link and a 90° (4 beams), directional antenna at a distance of 600 m

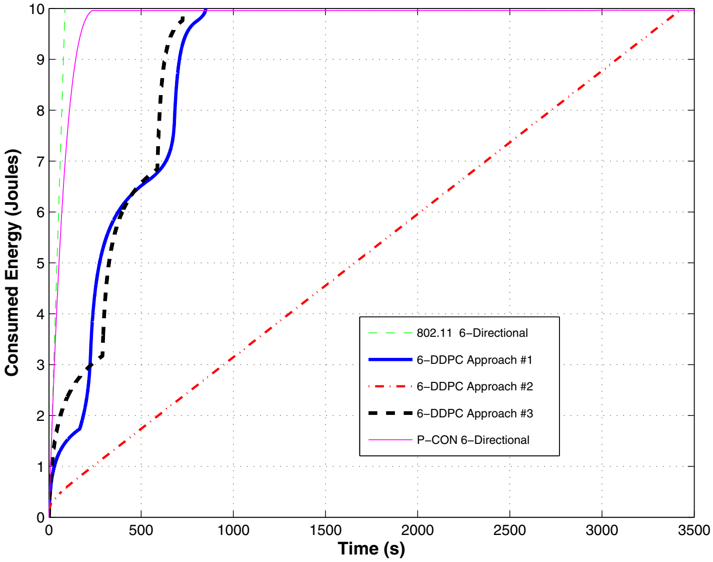


Fig. 6. Energy consumption with a Dtx-Orx communication link and a 60° (6 beams), directional antenna at a distance of 600 m

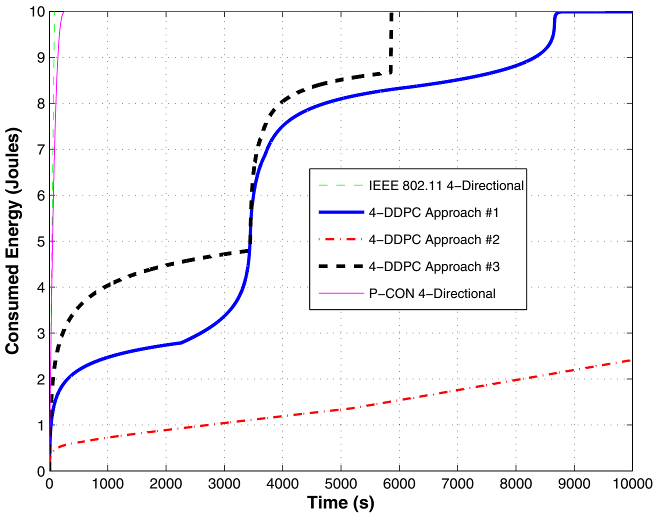


Fig. 7. Energy consumption with a Dtx-Drx communication link (flow 2) and a 90° (4 beams), directional antenna at a distance of 250 m

$\#2$ still provides the best performance, followed by the other DDPC approaches. Since two simultaneous transmissions occur in the same area, network performance is affected by packet loss caused by collisions. Table 2 shows that all the DDPC approaches experience some packet loss. In the event of packet loss, the transmitter automatically resends packets that have not been acknowledged. Retransmission of packets causes the throughput to decrease, and consequently greater energy consumption per received packet.

From the table, we observe that P-CON is the most affected by interference. It provides a packet delivery ratio of only 2.97% (percentage of successful packets from the total number of packets transmitted). In both flows, Approach $\#3$ provides the highest packet delivery ratio (92.38%) and the lowest packet loss (3.81%). The reason is that the transmit power in Approach $\#3$ jumps to $MaxTx$ whenever there is a loss of connection. This minimizes the effects of interference. Although Approach $\#2$ lasts the longest, Approaches $\#1$ and $\#3$ provide better packet delivery ratios. The is because Approach $\#2$ is more sensitive to interference since it operates with a low power level. In flow $\#2$, the packet delivery ratio of Approach $\#2$ is slightly higher than in flow $\#1$ ($78.16\% > 67.88\%$). This is because flow $\#2$ transmissions started slightly before flow $\#1$. Similar results were obtained with the other approaches.

In Table 2, we also observe that Approach $\#2$ has the highest average packet delay (731.52 ms), followed by Approach $\#1$ (442.26 ms) and Approach $\#3$ (238.10 ms). This can be attributed to interference since low power transmissions are more sensitive to this. Conversely, Approach $\#3$ transmits more often at a high power level (the absence of an ACK packet causes a jumps to the $MaxTx$ power level), so it provides the lowest average delay of the DDPC approaches. Note that average packet delay corresponds to received packets, so the low value for P-CON is due to a high packet loss rate.

Table 2. Static network performance with two flows using a Dtx-Drx (90° beamwidth antennas) communication link with a separation distance of 250 m

	Flow #1			
	Approach#1	Approach#2	Approach#3	P-CON
Packet Delivery Ratio	77.92%	67.88%	92.38%	2.97%
Average Delay	442.26 ms	731.52 ms	238.10 ms	4.56 ms
Packet Loss	11.04%	16.06%	3.81%	97.12%
	Flow #2			
	Approach#1	Approach#2	Approach#3	P-CON
Packet Delivery Ratio	77.95%	67.92%	92.47%	2.98%
Average Delay	440.22 ms	634.53 ms	237.68 ms	4.58 ms
Packet Loss	11.02%	10.92%	3.76%	97.11%

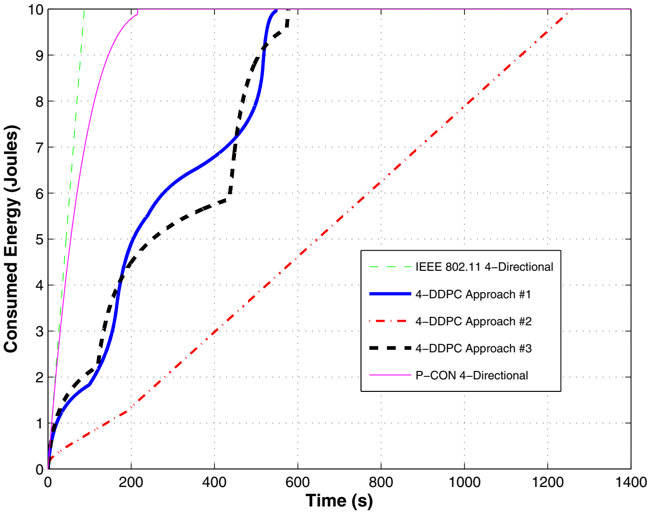


Fig. 8. Energy consumption with a Dtx-Drx communication link (flow 2) and a 90° (4 beams), directional antenna at a distance of 600 m

Figure 8 shows that a longer distance between the transmitter and destination (600 m) increases the energy consumption. This also increases the wait for ACK packets. As before, interference degrades network performance because of the retransmission of lost packets. Table 3 gives the packet delivery ratio of the DDPC approaches and P-CON. Although the battery life with all approaches is less than with a distance of 250 m, Approach #2 still transmits the longest. However, Approach #2 has the lowest packet delivery ratio. This is because the low transmit power used by Approach #2 is more susceptible to packet loss due to interference. Approach #3 performs better than Approach #1 because Approach #3 uses the highest power level more frequently. Therefore, Approach #3 is less affected by interference. As a result, Approach #3 shows the lowest packet loss rate and the highest packet delivery ratio. Even though

Table 3. Static network performance with two flows using a Dtx-Drx (90° beamwidth antennas) communication link with a separation distance of 600 m

	Flow #1			
DDPC Scheme	Approach#1	Approach#2	Approach#3	P-CON
Packet Delivery Ratio	68.36%	56.23%	70.72%	2.12%
Average Delay	715.89 ms	890.45 ms	590.46 ms	4.59 ms
Packet Loss	31.64%	43.77%	14.60%	97.88%
	Flow #2			
DDPC Scheme	Approach#1	Approach#2	Approach#3	P-CON
Packet Delivery Ratio	68.10%	56.11%	70.95%	2.14%
Average Delay	716.03 ms	890.38 ms	587.87 ms	4.48 ms
Packet Loss	31.90%	43.89%	14.43%	97.80%

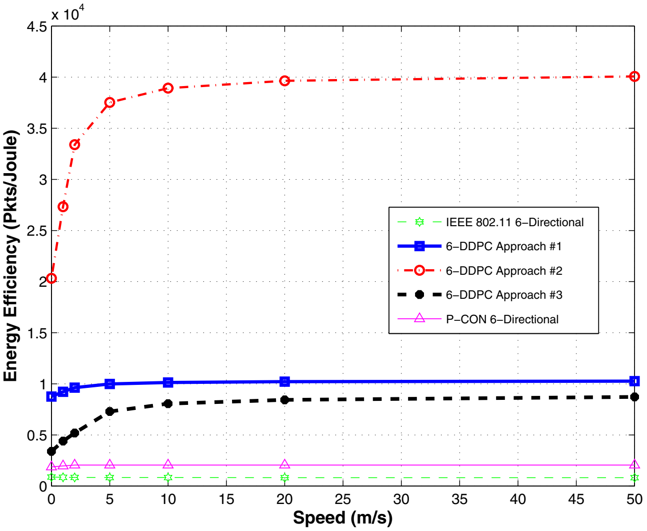


Fig. 9. Energy efficiency of two nodes in motion with speeds 1, 2, 5, 10, 20 and 50 m/s

P-CON uses a high power level most of the time, its packet delivery ratio is the lowest. The reason is that P-CON consumes its energy the fastest. This leaves a very short time (200 sec.) to deliver packets to the destination. In Table 3, we observe a similar pattern in the average delay as in the previous table. Approach #2 shows the highest average delay (890.45 ms) because it uses the lowest power level during transmission. Conversely, P-CON has the lowest average delay (459 ms) due to its high power transmissions. This confirms that a low transmit power results in a higher average delay.

3.4 DDPC Performance with Mobility

In this section, we evaluate the performance of the DDPC protocol with node mobility. In this case, the destination node first moves towards the transmitter. When the

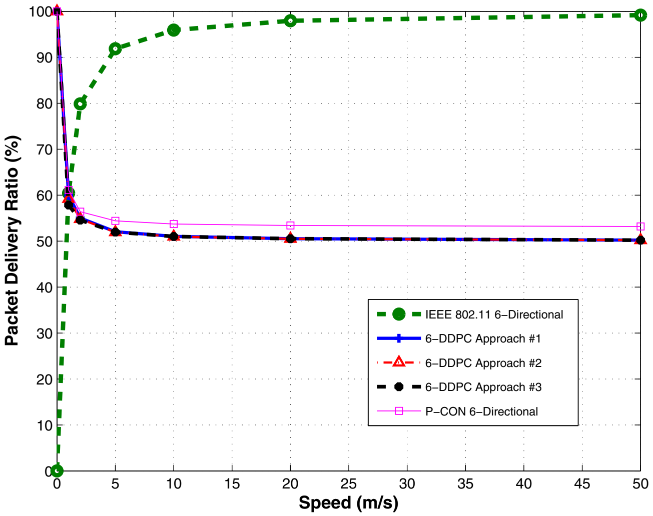


Fig. 10. Packet delivery ratio of two nodes in motion with speeds 1, 2, 5, 10, 20 and 50 m/s

destination node is 50 m away, it stops moving. Then, when the simulation time is $t = 1000$ sec., the destination node starts moving away from the transmitter. The destination node stops moving when the separation distance is 1 km. Simulation stops either when the time ends ($t = 2000$ sec.), or when the node has consumed all its energy.

Figure 9 shows the energy efficiency of the DDPC approaches at different speeds. At low speeds (< 5 m/s), the destination node takes longer to reach the source. Thus, there is less energy efficiency due to long distances. At high speeds (> 10 m/s), the energy efficiency improves. This is because the destination approaches the source faster, and less energy is consumed with shorter distances. IEEE 802.11 and P-CON consume their battery energy faster than the DDPC approaches. As a result, they provide the lowest energy efficiency at all speeds. DDPC Approach $\#1$ consumes less energy than Approach $\#3$. Approach $\#3$ consumes more energy than Approach $\#1$. Approach $\#2$ provides higher energy efficiency since the minimal power used during transmission is enough to maintain connectivity.

Figure 10 shows the packet delivery ratio with each approach when the destination is in motion at several speeds. It is interesting to see that the total data delivered per joule for IEEE 802.11 is low, however, its packet delivery ratio is the highest. Even though IEEE 802.11 consumes its energy rapidly, its constant high power allows it to deliver more packets to the destination. Thus it is suitable for applications where the transmission of critical information is more important than efficiency. Since P-CON consumes more energy compared to the DDPC approaches, it provides lower data delivery per joule, but still provides a higher packet delivery ratio. This is because the higher energy consumption allows P-CON to better maintain connectivity. The DDPC approaches achieve energy efficiency (conserve more energy) by sacrificing the packet delivery ratio. From the results, it is clear that as node speed increases, this ratio decreases because of connectivity loss due to motion. Variations in transmit power due to DDPC can potentially compromise network connectivity when there is high mobility.

4 Conclusions

In this paper, we proposed the *Dynamic Directional Power Control* (DDPC) protocol, which dynamically varies the transmission power. When connectivity with the destination is lost, DDPC reacts to restore connectivity while saving battery energy. An important characteristic of DDPC is that it takes into account the remaining battery energy when adjusting the transmit power. DDPC was implemented with directional antennas to improve range, reduce interference, and increase energy savings.

In a static network without interference and *Dtx-Drx* communication links, DDPC Approach #2 consumed the least battery energy. In a static network with simultaneous transmissions (interference), DDPC can allow two pairs of nodes to communicate in the same vicinity. With interference and short separation distances, Approach #3 provides the best packet delivery ratio and lasts the longest. However, Approach #3 also consumes battery energy faster than the other two approaches. Approach #3 also provides the lowest packet loss rate. When the separation distance between the transmitter and receiver is longer, Approach #2 outperforms the other two schemes. In this case, the packet delivery ratio is lower since DDPC consumes more energy with longer distances. We conclude that the DDPC protocol is well suited to static conditions with and without interference, in which case DDPC outperforms P-CON and IEEE 802.11 DDPC also offers better energy savings when nodes are in motion over short distances.

The focus of this paper was on energy savings and simple configurations were considered for which AODV is adequate. For multihop communications, the proposed protocol can be combined with a directional routing protocol such as that proposed in [13].

References

1. Meghanathan, N., Farago, A.: Power Sensitive Power Control in Ad Hoc Networks. In: ACM Southeast Conference, pp. 18–20 (2005)
2. Choudhury, R., Vaidya, N.H.: Impact of Directional Antennas on Ad Hoc Routing. In: Conti, M., Giordano, S., Gregori, E., Olariu, S. (eds.) PWC 2003. LNCS, vol. 2775, pp. 590–600. Springer, Heidelberg (2003)
3. Narayanaswamy, S., Kawadia, V., Sreenivas, R.S., Kumar, P.R.: Power Control in Ad-hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In: European Wireless Conference, pp. 156–162 (2002)
4. Kaddar, L., Mehaoua, A.: ESTREL: Transmission and Reception Energy Saving Model for Wireless Ad Hoc Networks. In: IEEE Conference on Local Computer Networks, pp. 784–785 (2007)
5. Park, S.J., Sivakumar, R.: Load-Sensitive Transmission Power Control in Wireless Ad-hoc Networks. In: IEEE Global Telecommunications Conference, pp. 42–46 (2002)
6. Choudhury, R.R., Yang, X., Ramanathan, R., Vaidya, N.H.: Using Directional Antennas for Medium Access Control in Ad Hoc Networks. In: ACM/IEEE International Conference on Mobile Computing and Networking, pp. 59–70 (2002)

7. Ko, Y.-B., Shankarkumar, V., Vaidya, N.H.: Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks. In: IEEE Conference on Computer Communications, pp. 13–21 (2000)
8. Gast, M.S.: 802.11 Wireless Networks: The Definitive Guide. O'Reilly Media, Sebastopol (2002)
9. Dricot, J.-M., Doncker, P.D.: High-Accuracy Physical Layer Model for Wireless Network Simulations in NS-2. In: International Workshop on Wireless Ad-Hoc Networks, pp. 249–252 (2004)
10. McCanne, S., Floyd, S.: NS Network Simulator, http://www.isi.edu/nsnam/ns/
11. Raman, B., Chebrolu, K.: The Enhanced Network Simulator (TENS), http://www.cse.iitk.ac.in/users/braman/tens/
12. Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., Yu, H.: Advances in Network Simulation. IEEE Computer 33, 59–67 (2000)
13. Nasipuri, A., Mandava, J., Manchala, H., Hiromoto, R.E.: On-demand Routing Using Directional Antennas in Mobile Ad Hoc Networks. In: International Conference on Computer Communications and Networks, pp. 535–541 (2000)

On Minimizing the Sum of Sensor Movements for Barrier Coverage of a Line Segment

Jurek Czyzowicz[1], Evangelos Kranakis[2], Danny Krizanc[3], Ioannis Lambadaris[4], Lata Narayanan[5], Jaroslav Opatrny[5], Ladislav Stacho[6], Jorge Urrutia[7], and Mohammadreza Yazdani[4]

[1] Département d'informatique, Université du Québec en Outaouais, Gatineau, QC, J8X 3X7, Canada, Supported in part by NSERC grant
[2] School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada, Supported in part by NSERC and MITACS grants
[3] Department of Mathematics and Computer Science, Wesleyan University, Middletown CT 06459, USA
[4] Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, K1S 5B6, Canada, Supported in part by NSERC and MITACS grants
[5] Department of Computer Science, Concordia University, Montréal, QC, H3G 1M8, Canada, Supported in part by NSERC grant
[6] Department of Mathematics, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada, V5A 1S6, Supported in part by NSERC grant
[7] Instituto de Matemáticas, Universidad Nacional Autónoma de México, Área de la investigación cientifica, Circuito Exterior, Ciudad Universitaria, Coyoacán 04510, México, D.F. México, Supported in part by CONACYT grant

Abstract. A set of sensors establishes barrier coverage of a given line segment if every point of the segment is within the sensing range of a sensor. Given a line segment I, n mobile sensors in arbitrary initial positions on the line (not necessarily inside I) and the sensing ranges of the sensors, we are interested in finding final positions of sensors which establish a barrier coverage of I so that the sum of the distances traveled by all sensors from initial to final positions is minimized. It is shown that the problem is NP complete even to approximate up to constant factor when the sensors may have different sensing ranges. When the sensors have an identical sensing range we give several efficient algorithms to calculate the final destinations so that the sensors either establish a barrier coverage or maximize the coverage of the segment if complete coverage is not feasible while at the same time the sum of the distances traveled by all sensors is minimized. Some open problems are also mentioned.

Keywords and phrases: Mobile Sensor, Barrier Coverage, Line segment, Efficient Algorithm, NP-complete, Movement Optimization.

1 Introduction

An important application of wireless sensor networks involves the surveillance of a given region. This surveillance can be done in two different ways: either sensors

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 29–42, 2010.
© Springer-Verlag Berlin Heidelberg 2010

are be placed throughout the region to monitor the activity in the entire region, or sensors are placed along the perimeter of a region where they establish a barrier that can detect intruders attempting to penetrate the region. Surveillance of a region by such a barrier is more efficient in comparison with complete coverage of the region, since it can be established with fewer sensors at a lower cost. When the perimeter of the region to be monitored is difficult to access or contaminated, it might not be feasible to place sensors right away on the perimeter of a region so that a barrier coverage of the perimeter is achieved. However, in such situation mobile sensors can be used, they can be dropped at some arbitrary initial positions and the mobile sensors are then instructed to move to some specific positions on the border to establish a barrier at the perimeter of the region. Since in sensor networks energy available to a sensor is very limited, one of the main considerations in deployment of sensor networks is the efficient use of energy. Thus, when using mobile sensors to establish a barrier at the perimeter of a region, one would be interested to determine for each sensor a specific position at the border so that sensors in these position establish a barrier coverage and the moves to these position can be done with the minimal possible energy consumption.

In a general setting of the barrier coverage problem, there is a predefined geometric planar region with a well defined boundary and a given set of mobile sensors. Each sensor, say S, has a pre-determined *sensing range* $r(S)$ (determined by the manufacturer). Thus when S is located at location u any other point p in the plane is within the sensing range of the sensor if and only if its Euclidean distance from u is at most $r(S)$. The sensors are initially placed in the plane in arbitrary locations either interior or exterior to the region. They are able to move in any direction in the plane and the energy consumption for movement is similar among the sensors and is proportional to the distance traveled. Starting from these initial positions we are interested in calculating final destination of each sensor so that the sensors in final destinations establish a barrier coverage of the region, i.e., no part of the boundary is outside the sensing range of all the sensors, and the sum of the distances traveled by all sensors is minimized. The above optimization problem, referred to as *MinSum*, represents the minimization of the total energy consumed by all the sensors needed to establish a barrier coverage of the boundary of the given region.

In this paper we restrict our study to the one dimensional barrier coverage problem. We are given a line and the barrier is represented by a finite segment on the line. The sensors are initially located on the line containing the barrier, possibly outside the given barrier. We consider the problem of minimizing the sum of movements of sensors within the line in order to achieve a barrier coverage. We assume that an intruder is a mobile agent that may cross the given barrier from any direction in the plane. As before an intruder can be detected only if it is within the sensing range (range for short) of at least one sensor of the wireless sensor network and thus the sensor network establishes barrier coverage if every point of the barrier is within the sensing range of at least one sensor.

Although the problem is restricted to a simplified one-dimensional barrier version, it will become apparent in the sequel that it still contains both challenging algorithmic questions and interesting solutions that illustrate the complexity of MinSum barrier coverage in this setting. Clearly we have to have a good understanding of the one-dimensional version before considering the two-dimensional problem.

1.1 Preliminaries and Notation

We now give several preliminary concepts and define more precisely several variants of the MinSum barrier coverage problem.

An instance of a barrier coverage problem consists of a closed line interval $I = [0, L]$, the *barrier* to be covered, on the real line with pre-defined endpoints 0 and $L > 0$. We also have n sensors $S_1, S_2, \ldots, S_n$ in initial positions $x_1 \leq x_2 \leq \cdots \leq x_n$ on the line (possibly outside the interval $[0, L]$), and the range of the i-th sensor is a given positive real number $r_i = r(S_i)$, $1 \leq i \leq n$.

Thus the set of points (not necessarily of $[0, L]$) which is within the range of sensor S_i in position x_i is the closed interval $I(S_i, x_i) = [x_i - r_i, x_i + r_i]$ of length $2r_i$. We call it the *covering interval* of S_i. The *total sensor range* of a given instance, denoted R, is the sum of lengths of covering intervals of all sensors, i.e., $R = \sum_{i=1}^{n} 2r_i$.

First of all observe that the barrier coverage problem is *feasible* if and only if the total sensor range R is at least as large as the interval $[0, L]$, i.e., $R \geq L$. In the sequel, we also consider the *non-feasible* case $R < L$. In this case we will be interested in optimizing the sensor movements so that the sensors in the final positions cover either a sub-interval or sub-intervals of I of total length R.

Given an instance of a barrier coverage problem, we call a *gap* a sub-interval of I none of whose points is within range of any sensor and which cannot be enlarged any further. Since the ranges of sensors are assumed to be closed intervals, a gap is an open sub-interval of $[0, L]$, except when one of the endpoints of the gap is either 0 or L. Thus if interval $[a, b]$ is a gap, we assume that a, or b is not a part of the gap unless $a = 0$ or $b = L$. We call an *overlap* either a sub-interval of I which is covered by more than one sensor and which cannot be enlarged any further, or a sub-interval of the line outside I which is covered by a sensor and which cannot be enlarged any further.

Optimization problems. Given an instance of the barrier coverage problem we investigate how to determine the final destinations of the sensors so that the barrier is covered by sensors and the sum of the distances traveled by the respective sensors to their final destinations in minimized. As mentioned before, the sum of distances traveled by sensors corresponds to the total energy needed by sensors to reach the final configuration. More formally, if the i-th sensor S_i moves by a distance m_i (a movement to the left, right will be indicated by $m_i < 0$, $m_i > 0$, respectively) from its original position x_i, the new position will be $x_i + m_i$ and the new covering interval will be $I(S_i, x_i + m_i)$. If the problem is feasible we are interested in studying the following optimization problem.

MinSum optimization problem $R \geq L$:

$$\text{minimize } \{\sum_{1\leq i\leq n} |m_i|\} \text{ subject to } [0, L] \subseteq \bigcup_{i=1}^{n} I(x_i + m_i). \tag{1}$$

When $R < L$ and thus the problem is not feasible, we are interested in a *best effort* solution, i.e., an arrangement of sensors that attains the largest possible coverage of $[0, L]$, while at the same time achieving the MinSum requirements of the movements of the sensors. We call *contiguous* an arrangement of sensors that attains the largest possible coverage of size R as a contiguous sub-interval of $[0, L]$, and *non-contiguous* an arrangement of sensors that attains the largest possible coverage of size R as a collection of possibly disjoint sub-intervals.
Non-contiguous MinSum optimization problem for $R < L$:

$$\begin{aligned}\text{minimize } \{\sum_{i=1}^{n} |m_i|\} \text{ subject to } & \bigcup_{i=1}^{n} I(S_i, x_i + m_i) \subseteq [0, L] \text{ and} \\ & |\bigcup_{i=1}^{n} I(S_i, x_i + m_i)| = R.\end{aligned} \tag{2}$$

Contiguous MinSum optimization problem for $R < L$:

$$\begin{aligned}\text{minimize } \{\sum_{i=1}^{n} |m_i|\} \text{ subject to } & \bigcup_{i=1}^{n} I(S_i, x_i + m_i) \subseteq [0, L] \text{ and} \\ & \bigcup_{i=1}^{n} I(S_i, x_i + m_i) \text{ is an interval of size } R.\end{aligned} \tag{3}$$

1.2 Related Work

Several recent papers in the area of sensor networks considered the problem of deployment of mobile sensors for coverage of a region, see for example ([11]), ([12]), and ([13]). Unlike the problem considered in this paper, they aim to provide coverage of an entire two-dimensional region, and their algorithms do not consider the optimization problems stated above.

The problem studied in our paper is motivated by securing an area by ensuring its border surveillance and intruder detection with a wireless sensor system. ([10]) proposes efficient algorithms to determine, after sensor deployment, whether a region is barrier covered. It also establishes optimal deployment patterns to achieve barrier coverage when deploying sensors deterministically. In addition, they consider barrier coverage with high probability when sensors are deployed randomly. In ([4]) the problem of local barrier coverage is introduced and it is shown that it is possible for individual sensors to locally determine the existence of local barrier coverage, even when the region of deployment is arbitrarily curved. Techniques for deriving density estimates for achieving barrier coverage and connectivity in thin strips are introduced in ([1]), where sensors are deployed as a barrier to detect moving objects and events. In all these instances the problem studied concerns *static* optimal sensor deployment patterns and there is no concept of mobility of the sensors.

Related to our study is the work in ([7]) but it does not consider the coverage problem. Also related is a supply and demand problem, known in the literature as *Earth Movers Problem* (or EMP for short), see ([5]), ([3]), ([9]). Despite some similarities EMP differs from our problem in several respects and the results for EMP cannot be used to solve the barrier coverage problem studied here.

There are two papers which are closely related to our study. The first is ([2]), where a similar but simpler problem was introduced and studied. Their optimization problem differs from our model in that they do not specify the sensor ranges to be employed; unlike in our paper they seek algorithms to move the sensors to "equidistant" locations on the barrier so as to optimize the efficiency of the barrier coverage regardless of the initial coverage of the sensors. For example, according to their model the n sensors will move from their initial positions to the specific locations $0, \frac{L}{n-1}, \ldots, \frac{iL}{n-1}, \ldots, \frac{(n-2)L}{n-1}, L$, respectively. In our work the algorithms are sensitive to the predefined sensor ranges (which are given as input to the problem) thus accomplishing the same barrier coverage task with less movement than may be done in ([2]). Similar observations apply to the other cases of the two dimensional versions of the problem considered in ([2]).

The second and most directly related research is done in ([6]) where the same geometric setting is being considered: n sensors on a line that want to establish a barrier coverage of a given line segment by moving the sensors to new positions, but a different optimization measure is being analyzed. Namely, the final positions of sensors that establish barrier coverage minimize "the maximum distance traversed" by any sensor, as opposed to the "sum of the distances covered" considered in the present paper. The motivation for the problem studied in ([6]) is to minimize the time required to attain coverage while in the problem studied here we minimize the total energy consumed. Despite the apparent similarity of the two problems the results and algorithms are quite different.

1.3 Results of the Paper

In this paper we study several interesting variants of the barrier coverage problem obtained by changing assumptions on the sensors and final destinations, e.g., when (a) the sensors may have different ranges, (b) the sensors have identical ranges, (c) the resulting coverage is contiguous or non-contiguous and study the complexity of the proposed algorithms. Several instances of the problem are shown to have efficient algorithmic solutions while others are shown to be NP-complete even to approximate up to constant factor (see Remark 1 after the proof of Theorem 1). Our results are summarized in Table 1 below.

Section 2 presents NP completeness results for MinSum problems for sensors with non-identical ranges. Section 3 deals with sensors of identical ranges. Subsection 3.1 includes the ordering lemma which is basis for the remaining results of the paper. Subsection 3.2 gives algorithms for different versions of the MinSum barrier coverage. The paper concludes with several proposals for possible extensions as well as related open problems.

Table 1. MinSum problem results for n sensors with barrier of length L and R the total sensor ranges

	identical ranges		non-identical ranges
coverage	contiguous	non-contiguous	
$R < L$	$O(n)$	$O(n)$	NP-complete
$R = L$	$O(n)$	not applicable	NP-complete
$R > L$	$O(n^2)$	not applicable	NP-complete

2 NP Completeness Results

In this section we consider the MinSum problems for sensors with non-identical ranges.

Theorem 1. *Let $S_1, S_2, \ldots, S_n$ be n sensors with ranges $r_1, r_2, \ldots, r_n$ located on a line containing segment $[0, L]$, in initial positions $x_1 \leq x_2 \leq \ldots \leq x_n$, $\sum_{i=1}^{n} r_i = R \geq L$, and k be a given number. The problem of calculating the movements of sensors on the line so that the sensors cover the segment $[0, L]$ and the sum of movement of the sensors is less than k is NP-hard.*

Proof. We give the proof only for the case $R = L$. The proof for the case $R > L$ is very similar. We prove it by reducing the 3-partition problem (see ([8])) to the problem of covering the line segment $[0, L]$ with sensors such that the sum of the movements of the sensors is minimized. The 3-partition problem is defined as follows: we are given a multiset $S = \{a_1 \geq a_2 \geq \cdots \geq a_n\}$ of $n = 3m$ positive integers such that $B/4 < a_i < B/2$ for $1 \leq i \leq n$ and $\sum_{i=1}^{n} a_i = mB$ for some B. The problem is to decide whether S can be partitioned into m triples T_1, $T_2, \ldots, T_m$ such that the sum of the numbers in each triple is equal to B.

Let $L = mB + m - 1$ and $k = m(m + 1)(B + 1)$. Consider a sensor movement problem as shown in Figure 1. We have a sensor S_i of range $a_i/2$ for every $1 \leq i \leq n$ positioned at $-a_i/2$. In addition, we have $m - 1$ blocks of sensors of range $1/(2k)$, each block containing k sensors. Each block of these sensors covers a subinterval of $[0, L]$ of size 1, leaving m gaps of size B on the line segment $[0, L]$. Clearly, any solution that covers the segment $[0, L]$ requires that all sensors are moved inside the segment without leaving there any gaps or overlaps, and any solution can be interpreted as a partition of S into subsets, with sensors with range $1/(2k)$ separating the subsets in the partition.

If there is a partition of S into m triples $T_1, T_2, \ldots, T_m$, the sum of each triple being B, then there is a solution to the movement of the sensors such that we only move sensors $S_1, S_2, \ldots, S_n$ and the three sensors corresponding to triple T_i are moved to fill the ith gap in the interval $[0, L]$. The sum of the moves of the three sensors corresponding to T_i into ith gap is less than $iB + (i - 1)$, and the sum of the moves of all sensors for all triples is thus less than $m(m+1)(B+1)/2 = k/2$ in this case. If such a partition does not exist, then any solution to the coverage of the line segment $[0, L]$ corresponds to either: (a) a partition of S into m subsets in which the sum of elements in at least two subsets differs from B by at least

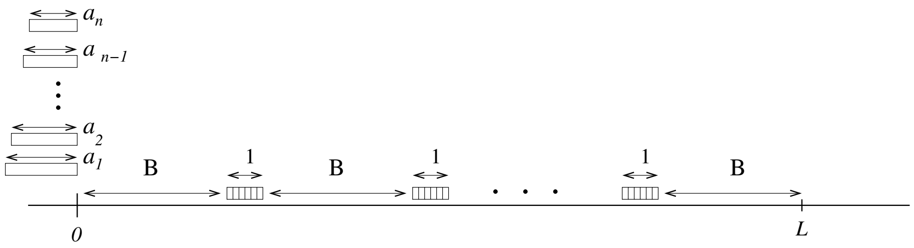


Fig. 1. Sensor arrangement for proving the NP completeness of the MinSum problem

1 in which case we need to move all the sensors in at least one block of sensors with range $1/(2k)$ at least distance 1; or (b) a partition of S into less than m or more than m subsets and this would require one to move at least k of the sensors with range $1/(2k)$ by a distance of 1 or more.

However, moving k of the sensors with range $1/(2k)$ by 1 increases the sum of movements of sensors by at least $k = m(m+1)(B+1)$. Thus the sum of movement of the sensors is less than $k/2$ if and only if the 3-partition problem has a solution. It remains to show that the transformation from the 3-partition problem to the sensor movement problem is polynomial.

Since 3-partition is strongly NP-complete ([8]), we may assume that the values $a_1, a_2, \dots, a_n$ are bounded by a polynomial cn^j for some constants c and j. Therefore, $B \le 3c_1 n^j$ and $k \le c_2 n^{j+2}$ for some constants c_1 and c_2. Our reduction uses $n + k(m-1)$ sensors and $n + km \le n + m^2(m+1)B \le c_3 n^{j+3}$ for some constant c_3. The 3-partition problem can be represented using $O(n \log n)$ bits. In the corresponding sensor movement problem we need $O(n \log n)$ bits for the positions and sizes of sensors $S_1, S_2, \dots, S_n$ and we need $O(\log k) = O(\log n)$ bits to represent the position and size of each sensor of size $1/(2k)$. Thus we need $O(n^{j+3} \log n)$ bits to represent the corresponding sensor movement problem, which shows that the transformation is polynomial. □

One can similarly show that when $R < L$ the problem of calculating the movements of sensors on the line so that the sensors give a maximal coverage the segment $[0, L]$ and the sum of movement of the sensors is less than k is NP-hard.

Remark 1. The proof of the above theorem also shows that if $NP \neq P$ there is no polynomial 2-approximation algorithm for the MinSum problem, since the result of a 2-approximation algorithm for sensor movements would be less than k if and only if the corresponding 3-partition problem has a solution. Clearly, the proof can be modified to show the non-existence result for any constant factor approximation algorithm.

3 Sensors with Identical Ranges

In view of the NP-complete results of the previous section, we consider in this section the MinSum problem for sensors of identical range, say r.

3.1 Ordering Property of Optimal Configurations

An important observation that will be useful in the MinSum optimization problem concerns the order of final positions of sensors in an optimal configuration. It is shown below that there exists an optimal solution of the MinSum problem so that the final destinations of sensors preserve the initial ordering of sensors. In other words, two sensors on their way to the optimal locations do not have to cross paths.

Lemma 1. *Let $x_i \leq x_j$ and $y_i > y_j$ be real numbers.*

$$|x_i - y_i| + |x_j - y_j| \geq |x_i - y_j| + |x_j - y_i| \tag{4}$$

Proof. It can be easily proved by considering the five possible arrangements of values x_i, x_j, y_i, y_j. □

Lemma 1 implies that there exists an optimal solution of the barrier coverage problem which preserves the initial order of position of the sensors.

Corollary 1 (Order Preservation). *For any of the MinSum optimization problems, if $x_1 \leq x_2 \leq \cdots \leq x_n$ are the initial positions of sensors $S_1, S_2, \ldots, S_n$ of identical range then there exists an optimal solution of the problem such that the final destinations of sensors satisfy $y_1 \leq y_2 \leq \cdots \leq y_n$, respectively.*

According to the order preservation lemma the MinSum problem is trivial when $R = L$ and thus we consider below only the cases $R > L$ and $R < L$.

3.2 Algorithms for MinSum Barrier Coverage

We now propose several efficient algorithms for sensors with identical ranges. We start with the Contiguous MinSum problem, $R < L$. We first give an $O(n)$ algorithm for maximal contiguous coverage of the line with n sensors which minimizes the sum of the movements of the sensors.

We say that sensors S_i and S_{i+1} are in *attached position* if the difference between their positions is equal to $2r$, i.e., there is no gap or overlap between the two sensors.

Lemma 2 (On an infinite line). *Let $S_1, S_2, \ldots, S_n$ be n sensors with identical range r located on a line in initial positions $x_1 \leq x_2 \leq \ldots \leq x_n$ with $R < L$. There is an $O(n)$ algorithm that calculates the movements of sensors on the line so that the sensors cover a segment of size $2rn$ and the sum of movements of the sensors is minimized.*

Proof. Let $y_1, y_2, \ldots, y_n$ be positions on the line such that $\sum_{i=1}^{n}(|x_i - y_i|$ is minimal among all such possible assignment of values. According to Lemma 1, there is an optimal solution such that $y_1 < y_2 < \ldots < y_n$. Furthermore, since the sensors cover a contiguous segment of the line, we have $y_i = y_1 + 2(i-1)r$ for $2 \leq i \leq n$. In fact, our algorithm determines a solution of this type.

Consider the possibility that the sensors $S_1, S_2, \ldots, S_n$ have moved to positions $y_1 = 0, y_2 = 2r, \ldots, y_n = 2(n-1)r$, respectively, on the line, i.e., the sensor S_1 is moved to location 0 and the other sensors are moved to attached positions following the initial order of sensors. Then the values $-x_1, 2r - x_2, \ldots, 2(n-1)r - x_n$ give the displacements of the sensors. Let l_1 be the number of sensors that move left, l_2 be the number of sensors that move right, l_3 be the number of sensors that remain stationary in this assignment, and $shifts_0$ be the sum of the absolute values of all shifts when S_1 is in position 0. If $l_1 > l_2 + l_3$ then consider the assignment of positions to sensors by shifting all positions of the sensor to the right by c where c is the smallest negative shift. In this assignment all left shifts of sensors are decreased by c, all the right shifts of sensors are increased by c and the zero shifts become c. Thus in this assignment the sum of the absolute values of all shifts, say $shifts_c$, is equal to $shifts_0 - c(l_1 - l_2 - l_3)$, which is smaller than $shifts_0$. Similarly, if $l_2 > l_1 + l_3$ then the assignment of positions by shifting positions of all sensors to the left by c, where c is the smallest positive shift, we obtain an assignment of positions to sensors in which the sum of the absolute values of all shifts is smaller than $shifts_0$. Thus we obtain an optimal assignment of positions to sensors when $l_1 \leq l_2 + l_3$ and $l_2 \leq l_1 + l_3$. By finding the median of $-x_1, 2r - x_2, \ldots, 2(n-1)r - x_n$ and shifting the configuration to the right or left by the median value so that the median of all the shifts becomes 0 we obtain an assignment that minimizes the sum of shifts. Clearly, the value of the median of n values can be calculated in $O(n)$ and so can the n shifted values of positions of the sensors. □

When sensors are in the positions determined by the algorithm of the previous lemma they give a maximal contiguous coverage of a segment of a line which minimizes the sum of all shifts, but not necessarily of the segment $[0, L]$. However, when $R < L$ we can easily modify the solution above so as to solve the MinSum contiguous problem by shifting the solution into the segment $[0, L]$ if the segment covered by the sensors from Lemma 2 is not in it already and we obtain the following theorem.

Theorem 2. *Let $I = [0, L]$ be a line segment and $S_1, S_2, \ldots, S_N$ be sensors with identical range r located on a line in initial positions $x_1 \leq x_2 \leq \ldots \leq x_n$ and $R < L$. There is a $O(n)$ algorithm to solve the Contiguous MinSum problem for $R < L$.*

MinSum problem, $R > L$: The optimal solution of this problem is more difficult to obtain, since it does not correspond to sensors being in attached positions. We give below an algorithm for the MinSum problem that is of time complexity $O(n^2)$. This algorithm is more complex to state and to verify its correctness and thus we break it into several lemmas.

Given an instance of the barrier coverage problem with $R > L$ we enumerate the gaps in the interval $[0, L]$ as $g_1.g_1, \ldots g_l$ from the left. Informally, the algorithm considers the gaps in the given interval $[0, L]$ in the left to right order. It eliminates each gap by removing the overlaps to the left and right of the gap in the inside-out manner, removing at every step the overlap whose "cost" is the lowest among the available gaps. The cost is related to the number of sensors whose positions must be shifted when eliminating the overlap.

Let A be an algorithm that solves the MinSum problem. We say that the algorithm is *locally optimal* with respect to gaps $g_1, g_2, \ldots, g_k$, $1 \leq k \leq l$, if the sum of moves of the sensors needed to eliminate gaps g_1, g_2, g_k is minimal, without creating any new gap or increasing the size of the other gaps.

Consider an instance of the MinSum problem for $R > L$ with sensors of identical range. We enumerate the overlaps of the sensor ranges from left to right as $o_1, o_2, \ldots, o_k$, each overlap is either the interval corresponding to the nonempty intersection of the ranges of two consecutive sensors, if the intersection is inside I, or the nonempty intersection of the range of a sensor with $(-\infty, 0)$ or (L, ∞). Thus all of a sensor range outside of the interval $[0, L]$ it treated as an overlap. When moving a sensor, say S_i, in order to achieve a contiguous coverage of the interval $[0, L]$, we assign to it a real number d_i, indicating the difference between the present and initial position and we call it its *shift value*. Thus negative values correspond to moving sensors to the left, while positive values correspond to moving sensors to the right.

Clearly, at any stage of the algorithm the sum $\sum_{i=1}^{n} |d_i|$ gives the cost of the moves of sensors performed so far. See Figure 2 for a possible initial and final configuration, including the shifts and overlaps.

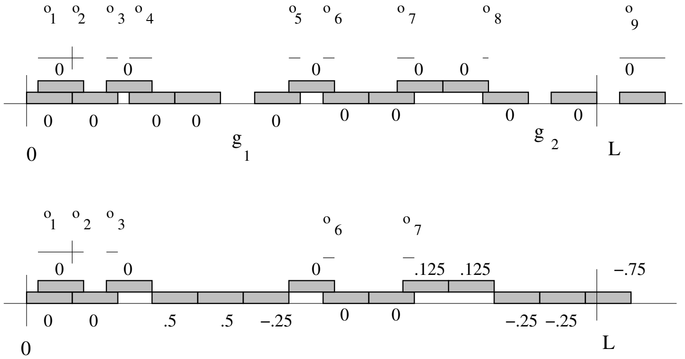


Fig. 2. Example of initial and final configurations

We first provide some claims concerning the necessary properties of any locally optimal solution of the problem, which will also form the foundations of our algorithm.

Lemma 3. *Consider a locally optimal solution with respect to gaps $g_1, g_2, \ldots, g_k$, $1 \leq k \leq l$. Then in this solution, no new overlaps are created. Furthermore, an initial overlap inside $[0, L]$ cannot be moved left or right and its size cannot be increased. Thus any locally optimal algorithm can only eliminate an existing overlap or make it smaller by moving its left sensor to the left, or (and) the right sensor to the right, or it leaves the overlap exactly as it is initially.*

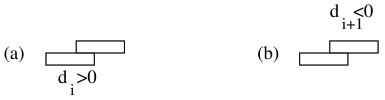


Fig. 3. Forbidden sensor configurations in a locally optimal solution

Proof. Let S_i and S_{i+1} be two overlapping sensors in a locally optimal solution. If $d_i > 0$, we could decrease the cost of the solution by moving S_i slightly to the left (see (a) of Figure 3), since S_i moved too much to the right. Similarly if $d_{i+1} < 0$, we could decrease the cost of the solution by moving S_{i+1} slightly to the right (see (b) of Figure 3). Thus the configurations in Figure 3 cannot occur in a locally optimal solution. However, creation of a new overlap necessarily corresponds to (a) or (b) of Figure 3. Similarly, moving an initial overlap left or right, or increasing its size necessarily corresponds to either (a) or (b) of Figure 3. □

Lemma 4. *Consider a locally optimal solution with respect to gaps* $g_1, g_2, \ldots, g_k$, $1 \leq k \leq l$. *Let* $S_i, S_2, \ldots, S_j$ *be the sensors in the portion of the solution which does not contain gaps any more and let* $d_i, d_{i+1}, \ldots, d_j$ *be the sequence of shift values of these sensors. If for some* m, $1 \leq m \leq j-1$ *we have* $d_m > 0$ *and* $d_{m+1} \geq 0$, *or* $d_m \leq 0$ *and* $d_{m+1} < 0$, *or* $d_m > 0$ *and* $d_{m+1} < 0$, *then* S_m *and* S_{m+1} *are in attached position in the solution.*

Proof. Since S_m and S_{m+1} are in the part where gaps were eliminated, they are either attached or they overlap. An overlap of these two sensors in either case would be one of the forbidden configurations in Figure 3. □

Lemma 5. *Let* $S_i, S_{i+1}, \ldots, S_j$ *be a sequence of sensors that in the initial configuration does not contain any gap but it contains overlaps* $o_k, o_{k+1}, \ldots, o_l$. *Let* m_t *be the integer such that* S_{m_t} *and* S_{m_t+1} *are the two sensors that form overlap* o_t, $k \leq t \leq j$. *If in a locally optimal solution overlap* o_t *is either eliminated or made smaller by moving the right sensor* S_{m_t+1} *to the right, then in this solution all overlaps* $o_{t+1}, \ldots, o_l$ *have been eliminated by moving the sensors* $S_{m_t+1}, S_{m_t+2}, \ldots, S_j$ *to the right into attached positions. If in a locally optimal solution overlap* o_t *is either eliminated or made smaller by moving sensor* S_{m_t} *to the left, then in this solution all overlaps* $o_k, o_{k+1} \ldots, o_{t-1}$ *have been eliminated by moving the sensors* $S_i, S_{i+1}, \ldots, S_{m_t}$ *to the left into attached positions.*

Proof. If overlap o_t is either eliminated or made smaller by moving S_{m_t+1} to the right, then all sensors to the right of it until the next right gap must be moved to the right so that we do not create a new overlap, which is forbidden by Lemmas 3. Since the shift values of $S_{m_t+1}, S_{m_t+2}, \ldots, S_j$ are all positive, they must be all in attached positions by Lemma 4. The proof of the second part of the lemma is analogous. □

Lemmas 3, 4, 5 above form the basis for the design of our MinSum algorithm.

Main Algorithm: MinSum algorithm for $R > L$:
Our algorithm proceeds by closing the gaps from left to right producing a locally optimal solution. For each gap, say g, we search to the left and to the right from the gap to find the "closest overlaps", say o_i and o_j of sensor ranges on each side of the gap that can be used to shrink the gap. For each of these two overlaps we calculate the cost of using o_i or o_j to shrink the gap, the cost being equal to the *number of sensors* that are being shifted. At any time the cheapest of the two overlaps is used to shrink the gap. The sensors that are moved in the shrinking process are put in attached position, unless the gap is smaller than the overlap. The pseudocode for the main algorithm is as follows.

Algorithm MinSum

```
Input:   $L$ and the initial positions $x_1, x_2, \ldots, x_n$ of sensors (assumed sorted).
Output:  The final positions $y_1, y_2, \ldots, y_n$ of sensors for the contiguous coverage
         of the interval $[0, L]$ that minimize the sum of movements.
1:  initialize array $d_1, d_2, \ldots, d_n$ of sensor shifts to 0;
2:  scan $x_1, x_2, \ldots, x_n$ and calculate the sequence of overlaps $o_1, o_2, \ldots, o_k$,
    the sequence of gaps $g_1, g_2, g_l$, and their sizes;
3:  for $i := 1$ to $l$ do //eliminate Gap $i$
      repeat
        find $o_j$, the closest overlap left of $g_i$ and its cost w.r.t. $g_i$;
          (if there is no such overlap, set the cost to $\infty$).
        find $o_k$, the closest overlap right of $g_i$ and its cost w.r.t. $g_i$;
          (if there is no such overlap, set the cost to $\infty$).
        if $(cost(o_j) \leq cost(o_k))$ then //right shift is done
          { if $size(o_j) < size(g_i)$ then $c := size(o_j)$ else $c := size(g_i)$;
            $size(g_i) := size(g_i) - c$;
            $size(o_j) := size(o_j) - c$;
            add $c$ to the values in array $d$ of sensors between $o_j$ and $g_i$;
          }
        else //left shift is done
          { if $size(o_k) < size(g_i)$ then $c := size(o_k)$ else $c := size(g_i)$;
            $size(g_i) := size(g_i) - c$;
            $size(o_k) := size(o_k) - c$;
            subtract $c$ from the values in array $d$ of sensors between $g_i$ and $o_k$;
          }
      until $size(g_i) = 0$;
4:  for $i := 1$ to $n$ do
      $y_i = x_i + d_i$; //the final positions of the sensors.
```

Now we can state the main theorem.

Theorem 3. *Let $S_1, S_2, \ldots, S_n$ be sensors with identical range r located on a line in initial positions $x_1 \leq x_2 \leq \ldots \leq x_n$ (not restricted to lie inside the segment $[0, L]$) and $R > L$. Algorithm MinSum above solves this instance of the MinSum problem in time $O(n^2)$.*

When calculating the cost of a shift for the overlap o_i on the left of the present gap, we have to take into account the fact that shifting those sensors to the right whose shift is negative at present is actually equivalent to undoing a left shift that was done when removing another gap to the left of the present gap. Thus shifting these sensors with negative moves to the right is decreasing the cost of the sum of movements done so far. Another factor that needs to be considered is the difference between overlaps of sensors inside interval $[0, L]$ and overlaps that are outside this interval. Therefore, we need to define the cost of moving a portion p of overlap s_j to the right and left, respectively. Due to the page limit, these Definitions and the detailed proof of Theorem 3 are given in the full paper.

4 Conclusion and Open Problems

We have studied the barrier coverage problem for a wireless sensor network when the perimeter to be covered is a finite line segment. In view of the results, an interesting problem is to study the barrier coverage by sensors with limited number of different ranges. For the case of a one dimensional barrier, one could consider the problem of barrier k coverage, whereby each intruder should be detected by at least k different sensors, for some fixed $k > 1$. Also, the possibility that the perimeter consists of several line sub-intervals could be investigated. Another class of problems concerns extensions to higher dimensions.

The two dimensional version of the problem is wide open. Specifically, one might consider other geometric barriers, e.g., circular barriers, convex barriers or boundaries of simple polygons. Also one might consider other types of sensor movements, e.g., the movement of the sensors towards the globally optimal position on the circular barrier may proceed through the interior of the circle as opposed to only moving on the perimeter.

Another interesting class of problems would be to examine the above questions in light of a "decentralized" sensor communication model. Finally, it would be interesting to investigate how to optimize other more realistic energy consumption metrics, e.g., sum of squares of movements of all the sensors.

References

[1] Balister, P., Bollobas, B., Sarkar, A., Kumar, S.: Reliable density estimates for coverage and connectivity in thin strips of finite length. In: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, pp. 75–86 (2007)

[2] Bhattacharya, B., Burmester, M., Hu, Y., Kranakis, E., Shi, Q., Wiese, A.: Optimal Movement of Mobile Sensors for Barrier Coverage of a Planar Region. In: Yang, B., Du, D.-Z., Wang, C.A. (eds.) COCOA 2008. LNCS, vol. 5165, pp. 103–115. Springer, Heidelberg (2008)

[3] Cabello, S., Giannopoulos, P., Knauer, C., Rote, G.: Matching point sets with respect to the Earth Mover's Distance. Computational Geometry: Theory and Applications 39(2), 118–133 (2008)

[4] Chen, A., Kumar, S., Lai, T.H.: Designing localized algorithms for barrier coverage. In: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, pp. 63–74 (2007)

[5] Cohen, S.: Finding Color and Shape Patterns in Images. PhD Thesis, Stanford University, Dept. of Computer Science (1999)

[6] Czyzowicz, J., Kranakis, E., Krizanc, D., Lambadaris, I., Narayanan, L., Opatrny, J., Stacho, L., Urrutia, J., Yazdani, M.: On minimizing the maximum sensor movement for barrier coverage of a line segment. In: Ruiz, P.M., Garcia-Luna-Aceves, J.J. (eds.) ADHOC-NOW 2009. LNCS, vol. 5793, pp. 194–212. Springer, Heidelberg (2009)

[7] Demaine, E.D., Hajiaghayi, M.T., Mahini, H., Sayedi-Roshkhar, A.S., Oveisgharan, S., Zadimoghaddam, M.: Minimizing movement. ACM Transactions on Algorithms (TALG) 5(3), 1–30 (2009)

[8] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. WH Freeman, San Francisco (1979)

[9] Klein, O., Veltkamp, R.C.: Approximation Algorithms for Computing the Earth Mover's Distance Under Transformations. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 1019–1028. Springer, Heidelberg (2005)

[10] Kumar, S., Lai, T.H., Arora, A.: Barrier coverage with wireless sensors. Wireless Networks 13(6), 817–834 (2007)

[11] Li, X., Frey, H., Santoro, N., Stojmenovic, I.: Localized sensor self-deployment with coverage guarantee. ACM SIGMOBILE Mobile Computing and Communications Review 12(2), 50–52 (2008)

[12] Yang, S., Li, M., Wu, J.: Scan-based movement-assisted sensor deployment methods in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 18(8), 1108–1121 (2007)

[13] Zou, Y., Chakrabarty, K.: A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. IEEE Trans. Comput. 54(8), 978–991 (2005)

On Alarm Protocol in Wireless Sensor Networks

Jacek Cichoń, Rafał Kapelko, Jakub Lemiesz, and Marcin Zawada

Institute of Mathematics and Computer Science,
Wrocław University of Technology, Poland
{Jacek.Cichon,Rafal.Kapelko,Jakub.Lemiesz,
Marcin.Zawada}@pwr.wroc.pl

Abstract. We consider the problem of efficient alarm protocol for ad-hoc radio networks consisting of devices that try to gain access for transmission through a shared radio communication channel. The problem arise in tasks that sensors have to quickly inform the target user about an alert situation such as presence of fire, dangerous radiation, seismic vibrations, and more. In this paper, we present a protocol which uses $O(\log n)$ time slots and show that $\Omega(\log n/\log\log n)$ is a lower bound for used time slots.

Keywords: wireless sensor network, radio channel, alarm protocol, oblivious leader election.

1 Introduction

We consider wireless sensor networks (WSNs) consisting of small programmable devices equipped with radio-enabled sensing capabilities and have been applied in information gathering ranging from the environment temperature, radiation, the presence of fire, seismic vibrations, and more. WSNs compared with wired networks provide many advantages in the deployment, cost and size. Wireless technology enables users to set up a network quickly, more it enables them to set up a network where it is inconvenient or impossible to wire cables. Moreover, common WSNs can consist of up to several hundreds of those small devices.

The most straightforward application of a WSN is to monitor remote or hostile environments. For example, a remote forest area can be monitored by deploying hundreds of sensors that configure themselves to form a network and immediately report upon detection of any event such as fire. Moreover, such networks can be easily extended by simply adding more devices without any rework or complex reconfiguration. The sensor nodes can ideally run for over a year on a single set of batteries. Given the cost of these sensor nodes, it is not feasible to discard dead sensor nodes, and it is also not possible to replace the batteries on these sensor nodes. Hence, there is a great need for energy-efficient protocols that can greatly reduce power consumption and increase the lifetime of wireless sensor nodes.

In this paper, we address the problem of designing protocol for an alert situation observed by the sensor nodes in a WSN and sending this information toward the sink that acts as a collector and an interface to the external world. The traffic is usually forwarded over multi-hops, that is, each node acts as a relay/router for some nodes

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 43–52, 2010.
© Springer-Verlag Berlin Heidelberg 2010

farther away to the sink. However, in this paper we restrict our attention to a quarter size sensors e.g. MICA2DOT. We even assume that such sensors cannot listen to the channel or receive messages of any kind, also they have a very limited energy supply. Due to this restriction sensors are incapable of forwarding messages. Therefore, we have to assume that the network of such sensors is single-hop or we have two classes of wireless sensors; the first class of sensors are devices equipped with full communication features such as receiving, sending and forwarding messages, and the second class of sensors are our very weak tiny devices, capable only of sensing and sending information in a single-hop fashion. For the first class of sensors, which is the backbone of the network, we can use well known convergecast algorithms [1,2,3,4,5]. In this paper we deal with the second class of devices.

In Sect. 3 we shall consider the sequence $p = ((\frac{1}{2})^i)_{i=0,\ldots,L}$ of probabilities of length $L = \lceil \log_2(n) \rceil + 1$, where n denote the number of sensors and we assume that at ith slot stations try to transmit with probability $(\frac{1}{2})^i$. Let $\mathrm{SCC}_{n,k}$ denotes the event of the successful transmission when $k \in \{1, \ldots, n\}$ stations are activated and let $\Pr[\mathrm{SCC}_n]$ = $\min\{\Pr[\mathrm{SCC}_{n,k}] : k = 1, \ldots, n\}$. The same sequence was investigated in a series of papers of Nakano and Olariu. In 2000 in [6] authors claimed that $\Pr[\mathrm{SCC}_n] \geq 0.6$ but they omit the proof (due to page limitations). In 2001 in the next paper [7] authors sketch a proof of inequality $\Pr[\mathrm{SCC}_n] \geq 1/(2\sqrt{e}) \approx 0.303$. However they do not observe that the inequality $(1-x)^{n-1} > e^{-nx}$, which is used in the proof, does not hold for all $x \in [0, 1]$ and $n > 1$. Next, in 2002 in [8] they claim that $\Pr[\mathrm{SCC}_n] \geq 0.5$ and for the proof they refer to the previous paper [6]. Let us remark that numerical calculation for small values of n (say $n < 1000$) confirms that $\Pr[\mathrm{SCC}_n] \geq 0.6$.

In Sect.3 we shall prove that $\Pr[\mathrm{SCC}_n] > 0.575$ and to the best of our knowledge this will be the first published proof of this fact. In Sect. 4 we analyze the lower bound on number of slots required by our protocols for successful transmission of alert messages with a controllable probability of success and we show that $\Omega(\log n / \log \log n)$ is asymptotically a lower bound for the number of necessary time slots. We will use the following simple lemma which we leave without proof:

Lemma 1. *Suppose A is an event, $f \geq 1$, $\Pr[A] \geq \lambda > 0$ and let $A_1, \ldots, A_m$ be independent copies of A. Then*

$$\left(m \geq \frac{\log f}{\log \frac{1}{1-\lambda}} \right) \longrightarrow \Pr[A_1 \cup \ldots \cup A_m] \geq 1 - \frac{1}{f} \ .$$

Related work. The alarm problem is similar to the wake-up problem [9,10,11]. In the wake-up problem it is assumed that any subset of sensors wake up spontaneously at arbitrary times and awake the remaining sensors. However, in the alarm problem it is not important to inform other sensors about a dangerous situation. Only the sink should be informed as soon as possible. Therefore, we want to design such a protocol, which is able to inform the sink with a minimal time complexity. Our considerations are directly related to the previously mentioned papers [6,7,8].

2 Model Description

We consider a *wireless sensor network* consisting of n processing units, called *sensors* with limited power and one distinguished station called the *sink* with an unlimited power. The sensors communicate directly with the sink through a shared radio channel and a transmission succeeds if exactly one station sends at a time. We assume that sensors **can only send messages** and that they cannot listen or recognize the state of the channel. We consider only *single-hop* networks in which each station can directly communicate with the sink through a shared communication channel. We also assume that stations are synchronized and that the time is divided into short time-slots $S_0, \ldots, S_L$ of the same length Δ. There is also a fixed vector $p_0, \ldots, p_L$ of probabilities.

Let $A \subseteq \{1, \ldots, n\}$ be a set of sensors which detect an alert and let $k = |A|$. Each sensor from the set A try to send an alert message in the ith slot S_i independently with probability p_i. The transmission will be successful if in some slot S_i precisely one sensor from A will transmit. Nakano and Olariu (see [6]) call this variant of leader election algorithm an oblivious one: all stations use the same probabilities which are fixed beforehand and does not depend on the history.

Our goal is to find a reasonable small L and a vector $p_0, \ldots, p_L$ of probabilities which will guarantee a successful transmission of an alert with a probability at least $1 - \frac{1}{f}$ where $f > 1$ is a given fixed parameter and $1 \leq k \leq n$ is arbitrary.

3 Upper Bound

By $n \geq 1$ we denote the number of stations. We divide time into $L+1$ slots. At ith time slot each station decides to transmit the alert message independently with the probability $p_i = (1/2)^i$ for $i = 0, \ldots, L$. Let $\mathrm{SCC}_{L,n,k}$ denote the event of the successful transmission when $k \in \{1, \ldots, n\}$ stations are activated. Then $\Pr[\mathrm{SCC}_{L,n,1}] = 1$ and

$$\Pr[\mathrm{SCC}_{L,n,k}] = 1 - \prod_{i=0}^{L} \left(1 - \binom{k}{1} \frac{1}{2^i} \left(1 - \frac{1}{2^i} \right)^{k-1} \right)$$

for $k > 1$. Finally we put

$$\Pr[\mathrm{SCC}_{L,n}] = \min\{\Pr[\mathrm{SCC}_{L,n,k}] : k = 1, \ldots, n\} \ .$$

Theorem 1. *If* $L = \lceil \log_2 n \rceil + 1$ *then*

$$\Pr[\mathrm{SCC}_{L,n}] \geq 1 - \frac{3}{4}(1 - \frac{1}{2}e^{-1/2})(1 - \frac{1}{4}e^{-1/4}) \approx 0.579 \ .$$

Proof. Let $\lambda = 1 - \frac{3}{4}(1 - \frac{1}{2}e^{-1/2})(1 - \frac{1}{4}e^{-1/4})$. Notice that $\lambda \approx 0.579$. Let us fix $k \in \{1, \ldots, n\}$. Then there exists $i \in \{0, \ldots, L-1\}$ such that

$$2^{i-1} < k \leq 2^i \ . \tag{1}$$

We shall consider the following three cases separately: $i = 0$, $i = 1$ and $2 \leq i \leq L-1$.

Case 1: If $i = 0$ then $(2^{i-1}, 2^i] = (1/2, 1]$ so $k = 1$ and $\Pr[\mathrm{SCC}_{L,n,1}] = 1 > \lambda$ for all $n \geq 1$.

Case 2: If $i = 1$ then $(2^{i-1}, 2^i] = (1, 2]$, therefore $k = 2$ and

$$\Pr[\mathrm{SCC}_{L,n,2}] \geq 1 - (1 - f_1(2)) \cdot (1 - f_2(2)) = 1 - \frac{5}{16} = \frac{11}{16}$$

for $n \geq 2$. Notice that $11/16 = 0.6875 > \lambda$.

Case 3: Suppose that $2 \leq i \leq L-1$. Let us consider functions

$$f_j(k) = k \cdot \frac{1}{2^j} \cdot \left(1 - \frac{1}{2^j}\right)^{k-1} \qquad (k \geq 1,\ 1 \leq j \leq L)\ .$$

If $j > 0$ then the function f_j is unimodal (with the maximum at the point $k = 1/\log(1/(1-(1/2)^j)))$ hence the minimum of the function f_j on interval $(2^{i-1}, 2^i]$ is achieved at one of the edges of this interval.

Let $l_j(i) = f_j(2^{i-1})$ and $r_j(i) = f_j(2^i)$ for $j = i-1, i, i+1$. From the inequality $(1 - 1/x)^x \leq 1/e$ we get

$$\frac{r_{i-1}(i)}{l_{i-1}(i)} = 2\left(1 - 2^{-(i-1)}\right)^{2^{i-1}} \leq \frac{2}{e}\ .$$

On the other hand we have

$$\frac{l_i(i)}{r_i(i)} = \frac{1}{2}\left(1 - 2^{-i}\right)^{2^i \cdot (-1/2)}, \qquad \frac{l_{i+1}(i)}{r_{i+1}(i)} = \frac{1}{4}\left(1 - 2^{-i}\right)^{2^i \cdot (-1/4)}\ .$$

Notice that those functions are decreasing, so the maximum is achieved for $i = 2$. Thus, $l_i(i)/r_i(i) \leq 8/9 < 1$, $l_{i+1}(i)/r_{i+1}(i) \leq 32/49 < 1$ for $i \geq 2$. Therefore, we deduce that minimum of the functions $f_{i-1}(x)$, $f_i(x)$, $f_{i+1}(x)$ on the interval are achieved at points 2^i, 2^{i-1}, 2^{i-1} respectively, and are equal to $r_{i-1}(i)$, $l_i(i)$, $l_{i+1}(i)$ i.e. $f_{i-1}(2^i)$, $f_i(2^{i-1})$, $f_{i+1}(2^{i-1})$.

Next, we notice that the functions $l_x(x), l_{x+1}(x)$ are decreasing and $r_{x-1}(x)$ is increasing for $x \geq 2$. This can be checked by inspecting the sign of the derivative (see Appendix A). Moreover

$$\lim_{x\to\infty} l_x(x) = \lim_{x\to\infty} \frac{1}{2}\left(1 - 2^{-x}\right)^{2^{x-1}-1} = \frac{1}{2}e^{-1/2}\ .$$

Hence $l_x(x) > (1/2)e^{-1/2}$ and therefore for each $u \in (2^{i-1}, 2^i]$ we have $f_i(u) > (1/2)e^{-1/2}$. Similarly, we have

$$\lim_{x\to\infty} l_{x+1}(x) = \frac{1}{4}e^{-\frac{1}{4}}, \qquad r_{x-1}(x) \geq \frac{1}{4} \text{ for } x \geq 2$$

Algorithm 1. Randomized Alarm Algorithm RAA(n,f,T_0,Δ)

```
 1: if NOT ALERT then
 2:     EXIT
 3: end if
 4: wait until $(GetTime() \geq T_0)$
 5: SendMessage()
 6: $L := \lceil \log_2(n) \rceil + 1$
 7: $r := \lceil 1.1553 \log f \rceil$
 8: for $j := 0$ to $r - 1$ do
 9:     for $i := 1$ to $L$ do
10:         wait until $(GetTime() \geq T_0 + (i + j * L) * \Delta)$
11:         if $(Random() < \max(1/n, (1/2)^i))$ then
12:             SendMessage()
13:         end if
14:     end for
15: end for
```

and for each $u \in (2^{i-1}, 2^i]$ we have $f_{i-1}(u) \geq \frac{1}{4}$ and $f_{i+1}(u) > \frac{1}{4}e^{-\frac{1}{4}}$. Notice that $\Pr[\mathrm{SCC}_{L,n,k}]$ is greater than or equal to

$$1 - (1 - f_{i-1}(k))(1 - f_i(k))(1 - f_{i+1}(k))$$

for $2 \leq i \leq L - 1$ and $0 \leq f_j(k) \leq 1$. Therefore theorem is proved. □

By Thm. 1 we are able successfully send an alert message with a probability at least $1 - \frac{3}{4}\left(1 - \frac{1}{2}e^{-1/2}\right)\left(1 - \frac{1}{4}e^{-1/4}\right) \approx 0.579$ in $\lceil \log_2 n \rceil + 2$ time-slots. However, we are interested in sending an alert message with probability at least $1 - \frac{1}{f}$ for some fixed $f > 1$. We shall achieve this goal by repeating the sequence $\left((\frac{1}{2})^i\right)_{i=1,\ldots,\lceil \log_2 n \rceil + 1}$ a sufficient number of times to obtain the needed probability of success. Namely, Lemma 1 implies that a sufficient total number of time-slots required to send an alert message with probability at least $1 - \frac{1}{f}$ is equal to

$$\left\lceil \frac{\log f}{\log \frac{1}{1-\lambda}} \right\rceil \cdot (\lceil \log_2 n \rceil + 1) + 1 \approx 1.1553 \cdot \log f \cdot (\lceil \log_2 n \rceil + 1) + 1 \qquad (2)$$

where $\lambda = 1 - \frac{3}{4}\left(1 - \frac{1}{2}e^{-1/2}\right)\left(1 - \frac{1}{4}e^{-1/4}\right)$.

Based on the above discussion we build a Randomized Alarm Algorithm (see Algorithm 1). The small correction of probabilities in line 11 of its pseudo-code is motivated by Lemma 2 from the next section. The following theorem summarize its basic property:

Theorem 2. *For each $n \geq 1$ and $f > 1$ the Randomized Alarm Algorithm* RAA *sends successfully an alert message in*

$$\lceil 1.1553 \cdot \log f \rceil \cdot (\lceil \log_2 n \rceil + 1) + 1$$

time slots with probability at least $1 - \frac{1}{f}$ for arbitrary number of activated stations.

It is worth to mention that in $\mathrm{RAA}(n, f, T_0, \Delta)$ algorithm each station which want to transmit an alert message sent a signal in no more than $2\lceil 1.1553 \cdot \log f \rceil$ time-slots on average.

4 Lower Bound

Let $\boldsymbol{p} = (p_i)_{i=1,\ldots,L}$ be a vector of probabilities. By $\mathrm{SCC}(\boldsymbol{p}, k)$ we denote the event of successful transmission of an alert message when k of n stations tries to transmit using the vector of probabilities $\boldsymbol{p}$.

Lemma 2. *Let $\boldsymbol{p} = (p_i)_{i=1,\ldots,L}$ be a vector of probabilities, let $q_i = \max\{p_i, \frac{1}{n}\}$ and let $\boldsymbol{q} = (q_i)_{i=1,\ldots,L}$. Then*

$$(\forall k \in \{1, \ldots, n\})(\Pr[\mathrm{SCC}(\boldsymbol{p}, k)] \leq \Pr[\mathrm{SCC}(\boldsymbol{q}, k)]) \ .$$

Proof. Let us fix a number $k \geq 1$ and let $f_k(p) = kp(1-p)^{k-1}$. The function f_k is unimodal, reaches a maximum at point $p = \frac{1}{k}$. Hence if $k \leq n \leq \frac{1}{p}$ then $f_k(p) \leq f_k(\frac{1}{n})$. □

We shall prove the following theorem:

Theorem 3. *If $\boldsymbol{p} = (p_i)_{i=1,\ldots,L}$ is an arbitrary vector of probabilities then there exists $k \in \{1, \ldots, n\}$ such that*

$$\Pr[\mathrm{SCC}(\boldsymbol{p}, k)] \leq 1 - \left(1 - \frac{3e}{n^{\frac{1}{2(L+1)}}}\right)^L \ .$$

Proof. Let us fix n and let us consider a sequence $\boldsymbol{p}$ of length L such that

$$\min_{1 \leq k \leq n} \Pr[\mathrm{SCC}(\boldsymbol{p}, k)] = \sup_{\boldsymbol{x} \in [0,1]^L} \min_{1 \leq k \leq n} \Pr[\mathrm{SCC}(\boldsymbol{x}, k)] \ .$$

Using Lemma 2 we may assume that $p_i \geq \frac{1}{n}$ for all $i \in \{1, \ldots, L\}$. We may also assume that $p_1 \geq p_2 \geq \ldots \geq p_L$. We additionally put $p_0 = 1$ and $p_{L+1} = 1/n$.

Lemma 3. *There exists $i \in \{0, \ldots, L\}$ such that*

$$\frac{p_i}{p_{i+1}} \geq n^{\frac{1}{L+1}} \ .$$

Proof. Suppose that $p_0/p_1 < n^{1/(L+1)}, p_1/p_2 < n^{1/(L+1)}, \ldots, p_L/p_{L+1} < n^{1/(L+1)}$. Then

$$n = \frac{p_0}{p_1} \cdot \frac{p_1}{p_2} \cdots \frac{p_L}{p_{L+1}} < n^{\frac{L+1}{L+1}} = n \ ,$$

what is impossible. □

Let us fix a such that $\frac{p_a}{p_{a+1}} \geq n^{\frac{1}{L+1}}$. We shall consider three cases separately: $0 < a < L$, $a = 0$ and $a = L$. In the next considerations we shall use several times the inequality $x/e^x < 1.5/x^2$ which holds for all $x > 0$ and the inequality $(1-x)^{1/x} < e^{-1}$ which holds for all $x \in (0,1)$.

Case 1: $0 < a < L$. Let $p = p_a$ and $q = p_{a+1}$. We choose $k = 1/\sqrt{pq}$. Notice that $p/q \geq n^{1/(L+1)}$, $kp = \sqrt{p/q}$ and $k^2 \geq n^{1/(L+1)}$ (because: $k^2 = (pq)^{-1} = p^{-2}(p/q) \geq p^{-2} n^{1/(L+1)} \geq n^{1/(L+1)}$). Let $k^* = \lceil k \rceil$. Then for arbitrary $x \in (0,1)$ we have $k^* x(1-x)^{k^*-1} \leq 2kx(1-x)^{k-1}$.

Subcase 1. If $i \leq a$ and $p \leq 1 - \frac{1}{e}$ then we have

$$k^* p_i (1-p_i)^{k^*-1} \leq 2kp_i(1-p_i)^{k-1} \leq 2kp(1-p)^{k-1} \leq 2kp(1-p)^k e =$$

$$\sqrt{\frac{p}{q}}(1-p)^{\frac{1}{p}\sqrt{\frac{p}{q}}} 2e < \frac{\sqrt{\frac{p}{q}}}{\exp(\sqrt{\frac{p}{q}})} 2e < \frac{3e}{n^{1/(L+1)}}$$

Subcase 2. If $i \leq a$ and $p > 1 - \frac{1}{e}$ then we have

$$k^* p_i (1-p_i)^{k^*-1} \leq 2kp_i(1-p_i)^{k-1} \leq 2kp(1-p)^{k-1} <$$

$$2kp\left(\frac{1}{e}\right)^{k-1} \leq 2\frac{ke}{e^k} < \frac{3e}{k^2} \leq \frac{3e}{n^{1/(L+1)}}$$

Subcase 3. If $a < i \leq L$ and $q \leq 1 - \frac{1}{e}$ then

$$k^* p_i (1-p_i)^{k^*-1} \leq 2kp_i(1-p_i)^{k-1} \leq 2kq(1-q)^{k-1} \leq 2kq(1-q)^k e =$$

$$2\sqrt{\frac{q}{p}}(1-q)^{\frac{1}{q}\sqrt{\frac{q}{p}}} e < 2\frac{\sqrt{\frac{q}{p}}}{\exp(\sqrt{\frac{q}{p}})} e < 2\sqrt{\frac{q}{p}} e \leq \frac{2e}{n^{1/(2(L+1))}}$$

Subcase 4. If $a < i \leq L$ and $q > 1 - \frac{1}{e}$ then

$$k^* p_i (1-p_i)^{k^*-1} \leq 2kp_i(1-p_i)^{k-1} \leq 2kq(1-q)^{k-1} < 2kq\left(\frac{1}{e}\right)^{k-1} \leq$$

$$\frac{2ke}{\exp(k)} < \frac{3e}{k^2} \leq \frac{3e}{n^{1/(L+1)}}$$

Therefore we shown that in all subcases of Case 1 we have

$$\Pr[\mathrm{SCC}_{k^*}] = 1 - \prod_{i=1}^{L}(1 - k^* p_i(1-p_i)^{k^*-1}) < 1 - (1 - \frac{3e}{n^{1/(2(L+1))}})^L$$

Case 2: $a = 0$. In this case we take $k = 1$ and since $p_1 \leq 1/n^{1/(L+1)}$ we get

$$\Pr[\mathrm{SCC}_1] = 1 - \prod_{i=1}^{L}(1 - 1 \cdot p_i(1-p_i)^{1-1}) \leq$$

$$1 - (1 - \frac{1}{n^{1/(L+1)}})^L < 1 - (1 - \frac{2e}{n^{1/(2(L+1))}})^L \,.$$

Case 3: $a = L$. In this case we take $k = n$. Then $np_L \geq n^{1/(L+1)}$. If $p_L \leq 1 - \frac{1}{e}$ we have

$$np_L(1-p_L)^{n-1} \leq np_L(1-p_L)^n e \leq np_L(1-p_L)^{\frac{1}{p_L}np_L} e < \frac{np_L}{\exp(np_L)} e < \frac{2e}{n^{1/(L+1)}}$$

and if $p_L > 1 - \frac{1}{e}$ then

$$np_L(1-p_L)^{n-1} < np_L \left(\frac{1}{e}\right)^{n-1} \leq \frac{n}{\exp(n)} e < \frac{2e}{n^{1/(L+1)}} \ ,$$

therefore

$$\Pr[\mathrm{SCC}_n] = 1 - \prod_{i=1}^{L}(1 - np_i(1-p_i)^{n-1}) \leq 1 - \left(1 - \frac{2e}{n^{1/(L+1)}}\right)^L < 1 - \left(1 - \frac{3e}{n^{1/(2(L+1))}}\right)^L \ .$$

Hence we have finished the analysis of cases and we see in each case we are able to find $k \in \{1, \dots, n\}$ such that

$$\Pr[\mathrm{SCC}_k] < 1 - \left(1 - \frac{3e}{n^{1/(2(L+1))}}\right)^L \ .$$

□

Let $\mathcal{W}$ denote the main branch of the Lambert function. Let us consider an arbitrary vector $\boldsymbol{p} = (p_i)_{i=1,\dots,L}$ of probabilities of length L. Let $\mathrm{SCC}_{L,n,k}$ denotes the event of successful transmission when k sensors are activated and let $\Pr[\mathrm{SCC}_{L,n}] = \min\{\Pr[\mathrm{SCC}_{L,n,k}] : k = 1, \dots, n\}$.

Theorem 4. *If* $L \leq \frac{\log n}{2\log(3e)} - 1$, $f > 1$ *and* $\Pr[\mathrm{SCC}_{L,n}] > 1 - \frac{1}{f}$ *then*

$$L \geq \frac{\log n}{2\mathcal{W}\left(\frac{3e}{2}\frac{f}{f-1}\log n\right)} - 1 \ .$$

Proof. If $L \leq \frac{\log n}{2\log(3e)} - 1$ then $3e/n^{1/(2(L+1)} \leq 1$ so we may apply the classical Bernoulli inequality $((\forall x \leq 1)((1-x)^n \geq 1 - nx))$ to Theorem 3 and obtain the following inequality

$$\Pr[\mathrm{SCC}_{L,n}] < \frac{3eL}{n^{\frac{1}{2(L+1)}}} \ .$$

Hence from $\Pr[\mathrm{SCC}_{L,n}] > 1 - \frac{1}{f}$ we deduce that $3eLn^{\frac{-1}{2(L+1)}} > 1 - \frac{1}{f}$, so also $3e(L+1)n^{\frac{-1}{2(L+1)}} > 1 - \frac{1}{f}$. This inequality may be solved by the use of the Lambert function $\mathcal{W}$, giving us the required inequality. □

Let us recall that $\log x - \log\log x < \mathcal{W}(x) < \log x - \frac{1}{2}\log\log x$ for $x \geq e$ (see e.g. [12]). Using this bounds we get

$$\frac{\log n}{2\mathcal{W}\left(\frac{3e}{2}\frac{f}{f-1}\log n\right)} > \frac{\log n}{2\log\left(\frac{3e}{2}\frac{f}{f-1}\log n\right)} = \frac{1}{2}\frac{\log n}{\log\log n + \log\left(\frac{3e}{2}\frac{f}{f-1}\right)} .$$

If $f > 1$ is fixed and n tends to infinity then

$$\frac{\log n}{2\mathcal{W}\left(\frac{3}{2}e\frac{f}{f-1}\log n\right)} \sim \frac{\log n}{2\log\log n} .$$

Let us finally remark that if $f = n$ then the inequality

$$1 - \left(1 - \frac{3e}{n^{1/(2(L+1))}}\right)^{L+1} > 1 - \frac{1}{f}$$

can be solved precisely giving us a bound $L > 0.236594\log n - 1$.

5 Conclusions

In this paper we show that there exists an oblivious alarm protocol for sensor network which use $O(\log n)$ time slots and that each oblivious alarm protocol for sensor network requires $\Omega(\frac{\log n}{\log\log n})$ time slot. The algorithmic gap remains to be clarified.

References

1. Alfandari, L., Paschos, V.: Approximating minimum spanning tree of depth 2. International Transactions in Operational Research 6, 607–622 (1999)
2. Clementi, A.E.F., Ianni, M.D., Monti, A., Rossi, G., SilvestriL, R.: Divide and conquer is almost optimal for the bounded-hop accumulation in ad hoc wireless networks. In: The Proceedings of SIROCCO (2005)
3. Clementi, A.E.F., Penna, M.D., Silvestri, R.: On the power assignment problem in radio networks. In: Mobile Networks and Applications, MONET (2004)
4. Clementi, A.E.F., Ianni, M.D., Lauria, M., Monti, A., Rossi, G., Silvestri, R.: A distributed protocol for the bounded-hops converge-cast in ad-hoc networks. In: The Proceedings of the 5th International Conference on AD-HOC Networks & Wireless (2006)
5. Zawada, M.: Analysis of the bounded-hops converge-cast distributed protocol in ad-hoc networks. In: The Proceedings of the 3nd International Workshop on Algorithmic Aspect of Wireless Sensor Networks (2007)
6. Nakano, K., Olariu, S.: Randomized leader election protocols in radio networks with no collision detection. In: ISAAC, pp. 362–373 (2000)
7. Nakano, K., Olariu, S.: Uniform leader election protocols in radio networks. In: ICPP, pp. 240–250 (2001)
8. Nakano, K., Olariu, S.: A survey on leader election protocols for radio networks. In: ISPAN 2002: Proceedings of the 2002 International Symposium on Parallel Architectures, Algorithms and Networks, Washington, DC, USA, p. 71. IEEE Computer Society, Los Alamitos (2002)

9. Gasieniec, L., Pelc, A., Peleg, D.: The wakeup problem in synchronous broadcast systems. SIAM J. Discrete Math. 14, 207–222 (2001)
10. Jurdzinski, T., Stachowiak, G.: Probabilistic algorithms for the wakeup problem in single-hop radio networks. In: ISAAC, pp. 535–549 (2002)
11. Klonowski, M., Kutylowski, M., Zatopianski, J.: Energy efficient alert in single-hop networks of extremely weak devices. In: Algosensors (2009)
12. Hoorfar, A., Hassiani, M.: Inequalities on the lambert w function and hyperpower function. Journal of Inequalities in Pure and Applied Mathematisc 9(51) (2008)

A Monotonicity of Functions from Section 3

In this appendix we prove monotonicity of functions considered in Sect. 3.

Lemma 4. *Let $f_i(k) = k\frac{1}{2^i}(1-\frac{1}{2^i})^{k-1}$. Then, the functions $f_i(2^{i-1})$, $f_{i+1}(2^{i-1})$ are decreasing and $f_{i-1}(2^i)$ is increasing for $i \geq 2$.*

Proof. Let $g_\alpha(x) = (1-\frac{1}{\alpha x})^{x-1}$. Then $f_i(2^{i-1}) = \frac{1}{2}g_2(2^{i-1})$, $f_{i+1}(2^{i-1}) = \frac{1}{4}g_4(2^{i-1})$ and $f_{i-1}(2^i) = 2g_{\frac{1}{2}}(2^i)$. Notice that

$$\frac{d}{dx}g_\alpha(x) = \left(1-\frac{1}{\alpha x}\right)^{x-1} \cdot \left(\frac{1}{x} + \frac{\alpha-1}{1-\alpha x} + \log\left(1-\frac{1}{\alpha x}\right)\right) .$$

We consider $x \geq 4$ and $\alpha \geq \frac{1}{2}$. Then $1 - 1/(\alpha x) > 0$, so $\left(1-\frac{1}{\alpha x}\right)^{x-1} > 0$. We are interested in the sign of derivative of the function g_α, so we only need to check the sign of the remaining part of the derivative. Let $z = \frac{1}{\alpha x}$. Then $0 < z < \frac{1}{2}$ and

$$\frac{1}{x} + \frac{\alpha-1}{1-\alpha x} + \log\left(1-\frac{1}{\alpha x}\right) = \alpha z + (\alpha-1)\frac{z}{z-1} - \log\left(\frac{1}{1-z}\right) .$$

We expand the right side of this equation and obtain

$$\alpha z - (\alpha-1)\sum_{i=1}^{\infty} z^i - \sum_{i=1}^{\infty}\frac{z^i}{i} = \alpha z - \sum_{i=1}^{\infty}(\alpha - 1 + \frac{1}{i})z^i = -\sum_{i=2}^{\infty}(\alpha-1+\frac{1}{i})z^i .$$

The last formula implies that if $\alpha = 1/2$ then this series is greater than zero and for $\alpha \geq 1$ this series is less than zero. □

Revisiting Clustering Based Efficient Broadcast for Wireless Multihop Networks with Memory Limited Nodes

Rafael Funke and Hannes Frey

University of Paderborn, Germany
{rfunke,hannes.frey}@upb.de

Abstract. The easiest way to broadcast a message in a wireless multihop network is to let each network node retransmit the broadcast message once it has received it. That way however many unnecessary redundant transmissions take place. A large body of literature exists which copes with the problem to keep the number of retransmissions small. Solutions described so far either guarantee that all nodes in the network are reached but suffer from nodes with strict memory limitations, or can be applied on nodes with memory limitations but sacrifice delivery guarantees.

In this work we look at a well known solution to reduce redundant broadcast transmissions: clustering nodes to form a connected network backbone where a broadcast limited on the backbone nodes still supplies all nodes in the network. We describe a new algorithm which guarantees that all network nodes are supplied by the cluster structure while the construction of the structure obeys strict memory limitations per node. We evaluate our solution in a log normal shadowing network simulation. Performance metrics considered are broadcast delay, delivery ratio, and flooding rate. For comparison we also consider simple flooding which requires just one storage location to memorize the currently handled broadcast task.

1 Introduction

In wireless infrastructure-less networks like sensor networks, sensor actuator networks, multihop ad hoc networks, or autonomous robot networks, a message is typically sent from source to destination via intermediate forwarding nodes. This is on the one hand required due to limited communication range of each individual node. On the other hand it is also a reasonable way of communication to save communication bandwidth. Many short range communication hops often have a better spatial reuse compared to just one direct transmission.

Since nodes can not resort to a backbone infrastructure, all communication has to be organized by the network nodes themselves. Here, so called clustering can be a helpful tool to organize communication. With clustering we refer to a method which logically groups nodes into subsets. Each subset is maintained by one node, the cluster head. All other nodes, called the cluster members, belong

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 53–66, 2010.
© Springer-Verlag Berlin Heidelberg 2010

to one cluster, i.e., they are assigned to one cluster head. To assure connectivity among the cluster heads, nearby cluster heads have to be connected by so called gateway paths. The nodes forming such gateway paths are called gateway nodes.

One particular communication form which we look at in this work is broadcast. Starting from one message originator, all nodes in the network have to be reached. Simple flooding, where each node repeats the received message once, is of course a possible way to reach all nodes. That way, however, often many redundant transmissions take place. Different treatments of such redundant transmissions have been described in the literature. In this work we look at one specific of these solutions: broadcasting over a cluster structure. If the subnetwork consisting of cluster heads and gateways is connected, it is sufficient to flood the message over that subnetwork to reach all network nodes.

In this work we focus our attention on broadcasts limited to cluster structures while having available only a low, constant amount of memory. Clustering has been described in many ways. However, the memory requirement at the nodes for constructing a cluster structure has obtained little attention so far. On small embedded wireless sensor and actuator devices, for instance, a few KB of volatile memory on the microcontroller has to be shared between network protocols and applications. In particular when thinking of mass deployment of such nodes where the cost per device needs to be kept low, just adding enough memory to support any clustering approach that determines all neighbor nodes before doing a clustering decision is prohibitive.

In the following section we summarize well known techniques to improve broadcast over simple flooding. Our summary includes in particular clustering approaches found in the literature. We come to the conclusion that existing flooding improvements either have broadcast delivery guarantees but are not designed to work with strict memory limitations, or work well under memory limitations but sacrifice delivery guarantees. An interesting question arises on how much memory is typically required in common wireless network scenarios to perform standard clustering based on the knowledge of neighbor nodes. We give some answers in that respect in Section 3 where we present results of an empirical simulation study. In Section 4 we then describe how clustering with connectivity guarantees can be performed under any constraint memory budget. Moreover, we describe a slight improvement for the gateway search which further reduces redundant retransmissions. The algorithm, both with and without the gateway search improvement, is then evaluated in a log normal shadowing network simulation and compared against simple flooding. We draw our conclusions in Section 6.

2 Related Work

2.1 Clustering

Early work describing ways of distributed cluster organization of a wireless multihop network can be found in [3,4]. The algorithms consist of a cluster head election and a cluster connection phase. Already in cluster election a node has to learn first all its neighbor nodes. In the cluster connection phase nodes yet have to be informed about their neighbors' neighbors. Algorithms described in sequel publications [6,8] based on that clustering approach require the same amount of memory per node.

Another clustering approach described in [12,10] eliminates the need for cluster heads. Clusters are formed among nodes such that all nodes in one cluster form a complete graph. Nodes belonging to more than one cluster can be treated as gateway nodes to connect adjacent clusters and (though not described in those references) a broadcast could be performed by flooding all gateway nodes. The algorithms to construct the cluster structure however require each node to know its complete neighborhood. Moreover, nodes have to be informed as well about their neighbors' neighbors.

2.2 Connected Dominating Sets

The head and gateway based cluster structure as described satisfies that each node either belongs to the set of cluster heads and gateway nodes, or it is a cluster member which is connected to one cluster head. In general, a node subset where each node is either in the set or has at least one neighbor in that set is referred a dominating set. If the sub graph defined by the dominating nodes is connected, the same way as done with cluster head and gateway nodes, a message reduced broadcast can be realized by just flooding the dominating set nodes.

In other words, the described clustering is a special case of constructing a connected dominating set. Some algorithms supposed to construct a connected dominating set which does not distinguish between cluster heads and gateway nodes, achieve this by a clustering approach, though. They construct cluster heads and connect them with gateways. The whole cluster head and gateway then is the dominating set result. Approaches of that kind are described in [1,9,2]. While the approaches in [1,9] require at least one-hop neighbor information, the memory requirements for nodes in [2] is significantly reduced. A node just has to store the one and two hop cluster heads it hears about. In a general graph, the number of these clusters is not bounded, though.

A dominating set can be constructed directly without using the clustering concept of course. The approach found in [18,19,17] starts with an initial marking of nodes which are essential for connectivity within their neighborhood. This requires that nodes know their immediate neighborhood. The initial marking is then refined by removing some nodes whose neighborhood is covered by others. This requires neighbor sets of neighbors to be compared, i.e., two hop neighbor information is required. Another approach found in [5] is to grow a spanning forest of small dominating sets and then connect the forest with a minimum spanning tree. The approach requires nodes to know their one and two hop neighborhood.

2.3 Neighbor Elimination

A further approach to reduce broadcast redundancy is to eliminate redundant rebroadcasting nodes starting from the broadcast originator. The concept of multipoint relays [14,7] follows such approach. Using one and two hop information each node determines in advance a subset of the one-hop neighbors covering all its two hop neighbors. All other neighbors are eliminated. The whole construction requires that nodes once know their one and two hop neighborhood.

As described in [15], eliminating redundant broadcast transmissions can also be done on the fly while the broadcast is going on. The basic idea is that a node receiving a message postpones its own retransmission for a short duration and first observes retransmissions it hears about in its neighborhood. When the postponed retransmission has to start, the node first checks if retransmissions it heard about already covers its own neighborhood. If this is the case, the planned retransmission can be canceled. A similar approach can also be found in [16]. In all these broadcasting variants nodes are supposed to know their one hop and two hop neighbors.

A neighbor elimination scheme described in [13] requires nodes to store at most a number encoding the current broadcast task and an additional counter. The work describes a probabilistic approach where nodes rebroadcast only with a certain probability. Despite this contention based approaches are described as well, where nodes cancel rebroadcasting when enough transmission from neighbor nodes or a transmission from a close by neighbor was overheard. For the latter approach distance can be estimated by the received signal strength, for example. All approaches can not guarantee broadcast delivery at all nodes.

Another approach described in [11] is to reduce broadcast redundancy with a cluster structure but to decide on the cluster structure on the fly while the broadcast is going on. In that approach every node overhears all other neighbors' broadcast transmissions and based on the states announced in each overheard broadcast message it decides for itself on the fly its cluster head, cluster member or gateway role. In that approach as well a node has to store only a number encoding the current broadcast and a counter to memorize the number of gateway and cluster head nodes it heard about, but only if the number of overheard cluster heads exceeds the number of overheard gateway nodes by a certain amount. With such heuristic however guaranteed broadcast delivery in can not be assured.

2.4 Summary

Table 1 summarizes the discussed flooding improvements. Memory requirements refers to the amount of information to be stored on each node. This can be constant (i.e. a node stores at most a constant number of nodes and counters), the complete one hop neighborhood, the complete two hop neighborhood, or the complete one and two hop cluster heads. As discussed, we distinguish between clustering, dominating set and neighbor elimination. Finally, we depict if the mechanism has guaranteed broadcast delivery (abbreviated as GD). Here delivery guarantees, however, require an idealized MAC layer where broadcast transmissions are not lost.

Table 1. Summary of the Related Work

Method	Memory Requirements	Technique	GD
[3,4,6,8,12,10]	one and two hop neighbors	clustering	yes
[1,9]	one hop neighbors	dominating set	yes
[18,19,17,5]	one and two hop neighbors	dominating set	yes
[2]	one and two hop clusters	dominating set	yes
[14,7,15,16]	one and two hop neighbors	neighbor elimination	yes
[13,11]	constant	neighbor elimination	no

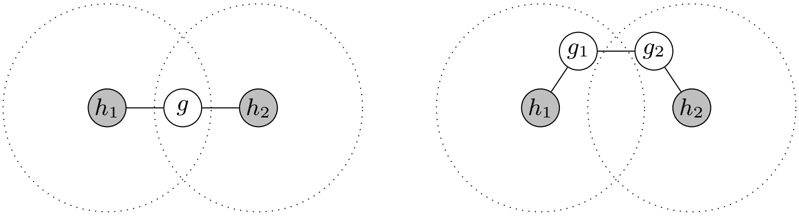
3 Problem Understanding

Obviously, approaches which require nodes to store one or two hop neighbors may fail on nodes with limited, constant memory. For any given memory size we just have to add enough neighbor nodes such that the memory exceeds. An interesting question however arises for approaches which just require nodes to store their neighboring cluster heads which are reachable in one or two intermediate hops.

In this case memory requirement can be kept low when we cluster that way such that each node is either a cluster head or connected to one, and no two cluster heads are immediate neighbors. In such cluster structure, which is also termed maximal independent set (MIS) clustering, it is sufficient to connect cluster heads with all other cluster heads which are reachable via one or two hop gateway paths (see Fig. 1 for an illustration).

Alzoubi et al. [2] found an upper bound for the number of neighboring cluster heads in unit disk graphs, where each node has a fixed transmission range and where an edge between two nodes exists if and only if the euclidean distance between them is below the transmission range. They proved that the number of cluster heads that can be reached by single and double gateways, is at most 47. Thus, it follows that in unit disk graphs any MIS-based clustering approach may save neighboring clusters without exceeding the constant memory available, as long as there is enough memory such that saving up to 47 addresses won't hurt.

However, what happens in more realistic graphs? In this work we consider that nodes are connected due to the log normal shadowing model where the transmitted signal is attenuated according to a path loss exponent and further a zero



(a) Single gateway connection. **(b)** Double gateway connection.

Fig. 1. Gateway constellations

mean log normal distributed random variable is added to the signal strength. Unlike in unit disk graphs, in log normal shadowing the results look fundamentally different, as there is no upper bound. Though it is very unlikely, in theory there can be placed infinitely many nodes around one node, where each node cannot communicate with any other of those nodes directly, but where single or double gateways exist, such that each node can communicate by one hop or two hops with the center node. Thus for log normal shadowing in theory the number of neighboring clusters is unbounded.

Of course it is very unlikely that these worst-case scenarios happen in the real world. To find out how many neighboring clusters we have to expect in real-world, we ran simulations with OMNeT++/MiXiM that built up a MIS cluster structure and counted the number of neighboring clusters. Our simulation scenario is that we place one node with nodeID 0 in the center. All other nodes are uniformly distributed in a circle around node 0 with radius 150m. The total number of nodes is varied as parameter between 100 nodes and 15000 nodes, which results in network densities between approx. 6 and 890. Further, the simulations were run twice, once for a unit disk graph channel model and once for a log normal shadowing model. For the unit disk model we chose the path loss exponent $\gamma = 3.2$. For the log normal shadowing model we chose the same value for the path loss exponent γ and we chose a zero-mean random variable with standard deviation $\sigma = 5.7$. For log normal shadowing, we assume block fading where the channel remains static for the whole simulation time of each run. On average, each node has a transmission range of 36.5m

The node 0 is initially selected to be clusterhead. All other nodes contend to be clusterhead by setting a random timer. Once the timer is expired, they will become clusterhead and notify surrounding nodes by sending a notification packet. Each node that receives the packet and is not yet clusterhead will become member and abort its timer. After all nodes have become cluster head or member, we count all cluster heads that can be reached by clusterhead 0 by single or double gateways.

Figure 2 shows our results. In our plots, network density represents, besides boundary effects, the average number of one-hop neighbors each node has. It is formally defined as $\#nodes\frac{avg\_node\_transmission\_range^2}{area\_radius^2}$. As we can see, the results for unit disk graph seems to converge to a mean value of 17 neighboring clusters. For log normal shadowing we see averages that are factor two to three higher than in unit disk graph, for network densities above 100, and the factor increases with higher network density. This is not unexpected as the number is bounded for unit disk graphs, but unbounded for log normal shadowing. Thus, we find that when looking at memory requirements for storing neighboring cluster heads, unit disk results may lead to results that are just a half or a third of the more realistic log normal shadowing model results. Further, the log normal simulation results show us that clustering approaches which store neighboring cluster heads will work in most cases, if each node has enough memory to store at least the amount of addresses which is given by the simulation results, but still there is no guarantee. Thus, for guaranteeing that an approach will work and for cases

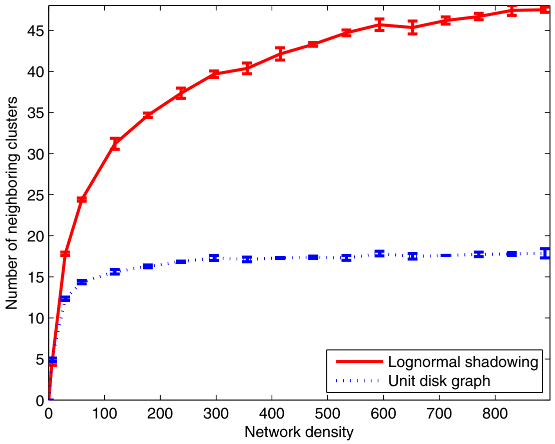


Fig. 2. Simulation results for average number of neighboring clusters

when memory is so scarce that even storing 30-50 addresses is too much, we have to look at new approaches.

4 Memory-Limited Clustering and Communication

In this section we present two novel memory-limited gateway search algorithms which both first use a well-known memory-limited MIS clustering algorithm to partition the network into cluster heads and members. Then they will determine gateway nodes that connect the unconnected cluster heads in a multihop fashion. A broadcast packet can then be delivered to all nodes by a cluster head- and gatewaynode-wide flooding.

4.1 The Basic Algorithm

For the first phase, the determination of cluster heads and members, we use a state-of-the-art MIS clustering algorithm like the one that can be found in [6]. Note, while the whole algorithm in [6] requires nodes to store one and two hop neighborhood information, the MIS construction part of that algorithm works with no memory requirements despite storing a single cluster head address. Note further, that our algorithm is not limited to exactly this approach, but can work with any other MIS clustering algorithm that works with limited memory.

In the second phase we select single and double gateways to connect cluster heads, like shown in Figure 1. There are existing approaches in the literature for determining single and double gateways between the cluster heads, but as stated in Section 2 there is no guarantee that they work with limited memory.

Next we describe an approach to find a single or double gateway between each pair of cluster heads, if at least one exists. If there is more than one possible gateway path between two cluster heads, then double gateways are ignored when there is at least one single gateway and the gateway path with maximum minimal per-link signal-to-noise ratio (SNR) will be chosen. Of course the memory requirements must be met for this phase and thus we neither want to save all possible gateway paths between each pair of heads, nor do we want to save all neighboring cluster heads.

In the proposed algorithm each node just needs to save one packet, containing at most two addresses and SNR values, and one further address and SNR value. The gateway search procedure is an election based approach where cluster heads may start gateway search once the cluster roles are determined. Nodes prioritize received gateway discovery message according to the node ID of the gateway search originator. If a node receives a gateway discovery with a higher originator's node ID, it will cancel the current gateway search and join the new one. Otherwise, it will stay put with the current ongoing gateway search. In either case the originator of the canceled gateway search has to be informed with a gateway discovery suppress message. The suppressed cluster head then has to backoff and retry its gateway search later on. To keep the description more readable we list the following pseudo code algorithm without considering the additional cases that an ongoing gateway search is canceled due to a newly received gateway search with higher priority.

Algorithm 1 gives a detailed description of our approach. Informally, the algorithm can be described as follows. In round i the head v_i^H sends out a discovery packet with a distance counter set to zero. All member nodes that receive the packet will modify the packet by writing the heads' address and the SNR of the received packet into it and by increasing the distance counter to one. Then all those nodes retransmit the modified packet using backoff. This packet will then be received by heads and members. Members will again write the address of the sending node and the received SNR into the packet, increase the distance counter to two and retransmit it after a backoff. Heads will save the first received packet together with the address of the sender and the minimum of received SNR and SNR values saved in the packet. If a head receives a packet which is not the first packet, it will check if the minimum of received SNR and SNR values saved in the packet is higher than its saved SNR value. If it is, the node will replace the saved packet and the saved SNR value. Further, a head will always ignore packets with distance counter set to two, if it received at least one packet with distance counter set to one. That means that potential double gateways between two heads will always be ignored if there is also a single gateway.

After the algorithm has been executed up to this point, each head in the neighborhood of v_i^H will have stored the best gateway path from v_i^H to itself. It then sends out the address of the single gateway or the addresses of the double gateway in a reply packet. Each node that finds its own address in the reply packet will set its gateway flag. In case the packet contains two gateway addresses (double gateway), it will increase the distance counter and resend the packet. Nodes receiving that packet and finding their address in such a packet will also set its gateway flag.

Algorithm 1. Gateway search algorithm

```
Phase 1: in round $i$ head $v_i^H$ starts discovery and sends discov(0,null,null)
Phase 2: upon receiving discov(0,null,null) from $v_i^H$ with SNR $\gamma_h$:
if I am member then
    set backoff timer and upon expiry send discov(1,[$v_i^H$,$\gamma_h$],null)
Phase 3: upon receiving discov(1,[$v_i^H$,$\gamma_h$],null) from node $v_1^M$ with SNR $\gamma_{m1}$:
if I am member then
    set backoff timer and upon expiry send discov(2,[$v_i^H$,$\gamma_h$],[$v_1^M$,$\gamma_{m1}$])
else if I am head and my address is not $v_i^H$ then
    if node has not stored a packet yet then
        Store packet, address $v_1^M$ and SNR $\min(\gamma_{m1}, \gamma_h)$
    else if stored SNR is lower than $\min(\gamma_{m1}, \gamma_h)$ then
        Replace stored packet, stored address and stored SNR by received packet, $v_1^M$
        and $\min(\gamma_{m1}, \gamma_h)$
Phase 4: upon receiving discov(2,[$v_i^H$,$\gamma_h$],[$v_1^M$,$\gamma_{m1}$]) from node $v_2^M$ with SNR $\gamma_{m2}$:
if I am head and my address is not $v_i^H$ and I have not stored a packet that matches
discov(1, · , ·) then
    if node has not stored a packet yet then
        Store packet, address $v_2^M$ and SNR $\min(\gamma_{m2}, \gamma_{m1}, \gamma_h)$
        Set discoveryOver timer to a large enough value
    else if stored SNR is lower than $\min(\gamma_{m2}, \gamma_{m1}, \gamma_h)$ then
        Replace stored packet, stored address and stored SNR by received packet, $v_2^M$
        and $\min(\gamma_{m2}, \gamma_{m1}, \gamma_h)$
Phase 5: upon expiry of discoveryOver timer:
if Stored packet describes two-hop path, i.e. is in format discov(1,[$v_i^H$,$\gamma_h$],null)
then
    send reply(0,$v_i^H$,$v_1^M$,null)
else if Stored packet describes three-hop path, i.e. is in format
discov(2,[$v_i^H$,$\gamma_h$],[$v_1^M$,$\gamma_{m1}$]) then
    send reply(0,$v_i^H$,$v_1^M$,$v_2^M$)
Phase 6: upon receiving reply(0,$v_i^H$,$v_1^M$,$v_2^M$):
if $v_2^M$==null then
    if $v_1^M$ is equal to own address then
        set gateway flag
else if $v_2^M$ is equal to own address then
    set gateway flag
    set backoff timer and upon expiry send reply(1,$v_i^H$,$v_1^M$,null)
Phase 7: upon receiving reply(1,$v_i^H$,$v_1^M$,null):
if $v_1^M$ is equal to own address then
    set gateway flag
Termination: delete saved packet, but keep gateway flag
```

After the algorithm has terminated, all member nodes that are on the best path from the originating cluster head to any cluster head that can be reached by at most two member hops, have the gateway flag set and thus cluster heads and gateway members form a connected dominating set. If any node now starts a cluster head- and gateway-wide flooding, all nodes can be reached.

4.2 Removing Redundant Gateways

A drawback of this approach is that it may produce redundant gateway nodes. For example, if three cluster heads are all in the neighborhood of each other, our algorithm will find gateways between each two of them. Thus the gateway paths form a circle and if a broadcast packet is sent, at least one cluster head will receive it twice. Unfortunately, we don't have a solution for this problem in general, but a further optimization which reduces the number of circles.

Our optimization is an algorithm based on Algorithm 1 that reduces the number of double gateways and thus reduces the number of redundant retransmissions. The idea of this algorithm is that we want to eliminate double gateways in case we know that there is a cluster head that both cluster heads have a single-gateway connection to. So, if a node shall get the outer gateway, i.e. the member farther away from v_i^H, for a double gateway connection, but is also connected to a cluster head that is connected by a single gateway to the originating cluster head, the following algorithm will drop that double gateway connection. The algorithm consists of two iterations, the first iteration is described by Algorithm 2 and the second iteration is described by Algorithm 3. Due to compactness, not the whole algorithms are shown, but only the phases differing to Algorithm 1.

Informally describing, in the first iteration the Algorithm 1 is repeated, except that no gateway flag is set and that each node receiving a reply packet addressing a single gateway from a cluster head sets a double gateway blocking flag. This flag will prevent the node from becoming an outer double gateway for v_i^H. Further, nodes will not retransmit reply packets. On termination, the algorithm will delete saved packets and keep the double gateway blocking flag. Then the second iteration algorithm starts, which is also equal to Algorithm 1, except for the following differences. Each node having the double gateway blocking flag set will not repeat discovery packets with distance set to one. Upon termination the algorithm will delete saved packets and the double gateway blocking flag, but not the gateway flag.

Algorithm 2. Improved gateway search algorithm with double gateway blocking; first iteration algorithm (unmentioned phases see Algorithm 1)

```
Phase 6: upon receiving reply($0, v_i^H, v_1^M, v_2^M$):
1. if $v_2^M$==null then
2.     set double gateway blocking flag
Termination: delete saved packet, but keep gateway flag and double gateway blocking flag
```

Algorithm 3. Improved gateway search algorithm with double gateway blocking; second iteration algorithm (unmentioned phases see Algorithm 1)

```
Phase 3: upon receiving discov(1,[$v_i^H$,$\gamma_h$],null) from node $v_1^M$ with SNR $\gamma_{m1}$:
if I am member then
    if double gateway blocking flag is not set then
        set backoff timer and upon expiry send discov(2,[$v_i^H$,$\gamma_h$],[$v_1^M$,$\gamma_{m1}$])
else if I am head and my address is not $v_i^H$ then
    see Phase 3 in Algorithm 1 (line 1 to 1)
Termination: delete saved packet and double gateway blocking flag, but keep gateway flag
```

4.3 Algorithm Runtime and Memory Requirements

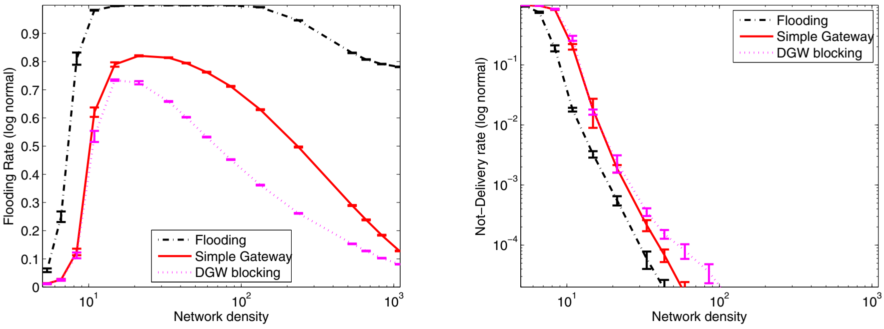
Our proposed algorithms consist of a constant number of phases, where each phase has a constant execution time. Thus each algorithm itself has a runtime of $O(1)$. As each node in the network needs to initiate the algorithm once, the total runtime is $O(n)$ for the whole network, while n is the number of nodes.

During one iteration of one of the algorithms any node needs to store at most one packet, one address and an SNR value. Further one gateway flag and one double gateway blocking flag need to be stored longer than one execution. As concurrent executions of the algorithm are suppressed and as each execution requires constant memory, the total required memory is constant.

5 Simulation Results

We have run simulations to evaluate our two proposed algorithms and compare them to simple flooding. For the simulations we chose a scenario where one node with nodeID 0 is placed in the center and 1000 nodes are distributed uniformly in a circle around this node. The radius of the circle is varied as parameter and ranges from 35m to 500m, which results in network densities from 1000 to 5. The channel parameters for log normal shadowing were chosen as in Section 3. The MAC layer we used is a simple CSMA-MAC, which is part of MiXiM. As MAC and PHY parameters we chose a bitrate of 256000bit/s, a contention window of 20 with at most 14 transmission attempts and 1mW transmission power. The simulation first executes the gateway search algorithm and then starts a broadcast at node 0. While for simple flooding all nodes are supposed to forward packets, for our two algorithms, the simple and the double gateway blocking algorithm, only the cluster heads and the nodes having the gateway flag set are supposed to forward packets. Those nodes will do a backoff and retransmit the broadcast packet at most once upon receiving a broadcast packet for the first time.

Figure 3 shows the flooding rate of all approaches along with the delivery rate. As we can see in Figure 3a for network densities of about 20 our simple gateway algorithm already performs at 80% flooding rate and for the network density



(a) Flooding rate

(b) Delivery rate

Fig. 3. Flooding rate and not-delivery rate for broadcast transmission using our clustering algorithm with simple and double gateway blocking (DGW blocking) compared to simple flooding

going up to 1000 our algorithm can drop the flooding rate down to 10%. Further, our double gateway blocking algorithm even performs better, as it can decrease the flooding rate by 10% more to about 70% for network density approximately 20. For increasing network densities the proportional difference increases up to the network density of about 100 where the double gateway blocking algorithm can approximately halve the flooding rate of our simple gateway algorithm. As we can see in Figure 3b, the not-delivery rate for all broadcast approaches drops below 1% for network densities of 20 and above and thus reaches a high reliability. Of course, though the schemes have 100% delivery guarantees in theory, in our simulation delivery below 100% is due to packet losses caused by MAC layer contention and due to the fact that sparse networks might just be disconnected.

Figure 4 shows the delay, which is the time from the first transmission until the last retransmission of the broadcast packet. As can be seen, for network densities above 100 our algorithms performs better than simple flooding. For network densities of about 1000 our double gateway blocking gateway selection nearly halves the delay compared to simple flooding. Here, not only the network density is responsible for the delay gain, but also the network diameter. As we do not vary the number of nodes as parameter, but the area in which the nodes are distributed, we get small network diameters for high-density networks and thus the number of hops that are at least necessary to deliver a broadcast to all nodes is lower than in low-density networks. Further, it is surprising that the delay is not dramatically increasing for high-density networks, as one might expect. This is due to the backoff behavior of our CSMA MAC. If many nodes reside in one collision domain, i.e. interfere with each other, and want to retransmit a packet at the same time, then due to backoff and contention nodes may fail to send multiple times due to busy channel and after 14 unsuccessful attempts they will give up. Due to that the delay in one collision domain is bounded to 14 backoff time slots and thus in our scenario flooding is not as bad as one might think in

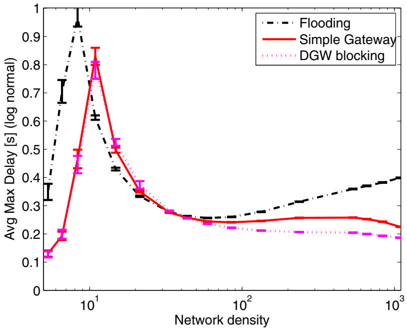


Fig. 4. Broadcast delay using our clustering algorithm with simple and double gateway blocking (DGW blocking) compared to simple flooding

terms of delay. Note that this is also the reason why the flooding rate for simple flooding in Figure 3a reaches 100%, but decays to < 80% again.

6 Conclusion

To the best of our knowledge, we presented for the first time a clustering approach that works with a low, constant amount of memory, using a state-of-the-art MIS clustering algorithm and a novel gateway selection algorithm. The purpose of the gateway selection algorithm is to determine nodes that are needed to connect the cluster heads. We then compared the broadcast behavior of simple network-wide flooding to the flooding of cluster heads and gateway nodes, determined by our algorithm. We showed that using our algorithm, we can reduce the flooding rate to about 10% for network densities ranging up to 1000. Further, we introduced an optimization to our algorithm, which we call double gateway blocking and which reduces the number of gateway nodes. We showed that using this optimization the flooding rate can even get up to 50% better compared to our original algorithm, while both algorithms require low constant amount of memory.

References

1. Alzoubi, K.M., Wan, P.J., Frieder, O.: New distributed algorithm for connected dominating set in wireless ad hoc networks. In: Proc. 35th Annual Hawaii International Conference on HICSS System Sciences, pp. 3849–3855 (2002)
2. Alzoubi, K.M., Wan, P.J., Frieder, O.: Message-optimal connected dominating sets in mobile ad hoc networks. In: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc), Lausanne, Switzerland, pp. 157–164 (2002)
3. Baker, D., Ephremides, A.: The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications 29(11), 1694–1701 (1981)

4. Baker, D., Ephremides, A., Flynn, J.: The design and simulation of a mobile radio network with distributed control. IEEE Journal on Selected Areas in Communications 2(1), 226–237 (1984)
5. Das, B., Bharghavan, V.: Routing in ad-hoc networks using minimum connected dominating sets. In: Proceedings of the IEEE International Conference on Communications (ICC) (1997)
6. Gerla, M., Tsai, J.T.C.: Multicluster, mobile, multimedia radio network. Journal of Wireless Networks 1, 255–265 (1995)
7. Jacquet, P., Minet, P., Mühlethaler, P., Rivierre, N.: Increasing reliability in cable free radio lans: Low level forwarding in hiperlan. Wireless Personal Communications 4(1), 65–80 (1997)
8. Jiang, M., Li, J., Tay, Y.C.: Cluster based routing protocol (CBRP) functional specification. Internet Draft (1998)
9. Kozat, U.C., Kondylis, G., Ryu, B., Marina, M.K.: Virtual dynamic backbone for mobile ad hoc networks. In: Proceedings of the IEEE International Conference on Communications (ICC), vol. 1, pp. 250–255 (2001)
10. Krishna, P., Chatterjee, M., Vaida, N., Pradhan, D.: A cluster-based approach for routing in ad-hoc networks. In: Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing, pp. 86–95 (1995)
11. Kwon, T.J., Gerla, M., Varma, V.K., Barton, M., Hsing, T.R.: Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks. Proceedings of the IEEE 91(8), 1210–1220 (2003)
12. Lin, C.R., Gerla, M.: A distributed control scheme in multi-hop packet radio networks for voice/data traffic support. In: Proceedings of IEEE International Conference on Communications (ICC), pp. 1238–1242 (1995)
13. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 151–162 (1999)
14. Qayyum, A., Laouiti, A., Viennot, L.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: Proceedings of the Hawaii International Conference on System Sciences, HICSS-35 (2002)
15. Stojmenovic, I., Seddigh, M., Zunic, J.: Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. IEEE Transactions on Parallel and Distributed Systems 13(1), 14–25 (2002)
16. Wei, P., Xi-Cheng, L.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: Proceedings of the First Annual Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc), Boston, USA, pp. 129–130 (2000)
17. Wu, J., Dai, F., Gao, M., Stojmenovic, I.: On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. Journal of Communications and Networks 4(1) (2002)
18. Wu, J., Li, H.: Domination and its applications in ad hoc wireless networks with unidirectional links. In: Proc. International Conference on Parallel Processing, pp. 189–197 (2000)
19. Wu, J., Li, H.: A dominating-set-based routing scheme in ad hoc wireless networks. Telecommunication Systems Journal 3, 63–84 (2001)

Locally Proactive Routing Protocols

Kahkashan Shaukat and Violet R. Syrotiuk

School of Computing, Informatics & Decision Systems Engineering
Arizona State University, P.O. Box 878809, Tempe, AZ 85287-8809
{kshaukat,syrotiuk}@asu.edu

Abstract. In a proactive routing protocol, each node periodically transmits control packets to distribute its local topology. Since the conditions of a mobile ad hoc network are rarely uniform, using the same period at each node may not result in the best performance. Therefore, we propose a protocol that is *locally proactive*, i.e., each node periodically transmits control packets using a time interval based on its local network conditions. Our objective is to simultaneously maximize the packet delivery ratio and minimize the control overhead. To accomplish it, we develop statistical models for each response for the OLSR routing protocol as a function of node speed and two timer intervals. In simulation we show that the locally proactive OLSR reduces the control overhead by 35-44% and obtains a packet delivery ratio that is statistically identical to that of the original OLSR protocol.

1 Introduction

A *mobile ad hoc network* (MANET) is a self-organizing collection of mobile wireless nodes without any fixed infrastructure. MANETs are appropriate for applications such as rescue or emergency operations after a disaster, tactical missions in a hostile or unknown territory, and extending the range and capacity of infrastructure based networks such as the Internet.

Routing is a fundamental operation in MANETs; consequently, the problem is well studied. Two major categories of routing protocols have emerged: reactive and proactive. A reactive protocol, such as AODV [11] or DSR [4], computes a route to a destination only when necessary. As a result, the source may experience a delay in path set-up before data transmission starts. A proactive protocol, such as OLSR [3] or TBRPF [9], distributes topology information periodically. Therefore routes to all destinations are always available, but at the price of the control overhead to keep routes fresh.

It is rarely the case that the conditions are uniform throughout a MANET. For example, a node's speed, the density of its neighbourhood, and the number of flows routed through it, are likely different for each node. For a proactive protocol, this suggests that in parts of the network where the conditions are changing rapidly control information should be distributed more frequently. Conversely, in parts of the network where the conditions are changing slowly a less frequent distribution may suffice. Based on these observations, we propose to transmit

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 67–80, 2010.
© Springer-Verlag Berlin Heidelberg 2010

control packets at each node using a periodic time interval based on its local network conditions – the resulting protocol remains proactive, but it is *locally proactive* in the sense that the periodic interval used to distribute topology information is determined locally at each node rather than a single static interval used globally.

Clearly, there is a trade-off between reducing control overhead and the success of routing data packets. If the interval at which topology information is distributed is too long, the routes become out-of-date and packets will be dropped. On the other hand, if the interval is too short, the control overhead is also high, consuming bandwidth that can otherwise be used to transmit data. Hence our objective is to both maximize the packet delivery ratio and minimize the control overhead.

In this paper, we take the proactive OLSR protocol and evaluate making it locally proactive. In OLSR, each node uses two types of control messages: HELLO messages to distribute two-hop neighbourhood information, and TC messages to distribute topology control information. In OLSR, HELLO and TC messages are transmitted periodically every $2\,s$ and $5\,s$, respectively. Using designed experiments [7], statistical models are developed for *packet delivery ratio* (PDR) and *control overhead* (CO) as a function of node speed, and the HELLO and TC intervals. A hybrid optimization technique that leverages information about the known gradients to constrain the perturbations of an unknown gradient [5] is used to find solutions for the models. `ns-2` simulations are performed to compare the performance of the locally proactive OLSR protocol to the original OLSR protocol. Simulations show that the locally proactive OLSR reduces the control overhead by 35-44% while retaining a packet delivery ratio that is statistically identical to the original OLSR protocol.

The remainder of this paper is organized as follows. §2 summarizes some techniques from designed experiments used in MANETs as well as reviews previous work on reducing control overhead in OLSR. §3 describes the methodology followed in this paper, while §4 presents the simulation results. Finally, we give conclusions and propose future work in §5.

2 Related Work

2.1 Designed Experiments in MANETs

The assessment of the behaviour of MANETs is non-trivial, since they are a collection of nodes with no fixed infrastructure or centralized control. Vadde et al. [16] use statistical *design of experiments* (DOE) to analyze the impact of factors and their interactions on MANET service delivery. They found that the MAC protocol and its two-way interaction with the routing protocol had the most significant effect on average delay. Optimizing this interaction decreased the average delay.

Barrett et al. [1] use *analysis of variance* (ANOVA) techniques to study the empirical effect of the interaction between the routing protocol and the MAC protocol for different mobility models in wireless radio networks. Their analysis suggests that different combinations of routing and MAC protocols result in varying performance.

Totaro et al. [15] demonstrate the main and two-way interaction effects of network density, node mobility, traffic load, and the MAC protocol on the packet delivery ratio and end-to-end delay using statistical DOE. They developed first-order linear regression models to predict these responses in MANETs.

Vadde et al. [17] apply DOE to identify significant timers in the AODV routing protocol and the number of retries before declaring link failure in the MAC protocol affecting MANET performance. They use *Response Surface Methodology* (RSM) to optimize the values of the significant factors to improve the average packet delay and the average throughput.

2.2 Reducing Control Overhead in OLSR

OLSR is a link state routing protocol for MANETs. Broadcasts of link state eventually provide each node with a complete view of the network state [3]. In an effort to reduce control overhead compared to classical link state protocols, OLSR uses *multi-point relay* (MPR) sets and *multi-point relay selector* (MS) sets. An MPR set for a node is a subset of its one-hop neighbours that reaches all its two-hop neighbours. MPRs are selected through the exchange of HELLO messages that also provide each node a view of its two-hop neighbourhood. MPRs are also the nodes that forward *topology control* (TC) messages. TC messages provide each node in the network with sufficient topology state information to compute the routing table. An MS set for an MPR consists of the nodes that have selected it as an MPR.

Several ideas to reduce control overhead in OLSR have been proposed. Ros and Ruiz [12] develop Clustered OLSR (C-OLSR). Their goal is to reduce control overhead by partitioning the network into clusters. They restrict the propagation of TC messages to inside every cluster. The generation and forwarding of inter-cluster topology information is done by the MPRs at the cluster level. C-OLSR outperforms the original OLSR protocol in terms of overhead generation and achievable throughput.

Xue et al. [18] present an optimization approach to reduce overhead by changing the format of the messages and how they are sent. They change the HELLO messages to advertise only those links that have changed during the HELLO interval. Rather than all nodes generating TC messages, only the MPRs generate them to further reduce control overhead. The authors also reduce redundancy on the transmission of TC messages; for example, they allow only one node to send a TC message in scenarios where a group of nodes have chosen each other as MPRs. These modifications decrease the routing overhead by about 17% in high node density scenarios; the decrease is about 4% at high mobility. The modified protocol is capable of maintaining the end-to-end delay.

In Hierarchical OLSR (HOLSR) [6], nodes exchange TC messages with a frequency that depends on the scope, defined as distance in hops between a pair of nodes. The authors use a “fisheye” technique [10] and capture topology information with high detail depending upon the hop count. The detail decreases as the distance increases. That is, TC messages from nodes that are further away are received at larger intervals. Routing tables are updated appropriately to incorporate the change. They measure the performance of the protocol based on the success rate i.e the transmission ability of the protocol and the end-to-end delay. HOLSR obtains a 87.32% success rate whereas OLSR only obtains 81%. The end-to-end delay is reduced to almost half.

Shaukat and Syrotiuk [13] develop Adaptive OLSR (A-OLSR) to monitor the value of the betweenness centrality measure in the two-hop neighbourhood of a node in order to decide when to transmit TC messages. Each node calculates its betweenness and uses Shewart control charts to monitor the value. If the value is in-control, no action is taken; when out-of-control, a TC message is sent to reflect a change in the topology. They show a reduction of 26 to 46% in control overhead with little impact on throughput or delay.

The work in this paper seeks to reduce the control overhead in OLSR while maximizing the packet delivery ratio.

3 From a Proactive to a Locally Proactive Protocol

In order to propose a locally proactive version of OLSR we need to determine how to set the HELLO and TC intervals depending on local network conditions.

3.1 Screening Experiments

The goal of the screening experiments is to understand how the node speed, HELLO interval, and TC interval affect the responses of packet delivery ratio (PDR) and control overhead (CO) in the OLSR protocol. The PDR is defined as the ratio of number of packets successfully delivered to a destination to the number of packets transmitted. The CO is the number of control packets transmitted for successful routing from source to destination. Speed is varied from biking speed ($5\,m/s \approx 11\,mi/hr$) to motor-vehicle speed ($20\,m/s \approx 45\,mi/hr$). The HELLO and TC interval are assigned levels such that large intervals do not make the network inactive, and small intervals do not cause congestion. The levels of the chosen factors are given in Table 1.

A general factorial design is used where each factor has a fixed number of levels and experiments are run with all possible combinations. A design consisting k factors with levels $l_1, l_2 \ldots l_k$ respectively is a $l_1 \times l_2 \times \ldots \times l_k$ factorial design. Thus, we have a $4 \times 12 \times 6$ factorial design with 288 design points; each

Table 1. Factors and their different levels

Factor	Factor Name	Units	Levels	Values
A	Node Speed	m/s	4	5, 10, 15, 20
B	HELLO interval	seconds	12	0.5-6 at increments of 0.5 s
C	TC interval	seconds	6	0.5, 1, 2.5, 5, 7.5, 10

Table 2. Simulation parameters

Parameter	Value
Simulator	`ns-2`, version 2.34
Simulation duration	500 s
Confidence interval	95%
Simulation area size	$500 \times 500\,m^2$
Number of nodes	20
Transmission range	250 m
Channel bandwidth	2 $Mbps$
Traffic type	Constant bit rate (CBR)
Packet arrival rate	10 $pkts/s$
Packet size	512 $bytes$
Mobility model	Stationary random waypoint
Node speed	$5, 10, 15,$ and $20\,m/s$
MAC protocol	IEEE 802.11b

design point is replicated ten times. The simulations are carried out in `ns-2` [14] with the duration of each simulation being 500 s. The simulations consist of 20 nodes uniformly distributed in a $500 \times 500\,m^2$ area. Each node is equipped with an omni-directional antenna with a transmission range of 250 m. The physical channel uses the free-space model with a 2 $Mbps$ channel capacity. The steady state random waypoint model [8] is used to generate the movement patterns; the pause-time is set to zero for continuous mobility. There is one source destination pair. The source transmits 512 $byte$ UDP packets at a rate of 10 $pkts/s$. Table 2 shows these and other simulation parameters.

3.2 Empirical Models

The statistical DOE approach facilitates building empirical models from data collected from experimentation. An evaluation of the results obtained from the screening experiments shows that a quadratic model fits the experimental data. The final models (excluding non-significant terms) for PDR and CO are presented in Equation (1). The ANOVA analysis is given in Table 3; only significant factors and interactions are listed. The ANOVA analysis for PDR reveals that factors A, B, C, A^2 and B^2 are significant, whereas, the two way interactions AB, AC, BC, and C^2 are not significant. It also shows that A (node speed) has a very large (57.24%) contribution to PDR; the other factors contribute less than five percent. The TC interval and HELLO interval have 39.89% and 16.62% contribution to the CO respectively.

$$\begin{aligned} PDR &= 97.39 - 1.73A + 2.55B - 0.11C + 0.03A^2 - 0.45B^2 \\ CO &= 21135.54 + 587.56A - 4876.69B - 3184.34C - 19.94AB \\ &\quad -34.73AC + 36.53BC - 8.49A^2 + 541.85B^2 + 231.39C^2 \end{aligned} \tag{1}$$

Table 3. ANOVA analysis for packet delivery ratio and control overhead (significant terms only)

(a) Packet delivery ratio

Source	Sum of Squares	df	Mean Square	F value	$p - value$ $Prob > F$	Percentage Contribution
A-Speed	1.012E+005	1	1.012E+005	5276.35	< 0.0001	57.24
$B - Hello\_Interval$	4683.85	1	4683.85	244.15	< 0.0001	2.65
$C - TC\_Interval$	5885.31	1	5885.31	306.78	< 0.0001	3.33
A^2	1533.62	1	1533.62	79.94	< 0.0001	0.87
B^2	3962.18	1	3962.18	206.53	< 0.0001	2.24
R^2	0.6887					
Adjusted R^2	0.6877					

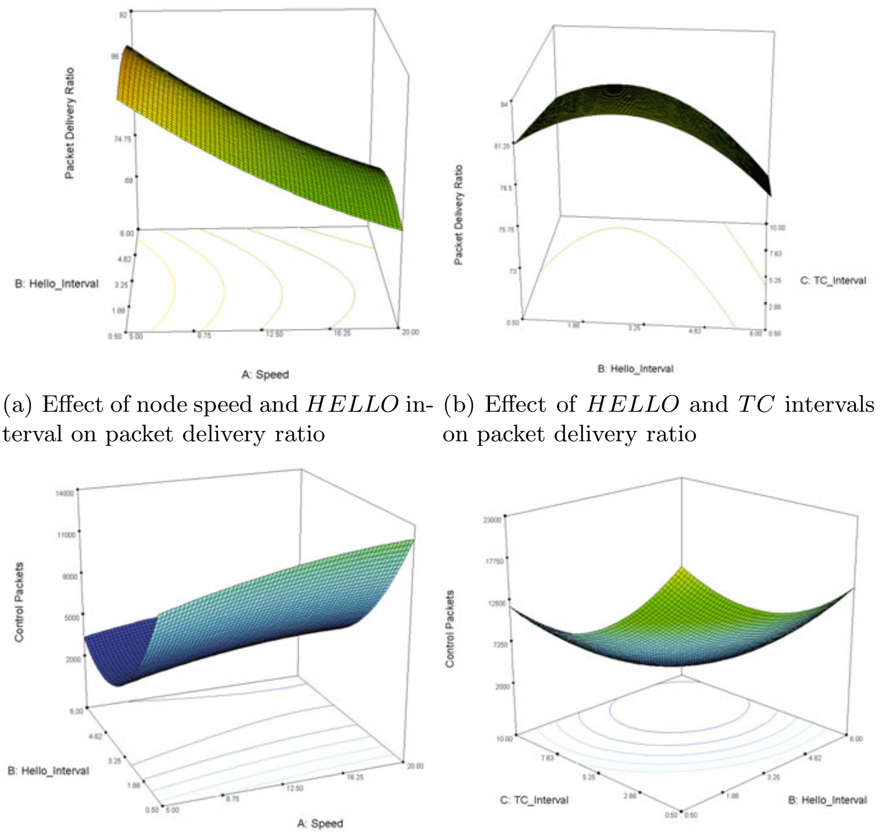
(b) Control overhead

Source	Sum of Squares	df	Mean Square	F value	$p - value$ $Prob > F$	Percentage Contribution
A-Speed	$1.398E + 009$	1	$1.398E + 009$	314.86	< 0.0001	1.44
$B - Hello\_Interval$	$1.617E + 010$	1	$1.617E + 010$	3640.43	< 0.0001	16.62
$C - TC\_Interval$	$3.880E + 010$	1	$3.880E + 010$	8734.91	< 0.0001	39.89
AB	$1.066E + 008$	1	$1.066E + 008$	24.00	< 0.0001	0.11
AC	$1.297E + 009$	1	$1.297E + 009$	292.04	< 0.0001	1.33
BC	$1.369E + 008$	1	$1.369E + 008$	30.81	< 0.0001	0.14
A^2	$1.297E + 008$	1	$1.297E + 008$	29.21	< 0.0001	0.13
B^2	$5.878E + 009$	1	$5.878E + 009$	1323.42	< 0.0001	6.04
C^2	$1.183E + 010$	1	$1.183E + 010$	2663.72	< 0.0001	12.16
R^2	0.8689					
Adjusted R^2	0.8685					

3.3 Multi-objective Optimization

The response surface and contours graphs for PDR and CO are presented in Figure 1. The response surfaces for PDR with respect to node speed and HELLO interval in Figure 1(a) shows that the stationary point, i.e., the point at which the PDR is highest, is a saddle-point. The response surface for PDR with respect to HELLO interval and TC interval in Figure 1(b) shows a maximum as the stationary point. The response surfaces for CO with respect to node speed and HELLO interval and HELLO interval and TC interval as shown in Figures 1(c) and (d) respectively, both show the stationary points as minima. Optimization of a response is generally obtained using a gradient approach such as steepest ascent (descent). In this case, the surfaces can be approximated as second order polynomials. While steepest ascent (descent) finds a solution, it may take a long time to do so. We use $SP(SA)^2$ [5], a hybrid technique that allows simultaneous perturbation of responses for which gradients are unavailable within a search region. It works well when statistical models exist for some responses but are unavailable for others. Here, the search region almost certainly contains the gradient path of steepest ascent (descent) for modelled responses. In this case, the modelled responses are PDR and CO, while the unmodelled response is packet loss ratio. This method finds the solutions in fewer steps as compared to the steepest ascent (descent) method.

For each speed, the PDR and CO models are used to find the optimized values of the HELLO and TC intervals by using $SP(SA)^2$. Table 4 tabulates these optimized values for different node speeds. Each node in a locally proactive OLSR protocol measures its node speed rounded to the nearest integer value, and uses it to index this table to obtain the HELLO and TC intervals. The HELLO



(a) Effect of node speed and $HELLO$ interval on packet delivery ratio

(b) Effect of $HELLO$ and TC intervals on packet delivery ratio

(c) Effect of node speed and $HELLO$ interval on control overhead

(d) Effect of $HELLO$ and TC intervals on control overhead

Fig. 1. Response surfaces and contours for packet delivery ratio and control packets

Table 4. Optimized HELLO and TC intervals for a given node speed

Node Speed	HELLO Interval	TC Interval
$2\,m/s$	$3.19\,s$	$5.29\,s$
$5\,m/s$	$3.42\,s$	$5.67\,s$
$10\,m/s$	$3.69\,s$	$6.55\,s$
$15\,m/s$	$4.62\,s$	$6.59\,s$
$20\,m/s$	$3.56\,s$	$6.55\,s$

and TC timers are set to expire at the end of the respective time intervals. When a timer expires, the node checks its speed once again and chooses the appropriate interval from Table 4.

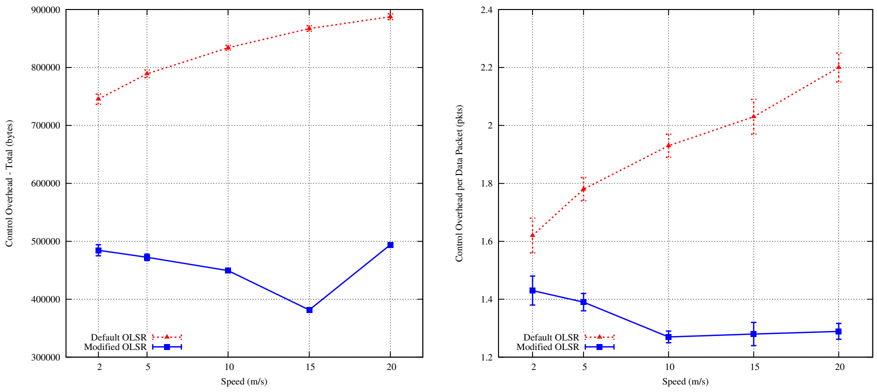
4 Evaluation of Locally Proactive OLSR

Simulations using the `ns-2` network simulator [14] with extensions for wireless mobility [2] are performed to observe the locally proactive OLSR protocol compared to the original OLSR. The simulation parameters used are the same as shown in Table 2. In addition to the speeds considered in the original design, we added the node speed $2\,m/s$ to see how the protocol performs outside the design space. A higher packet arrival rate is added ($100\,pkts/s$) and a second data flow is introduced to find out how the protocols behave under higher load. Control overhead, control overhead per data packet, packet delivery ratio, and average end-to-end delay are measured. End-to-end delay is the time taken for a packet to be transmitted from the source of the flow to the destination of the flow. The average of end-to-end delay for all the data packets successfully received is used here.

4.1 Simulation Results

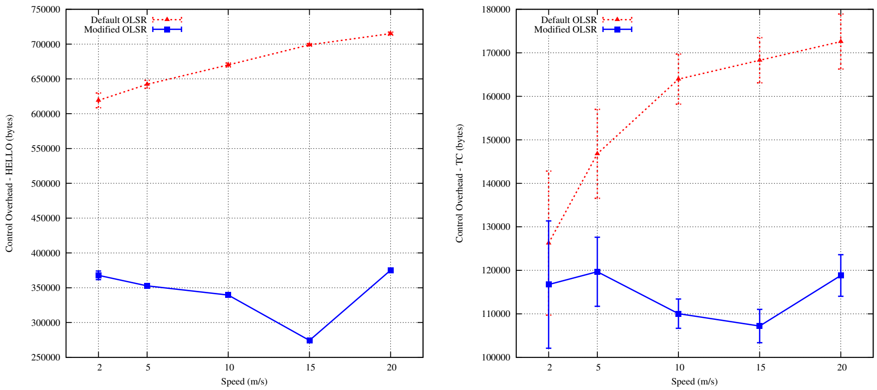
Figure 2(a) plots the control overhead transmitted during the simulation for speeds $2, 5, 10, 15$ and $20\,m/s$ with packet arrival rate of $10\,pkts/s$ for one data flow. The results show an enormous improvement in the control overhead being transmitted. The locally proactive protocol reduces the control overhead by 35% for speed $2\,m/s$ which reduces further to 56% for $15\,m/s$ and then a 44% reduction is observed when the speed increases to $20\,m/s$. Increasing data flows or packet arrival rate has a slight effect on transmitted control bytes. The plots look similar and are therefore not included here.

Figure 2(b) shows the control overhead incurred to transmit a data packet successfully as a function of node speed. At a speed of $2\,m/s$, control overhead per data packet transmitted by OLSR is 1.62 packets and by the locally proactive



(a) Control overhead (b) Control overhead per data packet

Fig. 2. Control overhead and control overhead per data packet transmitted for one data flow with packet arrival rate $10\,pkts/s$



(a) *HELLO* overhead

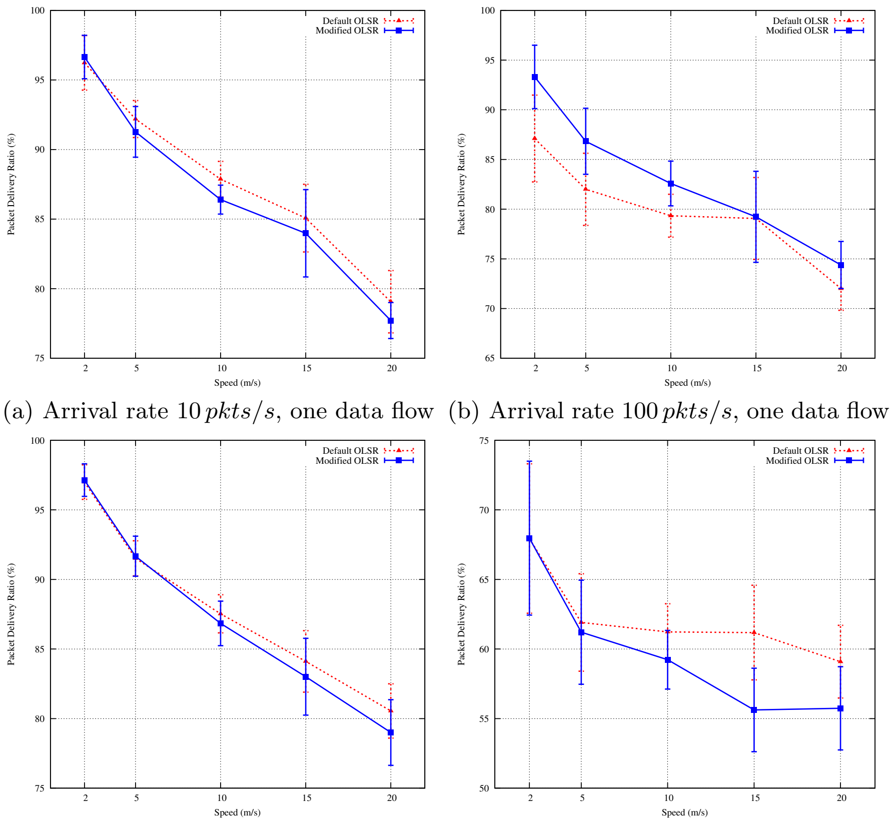
(b) *TC* overhead

Fig. 3. *HELLO* and *TC* overhead transmitted for one data flow with packet arrival rate $10\,pkts/s$

protocol is 1.1 packets, 32.07% fewer in the locally proactive protocol. When speed increases to $20\,m/s$, the control overhead per data packet transmitted by OLSR increases to 2.2 packets; the locally proactive OLSR only sends 1.31 packets as control overhead, a reduction of 40.56%. Furthermore, the locally proactive OLSR shows a saving of 51.24% on the control overhead sent at speed $15\,m/s$.

Figure 3 shows the plots of the two different kinds of control messages in OLSR, the HELLO and TC messages. At $2\,m/s$, the locally proactive OLSR protocol sends 40.59% less control overhead in terms of HELLO messages compared to OLSR. With an increase in speed the savings increase to almost 60% for $15\,m/s$ and are 47.53% for $20\,m/s$. The savings in TC messages are smaller. There is a 7.38% and 31.11% saving in control overhead of TC messages for speeds $2\,m/s$ and $20\,m/s$, respectively. The reduction in TC messages reach 36.25% at $15\,m/s$ speed. The locally proactive protocol seems to be best tuned for this speed. This data set is representative of all the collected data; other sets are not presented here.

Figures 4(a) and (b) show the packet delivery ratio for one flow when the packet arrival rate is changed from $10\,pkts/s$ to $100\,pkts/s$ while Figures 4(c) and (d) plot the packet delivery ratio for two flows. With one flow, using $10\,pkts/s$ arrival rate, the packet delivery ratio for OLSR drops down from 96.2% to 79% as the speed is increased from $2\,m/s$ to $20\,m/s$. When the locally proactive OLSR protocol is used, the packet delivery ratio improves slightly; the values are 96.7% and 77.7%, respectively. When the packet arrival rate is increased to $100\,pkts/s$, the packet delivery ratio for OLSR falls to 87.1% for $2\,m/s$ and 72% for $20\,m/s$. The locally proactive protocol obtains a better packet delivery ratio; the values are 93% and 74% for $2\,m/s$ and $20\,m/s$.

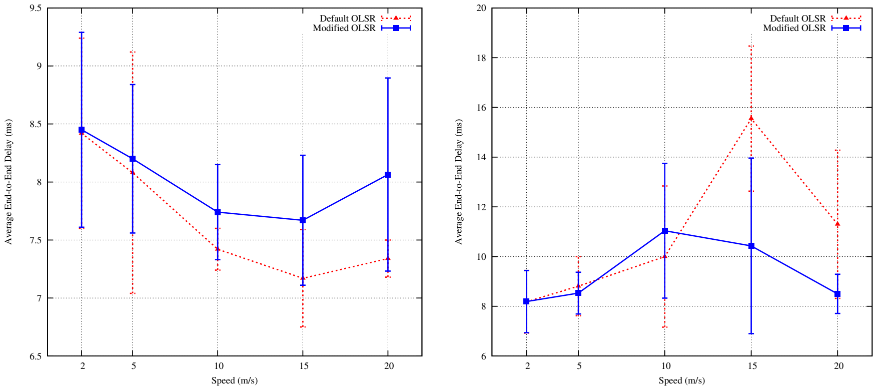


(a) Arrival rate 10 $pkts/s$, one data flow (b) Arrival rate 100 $pkts/s$, one data flow

(c) Arrival rate 10 $pkts/s$, two data flows (d) Arrival rate 100 $pkts/s$ two data flows

Fig. 4. Packet delivery ratio with one and two flows and packet arrival rate 10 and 100 $pkts/s$

Increasing the number of flows slightly increases the packet delivery ratio for the locally proactive protocol when the packet arrival rate is 10 $pkts/s$ as shown in Figures 4(c) and (d). The default OLSR protocol shows a higher packet delivery ratio. In case of OLSR it is 97% and 81% for 2 m/s and 20 m/s, respectively. The locally proactive protocol achieves 97.1% at 2 m/s. When the node speed is increased to 20 m/s, the packet delivery ratio drops to 79%. Further increases in the packet arrival rate reduces the packet delivery ratio. Observations reveal that packet delivery ratio drops to 68% and 66% respectively for OLSR and the locally proactive OLSR at a speed of 2 m/s and arrival rate of 100 $pkts/s$. An increase in speed reduces the packet delivery ratio; it reaches 59%, and 56% respectively for the two protocols at 20 m/s. The reduction in the packet delivery ratio can be attributed to increased contention for the transmission medium as the packet arrival rate increases.



(a) One data flow

(b) Two data flows

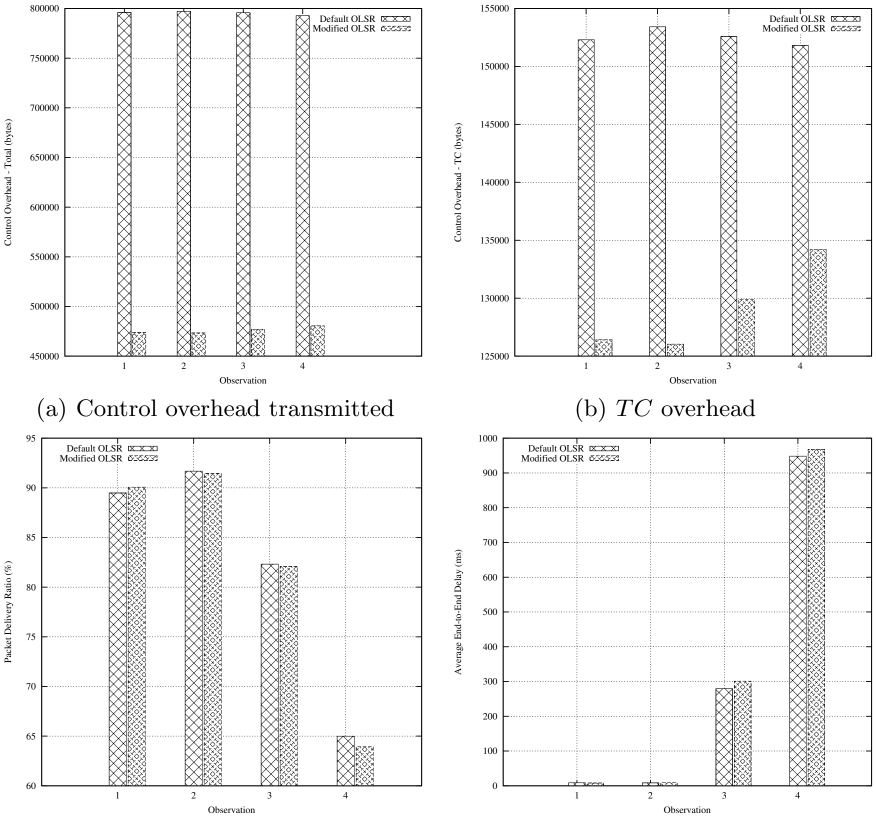
Fig. 5. Average end-to-end delay for one and two data flows with packet arrival rate $10\,pkts/s$

Since there is a drop in the packet delivery ratio, it is important to know if the decrease is statistically significant. *F-tests* and *t-tests* were performed on the mean values of the original and locally proactive OLSR protocols to determine the statistical significance of the drops. As a first step, *F-test* results were observed to determine the variance in the two data sets. They were found to have the same variance as $P(F \leq f)$ values were larger than 0.05. The next step was to perform *t-tests* on the means for equal variance. The *t-tests* show that $P(T \leq t)$ values were greater than 0.05 and indicate statistical insignificance in most cases. Exceptions were observed when the packet arrival rate was increased to $80\,pkts/s$ and $100\,pkts/s$ with two data flows; packet loss rates are higher in these cases.

Figures 5(a) and (b) plot the average end-to-end delay for one and two data flows with packet arrival rate $10\,pkts/s$. At $2\,m/s$, OLSR incurs a delay of $8.42\,ms$ while the locally proactive protocol has a delay of $8.45\,ms$. When the speed increases to $20\,m/s$ the delay for each protocol decreases to 7.34 and $8.06\,ms$ respectively. The locally proactive protocol has higher delay by 10%. The decrease in the average end-to-end delay can be attributed to the fact that the data packets that are successfully routed to the destination are done so very fast. When we increase the number of data flows, OLSR and the locally proactive protocol have a delay of 8.18 and $8.19\,ms$ respectively at $2\,m/s$. For $20\,m/s$ the protocols show a delay of 9.98 and $8.5\,ms$; there is a 24.81% reduction in delay.

4.2 Results for Variable Speed Scenarios

We now provide results when the simulations were run with variable speeds during the simulation period. Four such scenarios were generated each with ten different seeds. For brevity, only the results for one scenario are presented as the



(a) Control overhead transmitted

(b) TC overhead

(c) Packet delivery ratio

(d) Average end-to-end delay

Fig. 6. Variable speed scenarios showing control overhead, TC overhead, packet delivery ratio and average end-to-end delay

others are similar. In this scenario the nodes start moving with an average speed of $2\,m/s$ which is then changed to $5\,m/s$ after $83.33\,s$. The nodes change their speeds every $83.33\,s$ by following the sequence of average speed $10, 15, 20$, and $2\,m/s$ respectively. Thus, total duration of the simulation is $500\,s$ (83.33×6).

Figure 6 summarizes the observed results. Here, the observations 1 and 2 represent scenarios with packet arrival rate $10\,pkts/s$ for one and two data flows. Observations 3 and 4 represent the same scenarios with a packet arrival rate of $100\,pkts/s$. These results confirm the results presented earlier: a 40% drop in control overhead is observed by the locally proactive protocol. The packet delivery ratio of the locally proactive protocol follows that of OLSR closely. Once again the means is statistically identical. The average end-to-end delay of the locally proactive protocol is within 1% of the original OLSR protocol. The locally proactive protocol exhibits stable behaviour.

5 Conclusions and Future Work

The main goal of this work was to develop a *locally proactive* OLSR protocol, i.e., one in which each node periodically transmits control packets using a time interval based on its local network conditions. The objective was to optimize the timer intervals to jointly maximize the packet delivery ratio and minimize the control overhead. Using this approach, the control overhead is reduced by 35-44 % (up to 56% in some cases). In most cases the decrease in the packet delivery ratio is statistically insignificant. There is a slight increase in average end-to-end delay.

While the DOE experiments were performed for node speeds of $5, 10, 15$ and $20\,m/s$, the results include simulations run at $2\,m/s$ speed to find out how the protocol extrapolates scales for lower speeds. Observing the extrapolative behaviour of the locally proactive protocol at the other end of the spectrum, i.e, higher speed will be interesting.

The factors considered for DOE were node speed, and the HELLO and TC intervals. Since, there is a 35-44% decrease in the number of control bytes transmitted in the locally proactive protocol, one can expect that the freed bandwidth can be used to route more data packets and achieve a higher packet delivery ratio. Though the drop in the packet delivery ratio is statistically insignificant, a question arises as to whether there are other factors playing key roles in protocol performance. More investigation is required in this area to figure out what other factors may significantly affect packet delivery ratio.

Since the empirical models returned by DOE are quadratic, it would be interesting to find out how a central composite experimental design [7] fits the data rather than a general full factorial experimental design.

The results for the variable scenarios are stable and closely follow the original protocol in packet delivery ratio and average end-to-end delay. It would be interesting to observe how the locally proactive protocol behaves in scenarios where the speeds vary more widely. As well incorporating delay as an objective will be interesting.

References

1. Barrett, C., Marathe, A., Drozda, M., Marathe, M.: Characterizing the interaction between routing and MAC protocols in ad-hoc networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 92–103. ACM, New York (2002)
2. Carnegie Mellon University – Monarch (Mobile Networking Architectures) Project, Wireless and mobility extensions to ns-2, http://www.monarch.cs.cmu.edu
3. Clausen, T., Jacquet, P.: Optimized link state routing protocol, RFC 3626 (October 2003), http://hipercom.inria.fr/olsr/rfc3626.txt
4. Johnson, D.B., Maltz, D.A., Hu, Y.-C., Jetcheva, J.G.: The dynamic source routing protocol for mobile ad hoc networks (DSR) (November 2001) (internet draft), http://www.ietf.org/ids.by.wg/manet.html

5. McClary, D.W., Syrotiuk, V.R., Kulahci, M.: Steepest-ascent constrained simultaneous perturbation for multi-objective optimization. Transactions on Modeling and Computer Simulation 21(1) (January 2011)
6. Ming, L., Zhao, G., Xie, G., Kuang, X.: HOLSR: A novel routing scheme of ad hoc wireless networks for pervasive computing. In: Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA 2007), pp. 661–666 (July 2007)
7. Montgomery, D.C.: Design and Analysis of Experiments, 7th edn. Wiley, Chichester (December 2009)
8. Navidi, W., Camp, T.: Stationary distributions for the random waypoint model. IEEE Transactions on Mobile Computing 3(1), 99–108 (2004)
9. Ogier, R., Templin, F., Lewis, M.: Topology dissemination based on reverse-path forwarding, RFC 3684 (February 2004), http://rfc3686.x42.com/
10. Pei, G., Gerla, M., Chen, T.-W.: Fisheye state routing: a routing scheme for ad hoc wireless networks. In: Proceedings of IEEE International Conference on Communications (ICC 2000), vol. 1, pp. 70–74 (2000)
11. Perkins, C.E., Royer, E.M.: Ad hoc on-demand distance vector routing (AODV). In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, pp. 90–100 (February 1999)
12. Ros, F.J., Ruiz, P.M.: Cluster-based OLSR extensions to reduce control overhead in mobile ad hoc networks. In: Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, pp. 202–207 (2007)
13. Shaukat, K., Syrotiuk, V.R.: Using monitoring to control a proactive routing protocol. Ad Hoc & Sensor Wireless Networks 6(3-4), 299–319 (2008)
14. The University of California, Berkeley, The network simulator – ns-2, http://www.isi.edu/nsname/ns/
15. Totaro, M.W., Perkins, D.D.: Using statistical design of experiments for analyzing mobile ad hoc networks. In: Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005), pp. 159–168. ACM, New York (2005)
16. Vadde, K.K., Syrotiuk, V.R.: Factor interaction on service delivery in mobile ad hoc networks. IEEE Journal on Selected Areas in Communications 22(7), 1335–1346 (2004)
17. Vadde, K.K., Syrotiuk, V.R., Montgomery, D.C.: Optimizing protocol interaction using response surface methodology. IEEE Transactions on Mobile Computing 5(6), 627–639 (2006)
18. Xue, Y., Jiang, H., Hu, H.: Rough Sets and Knowledge Technology. Springer Link, Heidelberg (2008)

Coverage-Adaptive Random Walks for Fast Sensory Data Collection*

Constantinos-Marios Angelopoulos[1,2], Sotiris Nikoletseas[1,2],
Dimitra Patroumpa[1,2], and Jose Rolim[3]

[1] Research Academic Computer Technology Institute, 26504 Patras, Greece
{nikole,patroumpa}@cti.gr
[2] Department of Computer Engineering and Informatics, University of Patras, Greece
aggeloko@ceid.upatras.gr
[3] Centre Universitaire d' Informatique, Geneva, Switzerland
rolim@cui.unige.ch

Abstract. Random walks in wireless sensor networks can serve as fully local, very simple strategies for sink motion that significantly reduce energy dissipation a lot but increase the latency of data collection. To achieve satisfactory energy-latency trade-offs the sink walks can be made adaptive, depending on network parameters such as density and/or history of past visits in each network region; but this increases the memory requirements. Towards better balances of memory/performance, we propose three new random walks: the Random Walk with Inertia, the Explore-and-Go Random Walk and the Curly Random Walk; we also introduce a new metric (Proximity Variation) that captures the different way each walk gets close to the network nodes over time. We implement the new walks and experimentally compare them to known ones. The simulation findings demonstrate that the new walks' performance (cover time) gets close to the one of the (much stronger) biased walk with memory, while in some other respects (partial cover time, proximity variation) they even outperform it. We note that the proposed walks have been fine-tuned in the light of experimental findings.

Keywords: random walks, wireless sensor networks, data collection.

1 Introduction, Related Work and Contribution

Wireless Sensor Networks are visioned as large ad-hoc collections of very small autonomous devices, that can sense environmental conditions in their immediate surroundings while having limited processing and communication capabilities and energy reserves. This type of Wireless Networks are ideal for Smart/Green Building Management Systems as they provide rigorous monitoring and automated solutions with no special requirements in infrastructure. This way, they help save great amounts of energy (automated light, air-conditioning control,

* This work has been partially supported by EU/FIRE HOBNET project - STREP ICT-257466.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 81–94, 2010.
© Springer-Verlag Berlin Heidelberg 2010

etc) and material (since no wires are needed). The sink mobility strategy we propose can in fact abstract sensor data collection by entities (e.g. humans carrying mobile phones) moving in the building.

The collected sensory data is usually disseminated to a static control center (called data sink) in the network, using node to node *multi-hop* data propagation. Such settings have increased implementation complexity and sensor devices consume significant amounts of energy, since a distributed routing protocol for disseminating data towards the sink is executed in each sensor node. Furthermore, in the area around the control center, nodes need to heavily relay the data from the entire network, thus a hotspot of increased energy consumption emerges and failures due to strained energy resources of these nodes leads to a disconnected and dysfunctional network (see e.g. [1]).

Towards a more balanced and energy efficient method of data collection sink mobility can be used. The main idea is that the sink has significant and easily replenishable energy reserves and can move inside the region the sensor network is deployed, each time being in close proximity to a (usually small) subset of the sensor devices.

This data collection paradigm has many attractive properties. A mobile agent that moves closer to the nodes can help conserve energy since data is passively transmitted to the sink. Connectivity of the network is not required, thus sparse networks can be better handled, and additionally, fewer sensor devices may be deployed, to reduce the operational cost of the network.

However, many apparent difficulties arise as well since traversing the network area in a timely and efficient way is critical. Failure to visit some areas of the network will result in data loss, while infrequently visiting some regions will introduce high delivery delays.

There are many different approaches when considering the mobility pattern that the mobile sink should follow. A very attractive one is the *random movement*. Random walks can serve as local, very simple strategies for sink motion, that reduce energy dissipation a lot but increase latency. Different random mobility methods define various degrees of freedom of the movement such as allowing only predefined locations or moving only in certain directions (i.e. north, south etc). Also, different patterns assume different levels of locality, network information and memory.

To achieve satisfactory energy-latency trade-offs the sink walks can be made adaptive, depending on network parameters such as density and/or history of past visits in each network region, but this requires additional memory.

Related Work. Random Walks have been extensively studied in the past decades in the context of several disciplines. However, despite their numerous applications in ad hoc and overlay networks, not much research has been done on how they can be particularly applied in sensor networks with mobile entities, in a way that addresses the peculiarities of such networks (such as severe computing and communication constraints, the small memory and constrained battery, as well as the time-criticality of important applications).

A comparison of different random walk strategies for ad hoc networks is performed in [2]. The authors investigate the effectiveness of each strategy in terms of the expected hop count and the occurrence of deadlocks. In our work, we propose three new random walk strategies and compare them with four known ones. We study the performance of the strategies in terms of cover time, partial cover time and proximity variation, a new metric that we introduce here.

In [3] the authors study the problem of data collection in a sensor network using a mobile sink. Towards improved latency, they propose a biased, adaptive sink mobility scheme. The sink moves probabilistically, following a biased random walk that favours less visited areas in order to cover the network area faster, while it adaptively stops more time in network regions that tend to produce more data. Our proposed random walk strategies instead use only the information of their previous position in order to decide their next move, in contrast to [3] which assumes memory of O(n), where n is the number of nodes of the graph. However, despite the very small memory that we here use, we succeed to cover the network much faster than a simple random walk and in some respect (partial cover time, proximity variation) we even outperform the powerful biased walk of [3].

In [4] the problem of data gathering in a large-scale wireless sensor network with static nodes and a mobile patrol node is formulated as a classical random walk on a random geometric graph. The authors derive analytical bounds for the performance of the random walk in terms of node coverage. In order to improve this performance they propose an algorithm to constrain the random walk using the available side information, such as the awareness of previously visited sites. We also use some information (just the information of the previous position) in order to speed up the network traversal and our new walks are different.

In [5] authors investigate the theoretical aspects of the uneven energy depletion phenomenon around a sink, and address the problem of energy-efficient data gathering by mobile sinks. Authors also present a taxonomy and a comprehensive survey of state of the art on the topic.

Random walks have also been used to model different types of interactions in mobile computing scenaria, such as in [6,7] and to analyse communication in mobile computing, e.g. in [8,9] the authors propose protocols which exploit the coordinated, random motion of a small part of the network; they also propose a methodology for the analysis of the expected behaviour of protocols for such networks, based on the assumption that mobile hosts (whose motion is not guided by the protocol) conduct concurrent random walks in their motion space. In contrast our walks are not blind but instead adapt to the progress of the data collection process.

In [10] authors investigate the *Random Waypoint Model*, which is widely used in the simulation studies of mobile ad hoc networks and show that it fails to provide a steady state in that the average nodal speed consistently decreases over time; therefore it should not be directly used for simulation. They also propose a simple fix of the problem and discuss a few alternatives.

Our Contribution. In this paper we first investigate further four already known walks particularly suitable for data collection in wireless sensor networks, where sensors are static: the *Blind Random Walk*, the *Random Walk with Memory One*, the *Random Walk with Memory Two* and the *Biased Random Walk*. In all cases we abstract the network region by a lattice (grid) graph. The first walk is the simplest possible method a mobile sink can adopt in order to perform data collection, and as such it has minimum requirements in terms of memory, since each step is independent of the previous ones; however latency can become very high. The Random Walks with Memory One and Memory Two improve the performance of the simple random walk using memory, i.e. the sink remembers its previous and its two previous positions, respectively. The Biased Walk, in contrast, assumes that the sink has memory of size $O(n)$, where n is the number of vertices of the lattice abstracting the network. Based on the history of visits, the sink tries to adapt its trajectory in order to collect sensory data quickly and efficiently by avoiding already visited vertices. Somehow the Blind Random Walk and the Biased Random Walk represent extremes in the spectrum of possible configurations of memory requirements and achieved performance.

Towards improved trade-offs, we propose three new walks that require a small memory of constant size. The first one is the *Random Walk with Inertia* where the sink tends to keep the same direction as long as it discovers new nodes, while changing direction when it encounters already visited ones. Although it may appear to have a similar behaviour to the random waypoint model, however they radically differ in the rationale. While the random waypoint model arbitrarily chooses a random point in the network area as the next destination, the Random Walk with Inertia makes this decision based solely on local criteria and not in an arbitrary way. The second one is the *Explore-and-Go Random Walk*, where as long as there are undiscovered nodes on the nearby sub-regions of the network it tends to make a Brownian-like motion until all this area is covered. When no new sensors are discovered, it performs a more or less straight-line walk in order to move to a different, possibly unvisited area. The last one is the *Curly Random Walk* where the sink traverses the network area beginning from the center and expanding its traversal to the entire network area with consecutive circular-like moves. We show, through experimental evaluation, that these three walks improve significantly the total number of hops needed to collect the sensory data compared to the Blind Random Walk, while remainning light-weight in terms of memory needs and computational complexity. In fact, the performance (cover time) of the new walks get close to the one of the (stronger) biased walk, and even outperform it in some respects (i.e. partial cover time, proximity variation). Finally, we introduce a new metric for the performance evaluation (proximity variation) that captures how close each walk gets to the network nodes over time.

2 The Network Model

Sensor networks are comprised of a vast number of ultra-small homogeneous sensor devices (which we also refer to as sensors). Each sensor is a fully-autonomous computing and communication device, characterized mainly by its available power supply (battery), the energy cost of data transmission and the (limited) processing and memory capabilities. Sensors (in our model here) do not move. The positions of sensor nodes within the network area are random and in the general case follow a uniform distribution. We focus on data collection methods, so we assume that initially all sensors have sensory data to deliver to the sink. We also assume that no data is generated during the network traversal. This way, a cell is called "visited" when the sensors it contains have no data to send to the sink.

Sensor devices are equipped with hardware monitors that measure environmental conditions of interest, including light, pressure, humidity, temperature etc. Each device has a *broadcast* (digital radio) *beacon mode* of fixed transmission range $\mathcal{R}$, and is powered by a battery. Also a sensor is equipped with a *general purpose storage memory (e.g.FLASH) of size* $\mathcal{C}$.

There is a special node within the network region, which we call the sink S, that represents a control center where data should be collected. Here, we assume that the sink is *mobile*. The sink is not resource constrained i.e. it is assumed to be powerful in terms of computing, memory and energy supplies.

The network area A is a flat square region of size D×D. During the network initialization, a graph formation phase is executed by the sink. The network area is partitioned in $j \times j$ equal square regions, called cells. The center of each cell is considered as a vertex in a graph that is connected with undirectional edges only to the four vertices corresponding to adjacent cells. Thus, a virtual lattice graph $G_o = G(V, E)$ is created which is overlayed over the network area. We set $j = \lceil D/\sqrt{2}\mathcal{R} \rceil$, thus when the sink is located at the center of a cell, it can communicate with every sensor node within the cell area. By reducing the walk to an overlay graph we can perform some optimizations more easily; also, our mobility schemes can be deployed in areas of arbitrary topologies as long as we can abstract them by an overlay graph.

3 The Sink Mobility Random Walk Protocols

In this section we describe in detail the four sink mobility protocols that we experimentally evaluate.

3.1 Known Random Walks

Blind Random Walk. The Blind Random Walk on a grid is the simplest of all possible mobility patterns, since the next move of the sink is stochastically independent to the previous one: the sink selects its next position with the same probability for each one of the four coordinate directions. If the mobile element is currently on vertex u of degree deg_u, then the probability of moving to vertex v

$$p(f)_v = \frac{1}{deg_u}$$

This method is very robust, since it probabilistically guarantees that eventually all the cells of the network will be visited and thus data will be collected even from disconnected areas of few/faulty sensors or obstacle presence. However, in some network structures it may become inefficient, mostly with respect to latency, since the sink uses no memory of the past movements in order to select the next one and thus overlaps (i.e. visits to already visited vertices) occur.

Random Walk with Memory. The performance of the Blind Random Walk can be improved using some memory of past visits. The sink maintains a first-in-first-out (FIFO) list $\mathcal{M}$ which contains the last K cells visited during the random walk, i.e $\mathcal{M} = \{c_1, c_2, ..., c_K\}$. The next hop is chosen uniformly among the neighbours of the cell that are not in the memory list $\mathcal{M}$.

The use of memory eliminates loops in random walks, but it may also lead to a deadlock. Without memory, i.e. K = 0, the random walks become blind and can have loops but no deadlocks. For complete memory, the random walks can only have deadlocks and no loops. When the size of $\mathcal{M}$ is $0 \leq K \leq n-1$, the random walks can have both loops and deadlocks.

In this paper, we evaluate the performance of two random walks that use memory of size K = 1 and K = 2 respectively. Furthermore, we examined the performance of a random walk that uses memory of size K = 3 and verified that in the case of a grid topology network the walk results very often to a deadlock, since each cell has degree of four neighbours.

Biased Random Walk. In this walk, the sink associates a counter c_u for every vertex u; initially $c_u = 0 \; \forall u \in V$. When the mobile sink visits cell u it increases the associated counter c_u by 1. Thus, the frequency of visits of each area can be estimated and maintained by the sink. The selection of the next area to visit is done in a biased random manner depending on this variable. If the mobile element is currently in cell u of degree deg_u, then we define

$$c_{neigh}(u) = \sum_{v} c_v$$

for all $v : (u, v) \in E$, i.e. $c_{neigh}(u)$ is the total number of past visits in u's neighbourhood. Then the probability $p(f)_v$ of visiting an adjacent region v is taken as

$$p(f)_v = \frac{1 - c_v/c_{neigh}(u)}{deg_u - 1}$$

when $c_{neigh} \neq 0$. When $c_{neigh} = 0$ we have $p(f)_v = 1/deg_u$. Thus, less frequently visited regions are favoured when the sink is located at a nearby region.

So, with the use of the entire history of visits in each cell, the trajectory of the sink is improved and the time needed for traversing the whole network reduces, since overlaps are probabilistically discouraged. However, this walk requires a memory of size $O(n)$, where n is the number of cells.

3.2 Our New Random Walks

Random Walk with Inertia. In the heuristic Random Walk with Inertia the sink assigns a probability to each one of the four directions, North, South, East and West. At the beginning the probability distribution is uniform, i.e. the sink has the same probability to move towards any of the four directions. Afterwards, the probability distribution on each step of the walk changes adaptively to the nodes' discovery following the principle of *reinforcing the direction where newly discovered sensors were found and weakening the direction where already visited sensors have been located.* For instance, if the sink discovers new sensors after having gone West, then the probability of choosing the same direction for the next step is increased by δ, while the rest probabilities are reduced by $\frac{\delta}{3}$ each, as long as they remain non negative. δ is a constant probability; its specific value is optimized in light of the detailed experimental findings. We considered several $\delta \in [0.1, 0.5]$ and for each one compared the experimental results. The best performance resulted when $\delta = 0.2$ The reverse process of weakening the previously chosen direction and reinforcing the other three is followed when the sink reaches a cell that has already been visited. To summarize, the probability of following the same direction (p$_c$: current direction probability) at time t is:

$$p_c^{t+1} = \begin{cases} p_c^t + \delta \text{ , if new nodes discovered} \\ \\ p_c^t - \delta \text{ , if no new nodes discovered} \end{cases}$$

while each one of the probabilities towards the rest three directions (p$_r$: rest direction probabilities) are:

$$p_r^{t+1} = \begin{cases} p_r^t - \frac{\delta}{3} \text{ , if new nodes discovered} \\ \\ p_r^t + \frac{\delta}{3} \text{ , if no new nodes discovered} \end{cases}$$

Already visited cells are distinguished from newly discovered ones by simply checking if the sensors of the cell have data to send or not, i.e. we first visit and then check if visited or not (this is in contrast to the Biased Random Walk that assumes knowledge of the visit history).

We note that this walk has light-weight requirements. It assumes zero knowledge of the network and is relatively simple with low computational complexity. Furthermore, it requires a small, constant sized memory since the next step of the walk depends solely on the previous one. The sink tends to keep the same

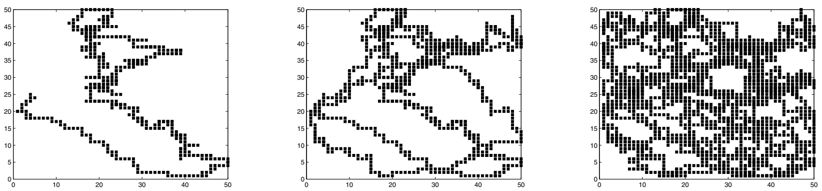


Fig. 1. An example of network traversal following Random Walk with Inertia in a 50 × 50 network area (snapshots after: 500, 1000 and $2 * 10^4$ hops)

direction as long as it discovers new nodes, while it changes its direction when it encounters already visited ones. This way, the sink makes long paths and traverses many different sub-regions of the network area very quickly, thus avoiding early overlaps. However, after most of the network area has been covered, there exist small unvisited sub-regions that are hard to find (see fig.1). At this mature stage of the network traversal, the motion of the sink tends to be very similar to the Blind Random Walk, since already visited cells lie very probably, towards every direction lie already visited cells. The fact that many different sub-regions are visited very soon makes this walk suitable for time critical applications, such as reactive event detection.

Explore-and-Go Random Walk. In this heuristic walk the movement of the sink consists in two types of motion: a) moving on a straight line and b) arbitrarily changing direction. The sink on each step chooses to follow one of these two motions, based on a bias factor β. For example if the bias factor is $\beta = 0.9$, then with 90% probability the sink will move towards the same direction, during the next step, while with 10% probability it will arbitrarily change its direction. There are two different values that are assigned to the bias factor during the walk, a low one and a high one. For as long as the sink keeps discovering new nodes, the bias factor β is assigned the low value. On the contrary, when the sink reaches an area that has already been visited, the bias factor is assigned the high one. Intuitively, this means that as long as there are undiscovered nodes on the nearby sub-regions of the network, the sink tends to make a Brownian-like motion, by frequently changing its direction in an arbitrary way, until all this area is covered. When no new sensors are discovered, it tends to perform a straight-line walk in order to reach a different and possibly unvisited area. The specific values of factor β were optimized in light of the detailed experimental findings, during the algorithm engineering phase. The values that give the best results are as following (F_{motion} gives the next mobility pattern chosen):

$$F_{motion} = \begin{cases} \textit{move straight, with probability } \beta \\ \\ \textit{change direction, with probability } 1 - \beta \end{cases}$$

$$\text{where, } \beta = \begin{cases} 0.1 \text{ , when new nodes were discovered} \\ \\ 0.9 \text{ , when no new nodes were discovered} \end{cases}$$

We note, that this walk has light-weight requirements, as well. It also assumes zero knowledge of the network area and is relatively simple, with low computational complexity. Finally, it requires a small, constant sized memory as the next step of the walk depends only on the current position of the sink. Following this walk allows the sink to systematically cover the network area and therefore greatly reduces overlaps by avoiding to leave small unvisited sub-regions that are difficult to find during the final phases of the network traversal, when most of the network area is covered (see fig. 2). On the other hand there are sub-regions that take too long for the sink to visit. Therefore, this walk is suitable for delay-tolerant applications and scenaria, which however, may require consistent data.

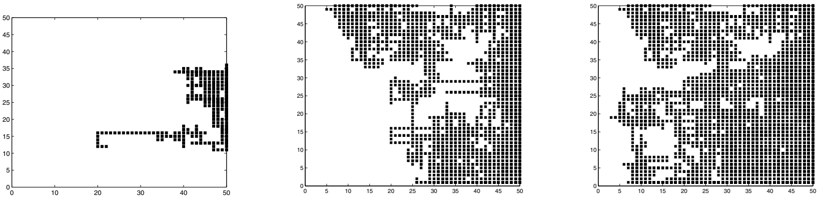


Fig. 2. An example of network traversal following Explore-and-Go Random Walk in a 50×50 network area. (snapshots after 500, 10^4 and $2 * 10^4$ hops.)

Curly Random Walk. The main concept behind this walk is to start by visiting a confined sub-region at the centre of the network and gradually allowing the sink to perform a motion of higher degree of freedom, thus eventually covering the entire network area. We try to achieve this by initially having the sink perform frequent narrow left turns which gradually get wider. This way the central sub-regions of the network are first visited and the rest areas, that lie towards the boundaries of the network, are consecutively visited.

The aforementioned type of motion can be modelled as a series of successive straight and left-turn moves. Let S denote a straight move towards the same direction as the last move and L denote a left turn. A series with the desired properties, that is to initially perform narrow left turns which gradually get wider, could be of the following form:

$$SLSSLSSSLSSSSL...$$

The probability distribution of left turns in this series is the geometric distribution, with probability mass function of i straight moves followed by 1 left turn is $P^i = (1 - p_L)^i p_L$, where p_L denotes the probability of a left turn and i the number of successive straight moves before the next left turn. We call these i straight moves plus the next left turn, $phase_i$.

Note, that during $phase_i$,

$$p_L^i = \frac{1}{i+1}$$

where p_L^i is the probability of a left turn. This is chosen so, towards an i+1 expected number of trials until the first left turn.

However, since each move is stochastically independent of all previous moves, the geometric distribution does not guarantee that the desired sequence of moves will be generated. For instance, sequence SSL has the same probability to occur as sequences SLS and LSS. In order to maximize the likelihood of the desired sequence to occur but also to minimize the gaps among visited nodes in each network sub-region, we allow the sink to remain in $phase_i$ for a certain amount of time. Through experimental fine tuning we identify the best possible choice in the particular setting and allow the sink not to shift to the next phase, until 30 left turns have occurred.

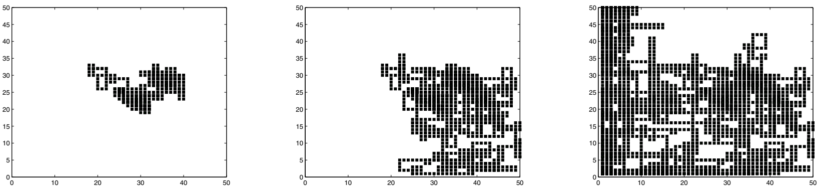


Fig. 3. An example of network traversal following Curly Random Walk in a 50 × 50 network area. (snapshots after 500, $2 * 10^3$ and $2 * 10^4$ hops.)

We call this walk Curly Random Walk because of the fact that it converges to a walk consisting of straight lines with curls, due to successive left turns. It requires very small memory as the only information the sink remembers is the number of left turns it has taken. No information about where or when these turns happened is kept. Here we have to note that the sink can present similar behaviour by increasing i at regular time intervals, which can a priori be decided (i.e. each $phase_i$ lasts for 5 minutes).

4 Performance Evaluation

4.1 Metrics

We measure three metrics that depict the performance of the protocols, *cover time, partial cover time* and a new metric called *proximity variation.* The first one corresponds to the total number of hops (time) needed to cover the whole network area. This metric is similar to latency, a classic metric used in the evaluation of sensor and ad-hoc networks protocols, as it captures the time the sink needs to collect the sensory data from the entire network. Partial cover time is the number of hops needed to cover a specific percentage of the entire network area (usually 90% - 95%). This metric is of great interest as the majority of overlaps occur while the sink tries to locate the last few unvisited sub-regions that are scattered in the network area; however, in most sensor network applications it is sufficient to collect a vast percentage of the total sensory data, so this metric is relevant and informative.

We also introduce a new metric, *the mean value (over all cells) of the smallest distance from the sink for all the cells.* More strictly, let $dist(x)$ be the function that returns the distance of cell x from the sink. Then our metric Proximity Variation (PV), equals:

$$PV = \frac{\sum_{i=1}^{n} min(dist(i))}{n}.$$

where n is the total number of cells and minimum of $dist(i)$ is taken over time.

We now explain the rationale behind our new metric. As forementioned, there lies a different intuition behind each walk on how the network area is traversed, regardless of the total number of hops needed. For example, while the Walk with

Inertia tends to create long straight lines by keeping the same direction as long as it discovers new nodes, the Explore-and-Go Random Walk tries to minimize the gaps of unvisited cells among visited areas by making a Brownian-like motion and systematically visiting neighboring sub-regions. The newly proposed metric captures this difference in their behaviour. As the Walk with Inertia traverses many different sub-regions early in the network traversal process, the sum of the smaller distance each cell has had from the sink decreases significantly faster compared to the Explore-and-Go Random Walk.

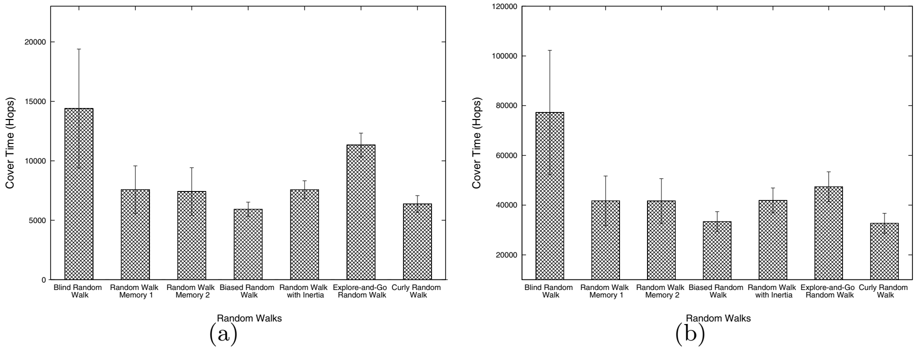
In a real network this corresponds to the way of covering the network area. If the PV metric converges to zero quickly, this means that the sink gets close to all sensors quite soon and data collection progresses fast; this is especially relevant in case when the role of sensors is not completely passive but includes some limited multihop propagation of data to accelerate data propagation at a reasonable energy cost. On the contrary, when the PV converges slowly to zero, it means that the network traversal is performed in a way that some areas may stay unvisited for long time.

4.2 Simulation Set Up

We used Matlab R2008b as our simulation environment. We evaluate the walks for two network dimensions, 25×25 and 50×50 square cells. We also select a random uniform distribution of the sensors over the network area. The width of each cell corresponds to one length unit. For the partial cover time metric, we set the percentage of the network to be covered to 95%. For each network dimension we conduct 2500 iterations and compute the mean values for each metric, as well as the 90% confidence intervals of the findings. In the figures below, the top ends of the bars indicate observation means and the line segments represent the confidence intervals surrounding them. One can observe that our measurements proved statistically smooth in the sense that there is high concentration around the mean values.

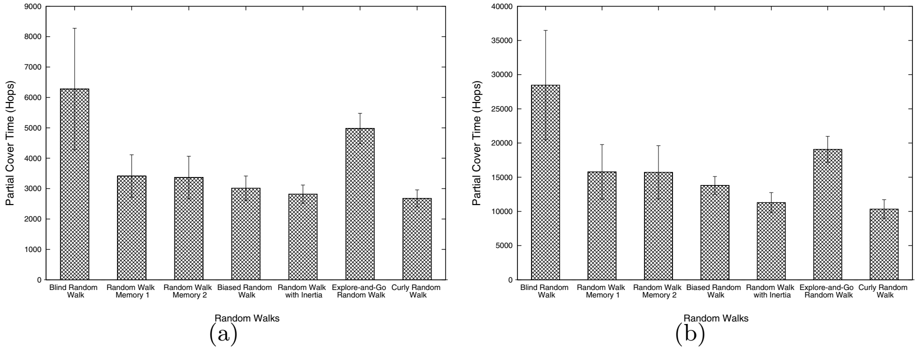
4.3 Findings

The performance of the walks in terms of *cover time*, for two different network sizes, is depicted in fig. 4. As shown in the figure, the Blind Random Walk has the worst performance among the seven mobility strategies. Using memory of size one or two, improves the performance a lot, as we can see in the figure. The Random Walk with Inertia and the Explore-and-Go Random Walk perform significantly better than the Blind Random Walk and their behaviour is comparable with the Random Walks with Memory One and Two. In fact, the Random Walk with Inertia is 45% better than the Blind Random Walk, while the Explore-and-Go Random Walk is 20% better than the Blind Random Walk for the 25×25 grid (fig. 4a) and 40% better for the 50×50 grid (fig. 4b). Furthermore, one can see that the Curly Random Walk's performance is even better than the other two proposed mobility strategies. Especially in the case of the 50×50 network (fig. 4b) it even outperforms the Biased Random Walk which is expected to



(a) (b)

Fig. 4. Cover Time (Number of Hops) for **(a)** 25×25 Grid **(b)** 50×50 Grid



(a) (b)

Fig. 5. Partial Cover Time (Number of Hops) for **(a)** 25×25 Grid **(b)** 50×50 Grid

have the best performance, since it optimizes its trajectory by remembering the whole history of visits in each cell. Furthermore, the performance of the Random Walk with Inertia is only about 20% worse than the performance of the Biased Random Walk, even though it only uses $O(1)$ memory.

Figure 5 shows the *partial cover time* of the seven walks, for the two grids. We observe that the Random Walk with Inertia and the Curly Random Walk cover the 95% of the network faster than the other walks (the Curly Random Walk is slightly better). These two walks perform better even than the stronger Biased Random Walk. The Explore-and-Go Random Walk is faster than the Blind Random Walk, too.

Figure 6 depicts the mean value of the smallest distance from the sink for all the cells over the total number of hops the sink has made for a network of 50×50 size. We compute the evolution of this metric until the entire network area is covered. Since the initial position of the sink for all the experiments is the center of the network area, at the beginning of each experiment this metric has the same value for all the walks. Also, because of the fact that each experiment is

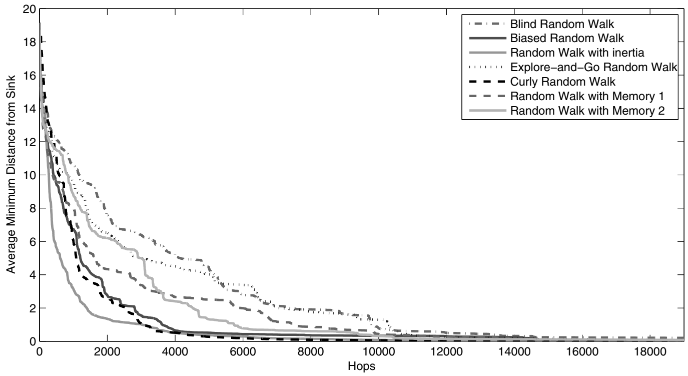


Fig. 6. The Proximity Variation for a 50 × 50 network

concluded when the whole network area is covered (all nodes are visited at least once), this metric converges to zero towards the end of each experiment.

However, because of the different way each walk traverses the network, significant differences are observed mainly in the speed with which zero is approximated. In particular, the Blind Random Walk converges very slowly to zero, followed by the Explore-and-Go Random Walk. This is because these two walks tend to move slowly inside the network area by frequently changing direction. On the contrary, the Random Walk with Inertia converges to zero very quickly because of the long paths it creates during the network traversal. These long paths allow the sink to quickly approach, within a few hops distance, cells of different sub-regions. The Curly Random Walk converges to zero very fast, too, since the circular expansion of the sink's movement on the network area results to a quick approach of different sub-regions.

5 Conclusions

In this paper we proposed the Random Walk with Inertia, the Explore-and-Go Random Walk and the Curly Random Walk aiming at accelerating data collection by a mobile sink in wireless sensor networks. We experimentally evaluated (and carefully fine-tuned them) and compared them to four known walks. We showed that although the new walks use only slightly more memory than the Blind Random Walk (constant memory) they cover the network area much quicker. In fact, their performance is close to the performance of the powerful Biased Walk that uses $O(n)$ memory (and in some respect even outperforms it). Finally, we introduced a new metric, Proximity Variation, that captures the differences on how each walk traverses the network with time and showed through our experiments that these differences do exist and should be taken into account when choosing a sink mobility algorithm. This work can be extended by considering other adaptation methods that use memory and history in different ways. Also, by suggesting new metrics and properties of the walk motivated by the relevant network domain and investigating additional topologies, shapes and densities. Finally, we plan to carry out a rigorous probabilistic analysis of the proposed random walks.

References

1. Efthymiou, C., Nikoletseas, S., Rolim, J.: Energy balanced data propagation in wireless sensor networks. Wireless Networks (WINET) Journal, 691–707 (2006)
2. Dhillon, S.S., Mieghem, P.V.: Comparison of random walk strategies for ad hoc networks. In: The Sixth Annual Mediterranean Ad Hoc Networking WorkShop (June 2007)
3. Kinalis, A., Nikoletseas, S., Patroumpa, D., Rolim, J.: Biased sink mobility and adaptive stop times for low latency data collection in sensor networks. In: GLOBECOM 2009: Proceedings of the 52nd IEEE International Global Communications Conference (2009)
4. Lima, L., Barros, J.: Random walks on sensor networks. In: WiOpt 2007: 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (April 2007)
5. Li, X., Nayak, A., Stojmenovic, I.: Sink mobility in wireless sensor networks. In: Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication. Wiley, Chichester (2010)
6. Dimitriou, T., Nikoletseas, S.E., Spirakis, P.G.: The infection time of graphs. Discrete Applied Mathematics, 2577–2589 (2006)
7. Nikoletseas, S.E., Raptopoulos, C., Spirakis, P.G.: The survival of the weakest in networks. In: WAOA, pp. 316–329 (2006)
8. Dimitriou, T., Nikoletseas, S.E., Spirakis, P.G.: Analysis of the information propagation time among mobile hosts. In: Nikolaidis, I., Barbeau, M., Kranakis, E. (eds.) ADHOC-NOW 2004. LNCS, vol. 3158, pp. 122–134. Springer, Heidelberg (2004)
9. Chatzigiannakis, I., Nikoletseas, S.E., Spirakis, P.G.: An efficient communication strategy for ad-hoc mobile networks. In: Welch, J.L. (ed.) DISC 2001. LNCS, vol. 2180, pp. 285–299. Springer, Heidelberg (2001)
10. Yoon, J., Liu, M., Noble, B.: Random waypoint considered harmful. In: Proceedings of Infocom 2003, pp. 1312–1321 (2003)

Joint Scheduling and Spectrum Allocation in Wireless Networks with Frequency-Agile Radios

Mohammad Faisal Uddin, Mohammad Nurujjaman, and Chadi Assi

Concordia University, Montreal, Quebec, Canada

Abstract. We study the benefits of optimal spectrum allocation in a wireless network with frequency agile radios and we present a cross-layer problem formulation for the joint routing and link scheduling under non-uniform spectrum allocation. We present a primal-dual decomposition to provide an exact solution for this complex optimization problem. Given the difficulty associated with such design, we propose a heuristic approach based on simulated annealing to solve the dual sub-problem of the decomposed model. Numerical results revealed that up to 44% improvement in network performance is obtained when variable-width spectrum band allocation is used, as opposed to the best fixed-width spectrum band allocation for larger networks. Numerical results also confirm that the primal-dual decomposition method using simulated annealing to solve the dual sub-problem, substantially reduces the computation time and achieves near optimal solutions.

1 Introduction

Wireless networks are currently under increasing expectations to provide both reliable and high end-to-end throughput services. Often, however, the capacity of these multihop networks is limited by interference caused by multiple concurrently active nodes, and thus to achieve a high aggregate throughput it is important to control interference while maintaining high concurrency. Methods such as interference cancelation and interference alignment [7] [5] have great potential for mitigating the problems caused by interference; they however involve significant computational complexity and cannot be implemented with commodity hardware [6]. Recently effective interference control is achieved using *variable-width* spectrum allocation and results show significant theoretical and practical capacity improvements [6]. The use of different spectrum widths is supported by current standards and off-the-shelf radio technologies; for example, the 802.11 proposes the use of 5, 10 and 20MHz channel widths for operation in different parts of the spectrum and wider widths may be obtained by *bonding* smaller width channels. Recent developments [3] show that the channel width may be configured dynamically and such adaptation brings unique benefits in terms of increasing communication range and throughput. The authors [3] have shown that with simple software modification, network interface cards on commodity hardware (e.g., Atheros) may communicate at 5, 10, 20 and 40MHz channels;

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 95–108, 2010.
© Springer-Verlag Berlin Heidelberg 2010

this potential has been exploited by [12] to improve the throughput of infrastructure networks by dynamically allocating spectrum to access points (APs) based on their loads. Modern radios, e.g., software defined and cognitive radios, are frequency agile and have recently received a lot of attention due to their ability of enabling very flexible spectrum access through their spectrum sensing capability and ability to dynamically reconfigure the allocated spectrum [2] [4] [13] [8]. Frequency agile radios partition the spectrum into several subchannels (e.g., OFDMA subcarriers which are in turn grouped into subchannels) of equal size and access the medium either through a block of contiguous number of subchannels (1-agile radio) or through a set of non contiguous subchannels which need not necessarily be frequency aligned [2]. This latter form of agile radio requires more sophisticated signal processing and hence increased hardware complexity. The former one, however, may be implemented through commodity WiFi hardware [3]. In this paper, we assume a multihop wireless network with WiFi-like radios, where these radios can be configured dynamically to transmit on any spectrum block of contiguously aligned subchannels. We investigate the benefits of such flexible spectrum partitioning through a cross-layer optimization which incorporates multi-path routing and link scheduling. Link scheduling determines the set of transmission links which can be concurrently active without violating the interference constraints. Under optimal scheduling, one needs to determine the spectrum allocation on active links to satisfy the traffic demands. The set of active transmission links along with their spectrum assignment are referred to as a transmission configuration. The joint optimization can then be obtained by constructing the whole set of such configurations. We model mathematically this combinatorial complex problem which requires enumeration of all possible configurations, and we present a primal-dual decomposition method, based on column generation, for solving it without exhaustive enumerations. Numerical results confirm that *variable* spectrum band allocation achieves significant improvement over *preset* spectrum widths. Our investigation has also shown that even with the decomposition, the complexity, and thus scalability, of the problem remains a major design obstacle. In our problem formulation, the dual sub-problem deals with a large number of integer variables which is quite hard (ILP) and time consuming to solve. To circumvent the complexity issue, we propose a heuristic based on simulated annealing (SA) to solve the dual sub-problem. Numerical results prove that using heuristic model to solve the dual sub-problem performs quite accurately and it and much faster than the ILP model. In the rest, the system model is presented in Section 2. We present the problem formulation in Section 3 and the primal-dual decomposition in Section 4. Section 5 presents a simulated annealing method for solving the dual sub-problem. Numerical results are given in Section 6 and we conclude in Section 7.

2 System Model

We model a multihop wireless network with N nodes as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of possible radio links. We

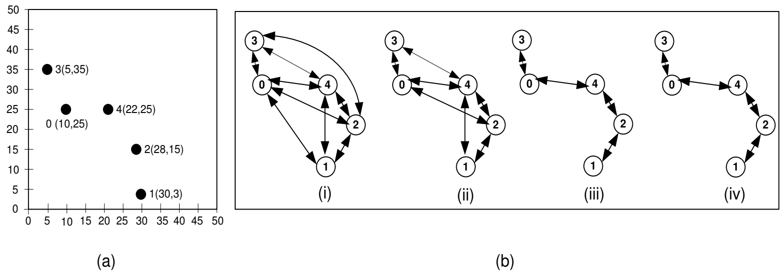


Fig. 1. (a) Network (b) Multi-layer graphs (i) 5MHz (ii) 10MHz (iii) 20MHz (iv) 40MHz

consider a total system spectrum of BMHz which may be divided into smaller spectrum blocks or sub-bands of non-uniform size, each block is identified by its start and end frequencies and we refer to it as a channel. The width of a sub-band is selected from the set $W = \{5, 10, 20, 40\}$MHz. We assume that each radio can dynamically configure the channel position and channel width in the permissible spectrum band by changing the start and end frequencies of its spectrum block. We model a network as a multilayer graph, each layer corresponds to one of the possible spectrum widths: $\{G_k = (V, E_k), w_k \in W\}$ and E_k is the set of possible radio links *permissible* under channel width w_k. Thus, $G = (V, E = \cup E_k)$ is the directed graph representation of the network. Consider a single transmission link between two neighboring nodes i and j that are separated by a distance d_{ij} from each other; in the absence of any interference, the signal to noise ratio (SNR) at the intended receiver is determined as:

$$SNR_{ij}^{k} = \frac{P_t g_{ij}}{N_0 w_k} \tag{1}$$

where, P_t is the transmission power, g_{ij} is the channel gain, N_0 is the power spectral density of the thermal noise and w_k is the channel bandwidth. When the received power is modeled by path loss with exponent $\alpha \geq 2$, then $g_{ij} = d_{ij}^{-\alpha}$. For fixed P_t and N_0, the SNR depends on d_{ij} and w_k. Eq.(1) shows that on a particular link, *smaller width yields a higher SNR at the intended receiver than a larger channel width.* The SNR must be larger than or equal to a threshold (β) to meet the transmission requirement. Let T_k be the transmission range on a particular channel of width w_k; by setting $SNR = \beta$ it follows:

$$T_k = \left(\frac{P_t}{N_0 w_k \beta}\right)^{\frac{1}{\alpha}} \tag{2}$$

Fig.1(a) shows a small 5-node network. We assume $\alpha = 2$, $N_0 = 10^{-6}$ watt/MHz, $P_t = 1mWatt$, $\beta = 1.3$ and we show in Fig.1(b) the corresponding multi-layer graphs. For different channel widths, we obtain $T_5 = 39.22m$, $T_{10} = 27.74m$, $T_{20} = 19.61m$, $T_{40} = 13.87m$. Fig.1(b.i) shows the directed graph corresponding to 5MHz width; here [1]$d_{1,3} > T_5$ and thus node 3 is outside the transmission

[1] $d_{1,3}$ is the Euclidean distance between nodes 1 and 3; locations are shown in Fig.1(a).

range of node 1 (and vice-versa). $d_{1,4} \leq T_5$, and nodes 1 and 4 are within transmission range of each other. For $w_k = 20$MHz (Fig.1(b.iii)), $T_{20} = 19.61m$, hence $d_{1,4} > T_{20}$ and nodes 1 and 4 fall outside each other's transmission range. Thus, a link $\ell \in E$ corresponds to a pair of nodes i and j within transmission range of each other and a particular channel width b_ℓ ($b_\ell = w_k \in W$). Now in the presence of concurrent transmission on neighboring links, a transmission may be corrupted as a result of strong interference caused by active links on the overlapping portions of the spectrum. Considering the cumulative effects of interference, a transmission is successful if the $SINR$ at the intended receiver is above a certain threshold. Formally, The interference constraint is written as:

$$SINR_\ell = \frac{P_t \cdot g_\ell}{N_0 b_\ell + \sum\limits_{t(\ell') \neq (t(\ell), r(\ell))} P_t \cdot g_{(t(\ell'), r(\ell))} \cdot I_{(\ell, \ell')}} \geq \beta \tag{3}$$

where, $t(\ell)$, $t(\ell')$ and $r(\ell)$, $r(\ell')$ are transmitter and receiver of links ℓ and ℓ' respectively. $I_{(\ell,\ell')}$ ($I_{(\ell,\ell')} \in [0, 1]$) is the interference factor [11] that indicates the fraction of overlap between the spectrum of link ℓ' on ℓ. The value of $I_{(\ell,\ell')}$ depends on the spectrum bands allocated on links ℓ and ℓ'. These spectrum bands are not predetermined, but rather they will be determined by the cross-layer optimization model. This, together with the difficulty associated with the physical model, makes it computational difficult to give solutions, except for very small size networks. We thus adopt a more simplified (protocol) model to characterize the interference constraints. The protocol model describes interference constraints according to a conflict graph, where nodes within a certain distance ($\leq T_k$) can communicate as long as the receiver is separated by a distance $IR_k = T_k(1 + \delta)$ from any active transmitter on a conflicting spectrum band; spectrum bands on two different links are said to conflict if they completely or partially overlap each other. IR_k is referred to as the interference range and δ is a small positive constant [14]. The conflict graph is denoted by $G_{CC} = (V_{CC}, E_{CC})$, where V_{CC} is the set of vertices corresponding to all links in E and E_{CC} is the set of edges. Two vertices in V_{CC} are connected by an edge if the corresponding links in E falls in each others interference range. Let $F_{CC}(= [f_{CC}(\ell, \ell')])$ denote the adjacency matrix of G_{CC}. $f_{CC}(\ell, \ell') = 1$ if links ℓ and ℓ' do not interfere and 0 otherwise. The capacity of the wireless link ℓ (with spectrum width b_ℓ) is a function of the $SINR_\ell$; we assume each link can be viewed as a single user Gaussian channel, and the Shannon Capacity of the link (ℓ) is given by: $C_\ell = b_\ell \log_2(1 + SINR_\ell)$. If we assume a fixed data transmission rate on all links utilizing the same spectrum width, even when the $SINR$ of a particular link exceeds the threshold β [10], then we obtain the following lower bound capacity:

$$C_\ell = b_\ell \log_2(1 + \beta) \tag{4}$$

For illustration, we consider 4 single-hop sessions on links ℓ_1, ℓ_2, ℓ_3 and ℓ_4 (assume, $b_{\ell_1} = 20$MHz, $b_{\ell_2} = 20$MHz, $b_{\ell_3} = 40$MHz, $b_{\ell_4} = 40$MHz), as shown in Fig.2(a). Assume a total system spectrum $B = 80$MHz. Assume further that links ℓ_1, ℓ_2 and ℓ_3 all interfere with each other (and thus cannot be allocated overlapping spectrum). We assume also that ℓ_4 interferes only with ℓ_3. The joint

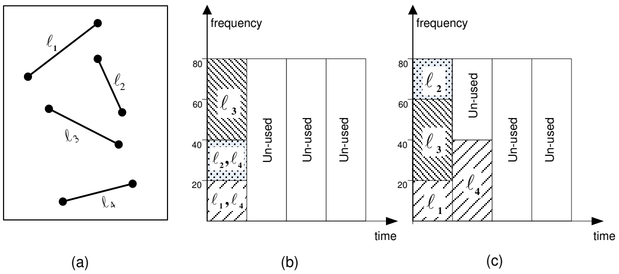


Fig. 2. (a) Network (b) Optimal scheduling and spectrum allocation (c) Non-optimal

optimal link scheduling and spectrum allocation is shown in Fig.2(b) where all links are active at the same time, and the spectrum is properly allocated to satisfy the demands. However, Fig.2(c) shows that if the spectrum bands are not properly allocated, then not all the links in the network may be concurrently active and thus more time slots (network resources) are needed to satisfy the traffic demand (a 50% increase in the schedule length over Fig.2(b)). The feasible solution in Fig.2(c) is obtained under non-optimal spectrum assignment.

3 Problem Formulation

We consider M concurrent multi-hop sessions, each of which corresponds to a source-destination pair (t_m,r_m) in the network. The traffic demand for each session m ($1 \leq m \leq M$) is given by R_m (bits). Traffic of a particular session may be split to sub-flows routed over different paths. The choice of these routing paths depends on the underlying schedule of different concurrent transmissions. The objective of our model is to minimize the system activation time for delivering the traffic without violating the interference constraints. We assume a STDMA access scheme where time is divided into slots and a link may be active in one or more time slots to meet the traffic requirement. We further assume a total spectrum width of B(MHz) and links ($\ell \in E$) assigned spectrum blocks of widths ($b_\ell \in W$) with s_ℓ and e_ℓ being the start and end frequencies of each block.

3.1 Scheduling and Spectrum Band Allocation

We formulate the problem of optimal link scheduling to identify the set of links which may be active concurrently (without violating the interference constraints). We recall from the multi-layer graph representation that a link between two nodes is characterized by b_ℓ (spectrum width allocated on that link). Thus, from now on, the term link is used to identify a pair of nodes and a spectrum band. Note that, since each block may fall anywhere in the permissible spectrum band (B), a block is identified by its start and its end frequencies, whose values are determined under optimal scheduling. Hence, a transmission configuration (p) is defined as the set of concurrently active links, and their corresponding band allocation in the spectrum. We introduce link binary variable, x_ℓ^p; $x_\ell^p = 1$ if link ℓ is active in configuration p, 0 otherwise. Given a single radio per node, a node can either transmit or receive; thus, the radio conflict constraint is:

$$x_\ell^p + x_{\ell'}^p \leq 1 + f_{CR}(\ell, \ell') \qquad \forall(\ell, \ell') \in E \quad \ell \neq \ell' \tag{5}$$

where, $f_{CR}(\ell, \ell') = 1$ if links ℓ and ℓ' do not have any common nodes and 0 otherwise. This constraint (Eq.5) assures that only links with no common radio may be active simultaneously. When link ℓ is active, the start and end frequencies of its allocated spectrum are related by the following:

$$b_\ell x_\ell^p = e_\ell - s_\ell \qquad \forall \ell \in E \tag{6}$$

where, e_ℓ and s_ℓ have integer values. When links ℓ and ℓ' are in each other's interference range and active in the same timeslot, they must be allocated non-overlapping spectrum blocks:

$$e_\ell \leq s_{\ell'} + C(1 - y_{\ell,\ell'}) \qquad \forall f_{CC}(\ell, \ell') = 0 : (\ell, \ell') \in E \;\; \ell \neq \ell' \tag{7}$$

$$e_{\ell'} \leq s_\ell + C y_{\ell,\ell'} \qquad \forall f_{CC}(\ell, \ell') = 0 : (\ell, \ell') \in E \;\; \ell \neq \ell' \tag{8}$$

C is a large constant and $y_{\ell,\ell'} = \{0, 1\}$ is a decision variable. When $y_{\ell,\ell'} = 1$, then Eq.(7) forces the spectrum block assigned on link ℓ to precede (not overlap) that of link ℓ'. Alternatively, when $y_{\ell,\ell'} = 0$, then Eq.(8) forces the spectrum block assigned on link ℓ' to precede that of link ℓ. Note that, when ℓ and ℓ' are not adjacent (or in interference range) of each other, then both links may be assigned overlapping spectrum blocks. The following constraint ensures that any band allocation to a link ℓ must fall inside the permissible spectrum B:

$$e_\ell \leq B \times x_\ell^p \qquad \forall \ell \in E \tag{9}$$

3.2 Routing

We assume multi-path routing and the choice of routing depends upon the scheduling of active concurrent transmissions. Let $\mathcal{P}$ ($|\mathcal{P}| = \bar{P}$) denote the set of all feasible transmission configurations for a network, and x_ℓ^p is a link binary parameter that indicates whether ℓ is active in configuration p ($p \in \mathcal{P}$) or not. Define λ_p to be the time (in second) during which configuration p is active. Let f_ℓ^m denote the amount of traffic (bits) of session m passing through link ℓ. The capacity of a link (ℓ) depends on both the channel width and SINR threshold and is given by Eq.(4). Let $\omega^+(i)$ be the set of all outgoing links from node i and $\omega^-(i)$ be the set of all incoming links to node i, the problem can be modeled as:

$$\text{Objective: Minimize} \sum_{p=1}^{\bar{P}} \lambda_p \tag{10}$$

Subject to:

$$\sum_{\ell \in \omega^{+}(i):i \in N} f_{\ell}^{m} - \sum_{\ell \in \omega^{-}(i):i \in N} f_{\ell}^{m} = 0 \qquad \forall i \in N - \{t_m, r_m\} \qquad \forall m = 1..M \quad (11)$$

$$\sum_{\ell \in \omega^{+}(t_m):t_m \in N} f_{\ell}^{m} - \sum_{\ell \in \omega^{-}(t_m):t_m \in N} f_{\ell}^{m} = R_m \qquad \forall m = 1..M \quad (12)$$

$$\sum_{\ell \in \omega^{+}(r_m):r_m \in N} f_{\ell}^{m} - \sum_{\ell \in \omega^{-}(r_m):r_m \in N} f_{\ell}^{m} = -R_m \qquad \forall m = 1..M \quad (13)$$

$$\sum_{p=1}^{\bar{P}} \lambda_p \times b_\ell \times x_\ell^p \times \log_2(1+\beta) - \sum_{m=1}^{M} f_\ell^m \geq 0 \qquad \forall \ell \in E \qquad (14)$$

where $f_\ell^m \geq 0$,$\lambda_p \geq 0$. The objective function (Eq.10) asks to minimize the total system activation time to satisfy all the traffic demands. Equations (11-13) present the flow conservation constraints in the network. Eq.(14) indicates that the total traffic routed through link ℓ can not exceed the total transport capacity of link ℓ. Finding the solution of the above model relies on determining the set of all feasible configurations $\mathcal{P}$. According to the number of links and the number of possible channel widths, the size of $\mathcal{P}$ can be extremely large. This makes the above LP computationally infeasible, since it may not be possible to enumerate all such configurations. Additionally most of these configurations will not be used in the optimal solution. To reduce this complexity a more effective way to solve the problem is to use a primal-dual approach for decomposing the problem into subproblems [14] [10] [1].

4 Column Generation (CG) Model

Column generation is an optimization technique that decomposes the linear program (LP) into a master model and its dual the pricing model. The master model is initialized with a subset of columns P_0 (in our case, a transmission configuration, $P_0 \subseteq \mathcal{P}$) of the LP. The pricing is a separate model for the dual LP and is solved to identify whether the master should be enlarged with additional columns or not. Therefore, as opposed to an LP where all the columns are used at the same time to obtain the optimal solution, CG alternates between the master and the pricing model, until the former contains the necessary columns required to find the optimal solution of the original LP.
A. MASTER PROBLEM

$$\text{Objective: Minimize} \sum_{p \in P_0} \lambda_p \qquad (15)$$

Subject to: Equations (11-13)

$$\sum_{p \in P_0} \lambda_p \times b_\ell \times x_\ell^p \times \log_2(1+\beta) - \sum_{m=1}^{M} f_\ell^m \geq 0 \qquad \forall \ell \in E \qquad (16)$$

During every iteration, when the master problem is solved, we need to verify its optimality. If it is, then we conclude our search, or else decide a new column to join in its basis that can improve the current solution. This can be achieved by examining whether any new column that is not currently in P_0, has a negative reduced cost. Denoting the dual variables corresponding to Eq.(16) by u_ℓ, the reduced cost ($\overline{cost}$) for any new column can be expressed as:

$$\overline{cost} = 1 - log_2(1+\beta) \times \sum_{\ell \in E} u_\ell \times b_\ell \times x_\ell \tag{17}$$

B. PRICING PROBLEM
Objective: Minimize $\overline{cost}$
Subject to: Equations (7-8)

$$x_\ell + x_{\ell'} \leq 1 + f_{CR}(\ell, \ell') \qquad \forall (\ell, \ell') \in E \ \ell \neq \ell' \tag{18}$$

$$b_\ell x_\ell = e_\ell - s_\ell \qquad \forall \ell \in E \tag{19}$$

$$e_\ell \leq B \times x_\ell \qquad \forall \ell \in E \tag{20}$$

$e_\ell = \{1, 2, .., B\}$, $s_\ell = \{1, 2, .., B\}$, $x_\ell = \{0, 1\}$, $y_{\ell,\ell'} = \{0, 1\}$.

5 A Simulated Annealing (SA) Based Pricing Problem

5.1 Basic Idea

The master subproblem of the CG presented above deals with non-integer variables (f_ℓ^m, λ_p) and usually is very easy to solve (LP) even for larger networks. However, the pricing subproblem deals with integer variables ($e_\ell, s_\ell, x_\ell, y_{\ell,\ell'}$) which is commonly difficult to solve (ILP) and presents scalability problems when the number of links increases. Therefore, there still remain a complexity in obtaining a solution even after decomposing the problem. To circumvent the complexity issue, we propose a heuristic method based on SA, an efficient meta-heuristic technique, to solve the dual subproblem. SA tries to find better solutions (in our case configurations) by comparing the cost of current and candidate solution. The heuristic algorithm is explained by the following steps:

Step 1: Generate an initial configuration $p_{initial}$ (defined in Section 3) from a set of randomly selected links without violating the interference constraints.

Step 2: Assign current configuration $p_{cur} = p_{initial}$.

Step 3: Compute $\overline{cost}_{cur}$ of p_{cur} using Eq.(17).

Step 4: Set initial temperature $T = \overline{cost}_{cur}$ and the temperature reduction factor RF to some constant. Initialize the outer-loop (OL), the maximum number of temperature reduction and inner-loop (IL), the maximum number of iteration with a particular temperature.

Step 5: Repeat *Step 6* to *Step 10* for OL times.

Step 6: Repeat *Step 7* to *Step 9* for IL times.

Step 7: Generate a candidate solution, p_{can}, through the function ***Schedule*** (which will be introduced in Section 5.2) and compute $\overline{cost}_{can}$ using Eq.(17).

Step 8: If ($\overline{cost}_{can} \leq \overline{cost}_{cur}$), accept the candidate configuration and set $p_{cur} = p_{can}$, also $\overline{cost}_{cur} = \overline{cost}_{can}$.

Step 9: If ($\overline{cost}_{can} > \overline{cost}_{cur}$), accept the candidate configuration with a probability $Pr = e^{(\frac{\overline{cost}_{cur} - \overline{cost}_{can}}{T})}$ and set $p_{cur} = p_{can}$, $\overline{cost}_{cur} = \overline{cost}_{can}$.

Step 10: Reduce $T (T = T \times RF)$.

We terminate the process as soon as we obtain a negative reduced cost ($\overline{cost}_{can} < 0$) and add the configuration (p_{can}) as a new column into the basis of master problem. The alternation between master and pricing problem continues as long as the pricing problem provides a negative reduced cost.

5.2 Link Scheduling

The function ***Schedule***, introduced earlier, generates a candidate configuration and is described through Algorithm 1. Let F^{ℓ}_{CR} be a set that contains all links

Algorithm 1. Link scheduling

```
$F^{\ell}_{CR}$, $F^{\ell}_{CC}$, $A$: Set of active links, $E$: Set of all links.
$s_\ell$, $e_\ell$, $b_\ell$: identify the spectrum block location and size used on link $\ell$.
$BA_\ell$: Spectrum availability vector for link $\ell$
Schedule ($p_{cur}$)
Initialize $BA_\ell = \emptyset$        $(\ell \in E)$
Set $A$ from current configuration ($p_{cur}$)
Set $BA_\ell$ corresponding to the spectrum band (all fragments) of $\ell$, $\forall \ell \in A$
Select random $\ell : \ell \in A$
$A \leftarrow A - \{\ell\}$; $E \leftarrow E - \{\ell\}$
while $E \neq \emptyset$ do
    Select random $\ell' : \ell' \in E$
    $E \leftarrow E - \{\ell'\}$
    if $F^{\ell'}_{CR} \cap A = \emptyset$ then
        if $F^{\ell'}_{CC} \cap A = \emptyset$ then
            $A \leftarrow A + \{\ell'\}$
            Update the status of the slots $\{0 : b_{\ell'}\}$ to be busy (1) in $BA_{\ell'}$
        else
            $K \leftarrow K \bigvee BA_\ell$        $(\forall \ell \in F^{\ell'}_{CC}, \ell \in A)$
            $s_{\ell'} = F(K, b_{\ell'})$; $e_{\ell'} = s_{\ell'} + b_{\ell'}$
            if $s_{\ell'}$ is valid then
                $A \leftarrow A + \{\ell'\}$
                Update the status of the slots $\{s_{\ell'} : e_{\ell'}\}$ to be busy (1) in $BA_{\ell'}$
            end if
        end if
    end if
end while
Set candidate configuration $p_{can}$ from $A$
```

in the network which share a common radio with link ℓ. Let F_{CC}^{ℓ} be a set that contains all links in the network which fall inside the interference range of ℓ. Let A be a set of active links ℓ where $x_\ell^p = 1$ in current configuration, (p_{cur}). To construct a new candidate (neighboring) configuration, we remove at random a link ℓ (or more) from A and search for links which may be added to the current configuration (and do not violate both interference and radio constraints). For example, after removing a link, another link ℓ' will be added to A if for every existing link $\ell \in A$, $\ell \notin F_{CR}^{\ell'}$ and $\ell \notin F_{CC}^{\ell'}$. If $\ell \in F_{CC}^{\ell'}$, then ℓ' will be added to A given that we can find a spectrum block to be assigned on ℓ' and which does not overlap with that of ℓ. Otherwise, ℓ' cannot be added.To facilitate the search for such non overlapping spectrum block, we use a slotted presentation of the permissible spectrum band. For each link ℓ, let BA_ℓ denote the set of contiguous spectrum slots or fragments, of unit length each, and of total length B(MHz). Let t_i^ℓ $(1 \leq i \leq B)$ be a binary variable indicating whether spectrum fragment i is assigned $(t_i^\ell = 1)$ to link ℓ or not $(t_i^\ell = 0)$. Thus, BA_ℓ is a bitset of length B for each ℓ in the network and is configured when ℓ is added to A. That is, BA_ℓ indicates the location of spectrum block b_ℓ in the permissible spectrum. Next, we illustrate the spectrum block assignment on link ℓ' $(\ell \in F_{CC}^{\ell'})$ that may be added to A. We perform a *bitwise OR* operation on all bitsets $BA_\ell, \forall \ell \in F_{CC}^{\ell'}$ and determine the resultant binary vector $K : K = \bigvee BA_\ell, \forall \ell \in F_{CC}^{\ell'}, \ell \in A$. We search through K for available contiguous spectrum fragments of size $b_{\ell'}$ to determine $s_{\ell'}$ and thus allocate the spectrum block to the link and add ℓ' to A. We use a first fit allocation where the first spectrum block available is allocated to ℓ'. Otherwise, if a block could not be found, then ℓ' is not added to A. This method is shown in Line 19 and makes use of the string matching algorithm of Knuth-Morris-Pratt (KMP) [9] to find the start frequency of the allocated spectrum spectrum block. This procedure is repeated for all links in the network, except those in A. Finally, we obtain a candidate configuration $p_{can} = A$.

6 Numerical Results

We consider randomly generated networks (5, 10, 20, 30 and 40 nodes) that are deployed over a 100m×100m area. The transmission power is assumed to be constant (P_{max}=1mW). The power spectral density of thermal noise is fixed to N_0=10^{-6}W/MHz. We assume a simple propagation model and a path loss exponent α=2; we calculate the transmission range (T_k) using Eq.(2) and the interference range (IR_k) from T_k considering $\delta = 0.2$. The SINR threshold is chosen as β=1.3. We consider a total system spectrum B=80Mhz, from 0MHz to 80MHz. We refer to our methods as PI_v and PH_v, where in the former the dual sub-problem is solved by ILP and in the second using SA. Both PI_v and PH_v consider optimal allocation of non-uniform spectrum bands and they are compared against a method (PI_f) with optimal but fixed spectrum allocation; in PI_f, the dual sub-problem is solved as an ILP. We obtain results for 5,10,20,40Mhz fixed spectrum blocks and present only the results of the one showing best performance, referred to as PI_{fb}. For larger network (e.g., 40

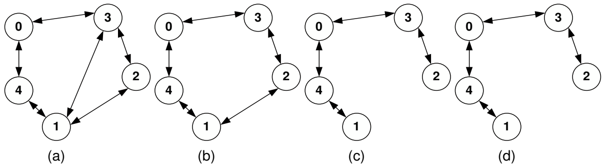


Fig. 3. A 5-node network, channel width of (a)5MHz (b)10MHz (c)20MHz (d)40MHz

nodes), we also present results obtained from a PH_{fb} design method (best fixed spectrum band allocation model where dual sub-problem is solved using SA). Our performance metrics are the schedule length (system activation time) and computation time (CPU time). The CG method is implemented in C++ and solved using CPLEX Concert Technology.

We first consider the small network shown in Fig.3. Fig.3(a)-(d) shows the different spectrum graphs, each representing one particular bandwidth (5MHz - 40MHz). We consider 3 sessions: (3,1), (0,3) and (1,2), with random traffic demands of (27.4085), (6.9143) and (9.72211)Mbits. Table 1 compares the performance of the best fixed spectrum band allocation PI_{fb} (in this case 40MHz) with variable-width spectrum allocation (PI_v and PH_v) in terms of optimal routing. When the spectrum width is fixed to 40MHz (PI_{fb}), the transmission range is shorter and the network becomes sparser (Fig.3(d)), and thus all the sessions will have single routes to deliver their demands. In contrast to that, both PI_v and PH_v explored other routing options (by searching through all the spectrum graphs of Fig.3) and used multi-path routing to deliver session 3.

Table 2 compares the system activation time of PI_{fb}, PI_v and PH_v and present the spectrum assignment for active links in each configuration. PI_v and PH_v show identical results and significantly outperform PI_{fb}; both PI_v and PH_v needed 1.33181(s) to deliver the demands; this is an improvement of 21.14% over PI_{fb}. Variable spectrum allocations *flexibly* select a mixture of small and large spectrum bands to increase concurrency among adjacent links while keeping interference under check and hence these methods achieve minimum system activation time to satisfy the traffic demand. In Table 2, it is shown that both 10MHz and 40MHz widths are selected in the optimal solution (for PI_v and PH_v). Here, 1.14142Mbits of session 3 are routed over a multi-hop path of larger transport capacity links (corresponds to 40MHz), while 8.58069Mbits of the session are routed over a single hop path (1-2) with 10MHz spectrum band.

Table 1. Results in a 5-node Network with 3 sessions

Session	PI_{fb}		PI_v		PH_v	
	Path	Traffic(Mbits)	Path	Traffic(Mb)	Path	Traffic(Mb)
1	(3,0,4,1)	27.4085	(3,0,4,1)	27.4085	(3,0, 4,1)	27.4085
2	(0,3)	6.9143	(0,3)	6.9143	(0,3)	6.9143
3	(1,4,0,3,2)	9.72211	(1,2)	8.58069	(1,2)	8.58069
			(1,4,0,3,2)	1.14142	(1,4,0,3,2)	1.14142

Table 2. Configurations for 5-node network: PI_{fb} (width= 40MHz),PI_v and PH_v

	Configurations	Active Links	s_ℓ-e_ℓ(MHz)	Active Links	s_ℓ-e_ℓ	Time(s)
PI_{fb}	1	(0,3)	0-40	—	—	0.143852
	2	(0,4)	0-40	—	—	0.367965
	3	(4,0)	0-40	—	—	0.202269
	4	(1,4)	0-40	(3,0)	0-40	0.202269
	5	(0,3)	0-40	(4,1)	0-40	0.202269
	6	(3,0)	0-40	(4,1)	40-80	0.367965
	7	(0,4)	0-40	(3,2)	0-40	0.202269
	Total System Activation Time					1.68886
PI_v	1	(1,2)	0-10	(0,3)	0-40	0.143852
	2	(4, 1)	0-40	(3,0)	40-80	0.570233
	3	(1,2)	0-10	(0,4)	10-50	0.570233
	4	(4, 0)	0-40	(3,2)	40-80	0.0237473
	5	(1,4)	0-40	(0,3)	40-80	0.0237473
	Total System Activation Time					1.33181
PH_v	1	(1,2)	0-10	(0,3)	0-40	0.143852
	2	(4, 1)	0-40	(3,0)	40-80	0.570233
	3	(1,2)	0-10	(0,4)	10-50	0.570233
	4	(4, 0)	40-80	(3,2)	0-40	0.0237473
	5	(1,4)	40-80	(0,3)	0-40	0.0237473
	Total System Activation Time					1.33181

This shows that *a mixture of large capacity short links with high concurrency and smaller capacity longer links are selected in the optimal solution.* Note from Table 2 that for PI_{fb} both links (1-4) and (3-0) are concurrently active and allocated the same spectrum band, since these links are outside the interference ranges of each other (Fig.3). However, we observe in configuration 6 that links (3-0) and (4-1) are concurrently active but allocated non-overlapping spectrum because node 4 (transmitter) falls inside the interference range of node 0 (as shown in Fig.3(d)). Next, we present numerical results of the different allocation schemes for different networks (5,10,20,30,40 nodes) and traffic instances (3-40 sessions). Each session m has a traffic demand randomly generated in the range of 0< R_m ≤35Mbits. Depending on the size of the network and traffic demand we used values of the outer-loop (OL) and inner-loop (IL) parameters of PH_v from 40 to 250. Cooling factor is assumed to be constant ($RF = 0.98$). The results are presented in Table 3 and 4, excluding the results of PI_v which were computational infeasible to get. We observe that the performance of PH_v is very close to that of PI_v with worst optimality gap (Opt. Gap) not exceeding 2.8% and most often PH_v reaches the same optimal solution of PI_v. Optimal fixed spectrum allocation results in poor performance, under-performing PH_v and PI_v by up to 44% for larger network (Table 4). This is due to the more effective and flexible spectrum partitioning obtained by variable-width band allocation; the spectrum is partitioned to yield enough spectrum blocks with different widths which promote transmission concurrency and control interference

Table 3. System activation time and CPU time for different networks, traffic demands

	Sys. Activation Time (sec)			CPU Time (sec)		PI_v vs. PH_v
Nodes_Sessions	PI_{fb}	PI_v	PH_v	PI_v	PH_v	Opt. Gap(%)
05_03	1.68886	1.33181	1.33181	3.47	1.04	0
05_10	4.22263	3.60719	3.60719	3.23	1.32	0
05_20	8.71577	7.76042	7.76042	4.36	1.43	0
05_30	14.3912	12.3543	12.3543	3.60	1.68	0
05_40	19.9229	16.9236	16.9236	3.70	1.91	0
10_03	1.28432	1.05623	1.05623	90.81	14.47	0
10_10	2.90966	2.30186	2.30583	247.02	111.11	0.17
10_20	8.83257	5.80744	5.80744	133.84	29.45	0
10_30	11.6592	8.44628	8.44931	158.82	72.74	0.03
10_40	16.414	11.6761	11.6761	138.16	73.66	0
20_03	0.800981	0.772502	0.772502	28398.91	760.96	0
20_10	2.58964	1.49001	1.49001	31334.04	1323.68	0
20_20	3.96756	2.44656	2.51543	321301.72	20936.9	2.8
20_30	6.02412	3.71392	3.71392	204278.22	47320.51	0
20_40	7.97393	4.69807	4.75531	218852.72	33593.2	1.2
30_03	0.772502	0.772502	0.772502	217716.64	1623.68	0
30_10	1.30669	1.30669	1.30669	232162.38	10870.26	0
30_20	2.11141	2.04858	2.05405	272508.00	20215	0.26
30_30	3.1206	—	3.0533	—	47727.4	—
30_40	3.78369	—	3.68477	—	46873.5	—

Table 4. System activation time and CPU time for a 40-node network

	Sys. Activation Time (sec)			CPU Time (sec)		PI_{fb}/PH_{fb}	PI_{fb}/PH_v
Nodes_Sessions	PI_{fb}	PH_{fb}	PH_v	PI_{fb}	PH_{fb}	Opt. Gap(%)	Improvement (%)
40_05	1.60925	1.60925	0.906884	18.52	2.60	0	44
40_10	1.70376	1.70376	1.25651	69.45	11.83	0	26
40_20	4.1462	4.1462	2.42652	21.35	7.21	0	41
40_30	4.1462	4.1462	2.78739	59.41	13.61	0	32
40_40	4.81622	4.81622	3.19487	100.74	14.85	0	33

among active neighboring links. These same performance characteristics cannot be obtained using fixed-width spectrum blocks due to the lack of flexibility in the spectrum allocation. Table 4 compares the performance of PI_{fb}, PH_{fb} and PH_v, with PH_v outperforming the other two and PI_{fb} and PH_{fb} achieving similar performance. The table shows that PH_{fb} is much more scalable than the other methods. In Table 3 we compare the CPU time of PI_v and PH_v; we observe that PH_v substantially outperforms PI_v where the CPU time for the former being 2× less for smaller networks (5 and 10 nodes) and up to 134× less for larger networks (30-nodes). For a 40-node network, PI_v did not return any solution after more than 15-days of runtime.

7 Conclusion

We studied the problem of optimal spectrum allocation in frequency-agile wireless networks. We showed that although narrower spectrum bands can split the permissible spectrum into more non-overlapping channels each with larger transmission range, such narrow bands will result in small capacity links and limit the spatial reuse. Wider spectrums however result in larger capacity links and better spatial reuse, but also results in smaller transmission range, yielding more hops along end-to-end routes and hence exacerbate the intra-path interference. Our joint design problem strikes a balance among these conflicting objectives. We formulated the problem as an optimization problem and adopted a primal-dual decomposition for solving it. We also proposed a heuristic approach based on SA to solve the dual sub-problem and numerical results revealed that SA significantly improves the computation time and achieves near optimal solution.

References

1. Bjorklund, P., Varbrand, P., Yuan, D.: Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach. In: Proc. IEEE INFOCOM 2003, vol. 2, pp. 818–824 (2003)
2. Cao, L., Yang, L., Zheng, H.: The impact of frequency agility on dynamic spectrum sharing. In: Proc. IEEE DySpan (to appear 2010)
3. Chandra, R., Mahajan, R., Moscibroda, T., Raghavendra, R., Bahl, P.: A case for adapting channel width in wireless networks. In: Proc. ACM SIGCOMM 2008, pp. 135–146 (2008)
4. Feng, Z., Yang, Y.: Joint transport, routing and spectrum sharing optimization for wireless networks with frequency-agile radios. In: Proc. IEEE Infocom 2009 (2009)
5. Gollakota, S., Perli, S.D., Katabi, D.: Interference alignment and cancellation. In: Proc. of SIGCOMM, Barcelona, Spain, pp. 159–170 (2009)
6. Gummadi, R., Patra, R., Balakrishnan, H., Brewer, E.: Interference avoidance and control. In: Proc. 7th ACM Hotnets-VII (2008)
7. Halperin, D., Ammer, J., Anderson, T., Wetherall, D.: Interference cancellation: Better receivers for a new wireless mac. In: Proc. of HotNets (2007)
8. Hou, Y.T., Shi, Y., Sherali, H.: Optimal spectrum sharing for multihop software defined radio networks. In: Proc. of Infocom, Alaska, pp. 1–9 (May 2007)
9. Knuth, D.E., Morris Jr., J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM Journal on Computing 6(2), 323–350 (1977)
10. Kompella, S., Wieselthier, J.E., Ephremides, A., Sherali, H.D.: A cross-layer approach to end-to-end routing and sinr-based scheduling in multi-hop wireless networks. In: Proc. WiOPT 2008, pp. 261–266 (2008)
11. Mishra, A., Shrivastava, V., Banerjee, S., Arbaugh, W.A.: Partially overlapped channels not considered harmful. In: Proc. SIGMETRICS, pp. 63–74 (2006)
12. Moscibroda, T., Chandra, R., Wu, Y., Sengupta, S., Bahl, P., Yuan, Y.: Load-aware spectrum distribution in wireless LANs. In: Proc. IEEE ICNP 2008, pp. 137–146 (2008)
13. Shu, T., Krunz, M.: Coordinated channel access in cognitive radio networks: A multi-level spectrum opportunity perspective. In: Proc. IEEE Infocom 2009 (2009)
14. Zhang, J., Wu, H., Zhang, Q., Li, B.: Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In: Proc. 2nd International Conference on Broadband Networks, pp. 631–640 (2005)

Towards Modeling Realistic Mobility for Performance Evaluations in MANET

Alex Aravind* and Hassan Tahir

Computer Science Program
University of Northern British Columbia
Prince George, BC, Canada - V2N 4Z9
{csalex,tahir}@unbc.ca
http://www.unbc.ca/

Abstract. Simulation modeling plays crucial role in conducting research on complex dynamic systems like mobile ad hoc networks and often the only way. Simulation has been successfully applied in MANET for more than two decades. In several recent studies, it is observed that the credibility of the simulation results in the field has decreased while the use of simulation has steadily increased. Part of this credibility crisis has been attributed to the simulation of mobility of the nodes in the system. Mobility has such a fundamental influence on the behavior and performance of mobile ad hoc networks. Accurate modeling and knowledge of mobility of the nodes in the system is not only helpful but also essential for the understanding and interpretation of the performance of the system under study. Several ideas, mostly in isolation, have been proposed in the literature to infuse realism in the mobility of nodes. In this paper, we attempt a holistic analysis of creating realistic mobility models and then demonstrate creation and analysis of realistic mobility models using a software tool we have developed. Using our software tool, desired mobility of the nodes in the system can be specified, generated, analyzed, and then the trace can be exported to be used in the performance studies of proposed algorithms or systems.

Keywords: Mobile computing, mobility model, realistic mobility, trace generation.

1 Introduction

Mobile ad hoc networking (MANET) has been an active field of research in the last few decades. MANETs encompass a wide range of dynamic systems which are typically complex. Due to their versatility, dynamic nature, and inherent complexity, it is difficult to characterize them analytically or conduct experiment using mobile nodes and wireless networks in realistic scenarios. Therefore, nearly all published research works in this field are heavily based on simulations [3,9,14,17]. However, a number of studies, particularly in recent times,

* Corresponding Author.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 109–122, 2010.
© Springer-Verlag Berlin Heidelberg 2010

have criticized the lack of rigor in simulation studies in this field and questioned the credibility of the published claims [3,9,14,17,26,27]. Specifically, these studies indicate that the credibility of the simulation results in the field has decreased while the use of simulation has steadily increased. Part of this credibility crisis has been attributed to the simulation of mobility of the nodes in the system. Mobility is one of the important characteristics of MANETs. It has fundamental influence on the behavior and performance of the system. Although several recent studies have expressed concern related rigor of simulation studies, there is no consensus on how to proceed to address the concern. We are interested in looking at the recommendations and criticisms and see how they could impact in deciding suitable mobility models for simulation studies in MANET.

1.1 Simulation Purpose and MANET Types

The opinion on the purpose of simulation varies. The authors in [9] argue that it is to provide a proof of concept and general performance characteristics. On the other hand, most critics on simulation studies emphasize thorough simulation with accurate modeling of the system. That is, the opinion on the purpose of simulation ranges from a very simple model with basic simulation for proof of concept to sophisticated modeling with thorough simulation to understand the system close to its real implementation. In [3], Stojmenovic discusses these differences in detail and summarizes the existing and new recommendations on simulation practices. From this analysis, it is clear that there is a no consensus on the purpose of simulation.

Mobile ad hoc network encompasses a wide range of systems. The network deployment region and the intended application are key factors in creating such variations. The region could be a simple regular open space without any obstacles and guiding structures, a city area with roads, building, etc. with intricate details, or anything in between these two extremes.

The purpose of the simulation and the types of system intended mainly attribute to the level of accuracy needed in modeling and simulation. Accuracy in modeling and simulation often relates to bringing realism into the model. When such modeling would be desirable, one could argue that accurate realistic modeling is generally a very challenging task and investing huge effort to achieve such realism may not even be necessary or feasible for all studies.

1.2 Where Do We Stand?

Simulation with a very simple model and simulation with high realism are two extreme cases. Between simple and sophisticated modeling, there is a spectrum of simulation models in which several useful models can be chosen with varying levels of details. We feel that the purpose and contribution of the research should dictate the level of details.

Simulation is a powerful tool and often it is the only effective tool available to study complex systems. Restricting the ways and the purposes of conducting simulations into a few fixed levels, irrespective of the system and intent, has the potential of avoiding useful simulation studies. We feel that simulation studies must be recognized in various levels according to the purpose, and the results must be evaluated and interpreted based on the rigor in which the system was modeled and simulated. This brings the motivation for our contribution in this paper.

1.3 Motivation

Simulation of MANET usually involves modeling its various components, and that often includes modeling mobility of the nodes in the system. Such an integral role of mobility in MANET motivates a number of questions. Is there a mobility model which is generally acceptable for all simulation studies in MANET? If so, what are the attractive properties of that mobility model? The variations in the purpose of simulation and type of MANET indicate that there cannot be a single mobility model suitable for all simulation studies in MANET. If that is the case, then the next question is why are a few mobility models heavily used so far in most simulation studies in MANET? How does the chosen mobility model, from a few available choices, fulfill the objective of the study?

Not all mobility models are easy to implement from the scratch. Some, particularly involving realism, are very challenging to implement. In such cases, how to ease this burden? What are the bases for realism?

These are some of the questions that set basis for our contribution in this paper.

1.4 Contributions

After presenting a simple classification of mobility models, we characterize a unified framework for important geometric objects in the simulation region that has potential to influence the mobility of the nodes in the system. Then, after analyzing several ways of generating realistic mobility models, we present a new software tool to generate realistic mobility models based on the geometry extracted from a generic universal GIS database. Using the proposed software tool, users can generate, analyze, and adjust various scenarios of the specified mobility model. By suitably controlling the parameters, various mobility models with different level of realism can be generated and visualized. Then, after satisfactory analysis, the trace can be exported to be used in the simulation studies. In the end, we discuss the related works.

2 A Classification

Though several attempts have been made in the literature to classify mobility models, the distinction between random mobility and realistic mobility is not clear and the boundary between them is often fuzzy. For example, many mobility models that just avoid sharp turns and sudden speed change are referred as realistic mobility models in the literature[13,16,18,24], although they are heavily influenced by randomness.

We propose a different classification to make the distinction clear. We classify the mobility models into two classes: *unguided mobility* and *guided mobility.* In unguided scheme, the mobility is mainly governed by the characteristics of the node. In guided scheme, the mobility of a node is governed by the characteristics of both the node as well as the geometry of the region. Primary geometry includes mainly the transportation paths such as roads, railways, ferry routes etc. Many additional geometrical elements such as malls, gas stations, hotels, etc. can be introduced into the region to create more realism.

Although random mobility is very useful for many research studies, mobility of living creatures do not follow a complete random mobility pattern. Often they are guided by many factors such as infrastructure (roads, malls, trails, playgrounds, mountains, etc.), profession or role (students, bus drivers, factory workers, flight attendants, etc.), activities, time, etc. That is, mobile nodes in real life often follow constrained and predictable mobility patterns. Similarly, in simulations, the mobility of node may be bounded by streets, freeways, obstacles or buildings.

3 Obstacles, Attraction Points, and Repulsion Points

Generally, regions of interest have objects that could influence the mobility of the nodes in the system. These objects are broadly identified in the literature as *obstacles* that the mobile nodes cannot pass through, *attraction points* that the mobile nodes might attracted to, and *repulsion points* that the mobile nodes would want to avoid going near to. To bring all these objects into a generic framework, we refer them collectively as **force points** and the force between an individual class of nodes and a force point can be defined using suitable force function. Let FP be the set of all force points and N be the set of different classes of mobile nodes. Then, the domain of the inter-force function FF is $FP \times N$ (Cartesian product of FP and N) and the range is R (real numbers). For any class of mobile nodes, the force point is attraction point if the value of inter-force falls in the range $[-\infty, 0)$. Similarly, the force point is repulsion point if the value of inter-force falls in the range $(0, \infty]$. The force point is obstacle, when the force is 0.

These force points could be stationary or mobile. For example, ice cream trucks, ambulance, trucks with inflammable materials can be considered as mobile force points. Parks, theater, museum, etc., can be considered as stationary force points. So each force point may be attached with a speed parameter. The value of this parameter may vary with time. Now, we have three parameters related to each force point - speed, time, and inter-force. If we consider these three parameters in three orthogonal dimensions, each force point will have a unique position in this space. This unified characterization of obstacles, attraction points, and repulsion points, along with their mobile characteristics as force points in a common framework may be useful to model and study the characterization of mobile nodes in MANET.

4 Generation of Realistic Mobility Models

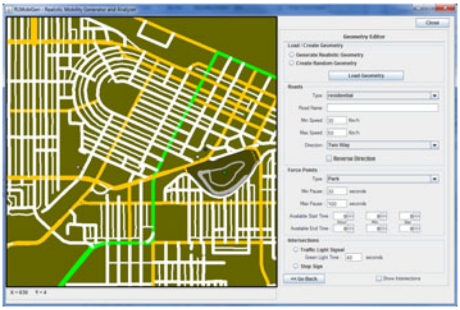
There are several approaches proposed in the literature to introduce realism into mobility models. They may be classified into the following three categories.

- Introducing realism in random mobility models: (i) making turns, directions, speeds, and pauses in constrained ways; (ii) introducing roads, highways, obstacles, attraction and repulsion points randomly or based on some graphs in the simulation region; (iii) introducing group mobility based on social networks. These classes of models are often referred as *realistic mobility models*.
- Mobility models based on geographic information system (GIS).
- Mobility models derived directly from real traces.

In these, models derived from real traces would give highly realistic mobility model. But, the availability of such traces are rare and even available historical data are remotely useful as they seldom reflect the future pattern. Random mobility models are synthetic and therefore they may not reflect mobility of a practical system. However, it was considered reasonable to use such models, and particularly one of the random mobility models called random waypoint model is expected to be remain used as the benchmark mobility model in MANET simulations[15]. Random waypoint model is simple and supported by mathematical foundations. It will remain natural choice for most proof of concept type simulations. However, for simulation of more realistic MANETs, we believe that the rate of use of random mobility models in MANET simulations would decrease drastically if better alternates are available. Practical systems will rarely have nodes with random dominated mobilities. Therefore, we consider mobility models based on GIS would be more desirable in designing realistic MANET simulations.

Next we elaborate what we mean by mobility models based on GIS. In this scheme, the mobility regions are expected to contain representation of real life artifacts such as roads, malls, parks, trails, etc. This is important because most new systems will be built either in an existing real region or in an existing real region with additional artifacts. Very few specialized projects will be deployed on a completely plain region. Our objective is to develop a software to reflect these two scenarios and create mobility models based on the obtained realistic geometry from GIS data. GEMM, Obstacle Mobility Model, MobiREAL, and RMobiSim proposed for MANET may be considered as related works. These softwares neither extract geometry from GIS nor as comprehensive as ours. We encountered several challenges in generating geometry from GIS data.

Our next task was to find a suitable GIS. Google Earth is good choice but it is not free. There are free databases from where real time data can be extracted (e.g. GeoBase [33], Tiger Database [40], OpenStreetMap [38]). Tiger database only provides the geographic data of United States and GeoBase only provides the geographic data of Canada. Therefore, we have used OpenStreetMap (OSM) which provides the geographic data of the whole world and is freely available over



Load / Create Geometry
○ Generate Realistic Geometry
○ Create Random Geometry

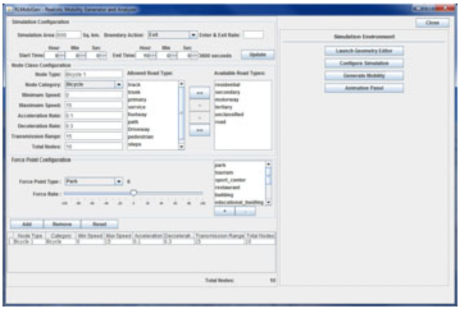
Roads
Type: residential
Road Name:
Min Speed: 30 Km/h
Max Speed: 50 Km/h
Direction: Two Way
□ Reverse Direction

Force Points
Type: Park
Min Pause: 30 seconds
Max Pause: 100 seconds
Available Start Time: 0 0 0
Available End Time: 0 0 0

Intersections
○ Traffic Light Signal
Green Light Time: 40 seconds
○ Stop Sign
□ Show Intersections

X = 630 Y = 4

Fig. 1. Geometry Editor



Simulation Configuration
Simulation Area: 2500 Sq. km. Boundary Action: Exit Enter & Exit Rate:
Start Time: Hour 0 Min 0 Sec 0 End Time: Hour 1 Min 0 Sec 0 3600 seconds

Node Class Configuration
Node Type: Bicycle 1
Node Category: Bicycle
Minimum Speed: 0
Maximum Speed: 15
Acceleration Rate: 0.1
Deceleration Rate: 0.3
Transmission Range: 15
Total Nodes: 10

Allowed Road Type: track, trunk, primary, service, footway, path, Driveway, pedestrian, steps
Available Road Types: residential, secondary, motorway, tertiary, unclassified, road

Force Point Configuration
Force Point Type: Park
Force Rate:
park, tourism, sport_center, restaurant, building, educational_building

Node Type	Category	Min Speed	Max Speed	Acceleration	Decelerat...	Transmission Range	Total Nodes
Bicycle 1	Bicycle	0	15	0.1	0.3	15	10

Total Nodes: 10

Fig. 2. Simulation Configuration

Internet and can be used by anyone for both commercial and non-commercial purposes. Thus, our software uses OSM to extract geometry.

5 A Realistic Mobility Generation Software

To generate, analyze, and adjust various scenarios of the mobility model, we designed a software tool named as RLMobiGen. By suitably controlling the parameters, various mobility models with different level of realism can be generated and visualized. Then, after satisfactory analysis, the trace can be exported to be used in the simulation studies. The proof of concept of this tool was presented in [6], and now it is in the final phase of completion and will be ready for public use in the near future. In addition to its rich functionality, RLMobiGen is very user friendly and has nice graphical user interface. Its GUI is organized as hierarchal panels. User inputs the specifications at different component levels.

The software has the following five main logical components.

- **Simulation Initializer:** Simulation initializer (SI) is responsible for generating the basic simulation environment. It initializes the simulation with four parameters: simulation area, simulation start time, simulation end time and boundary action.
- **Geometry Generator:** We have developed a separate GUI for RLMobiGen geometry generator. Users can export the OSM XML Data from the OSM website and specify the OSM file through the GUI. This tool automatically parses the data from the file and generates a street graph. Fig. 1 illustrates the downtown area of Prince George, BC, Canada. We have classified the data into four different classes: roads, force points (offices, parks, etc), intersections (i.e. road intersects other road) and connecting points (i.e. road intersects the simulation boundary).

  An *intersection* is a road junction where two or more roads either meet or cross at the same level. It has following two attributes: *Traffic Lights:* refers to red, yellow and green signal, where node stops at red light and moves on green light. User can mention the time duration of green light for individual intersections. System simulates the traffic light for each individual road of the intersection using round-robin scheduling algorithm. *Stop Sign:* refers to the point where node comes to complete stop and then proceed towards its desired destination only if the way ahead is clear.

A *connecting point* is a point on the road where a road crosses the simulation boundary region. In realistic mobility scenario a node may enter in and exits from a simulation region. In a trip, Node may choose either one of the attraction points or a connecting point as its destination and proceeds towards it. System collects all the possible connecting points based on the road traffic flow (i.e. unidirectional or bidirectional).

– **Mobility Generator:** Mobility generator (MG) is responsible for creating nodes and generating their mobility trace. The trace also contains information required for visualization and statistical metrics computations based on the parameters set in SI, GG and MG. User sets the parameters in MG for node's carrier and node as shown in Fig. 2.

Our model provides basic node categories such as bicycle, bus, car, motorcycle, pedestrian and taxi. From these, the user can specify several node classes suitable for the intended simulation. Each node is attached with type, minimum speed, maximum speed, acceleration rate, deccelaration rate, allowed road type, transmission range, a set of force points.

Another interesting issue in generating mobility trace with a bounded region is what happens when a node hits the boundary (i.e. connecting points). This can be handled in many ways. We implement three simple approaches and we refer them as exit, replace, and enter-and-exit.

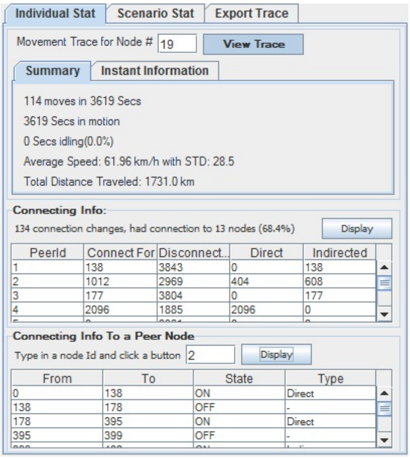
1. *Exit:* The node instantaneously disappears when it hits the boundary. In this model, the number of nodes in the system will decrease as system progresses.
2. *Replace:* When a node hits the boundary, it reappears from another random location on any of the road in the simulation region. In this scheme, the number of nodes remains the same throughout the simulation.
3. *Enter-and-Exit:* The node instantaneously disappears when it hits the boundary. Also, nodes appear from the boundary randomly. In this model, the number of nodes in the system will vary according to enter and exit rate set by the user.

Mobility Scenario Manager: This component is responsible for extracting various statistical insights and providing visualizations.

Mobility Trace Exporter: RLMobiGen also allows user to export the mobility trace into GloMoSim/QualNet, NS-2, NAM, JPG, PDF and XML format, as illustrated in Fig. 5.

6 Performance Metrics

Performance metrics used to analyze mobility models fall mainly into the following categories.



Movement Trace for Node # 19

114 moves in 3619 Secs

3619 Secs in motion

0 Secs idling(0.0%)

Average Speed: 61.96 km/h with STD: 28.5

Total Distance Traveled: 1731.0 km

Connecting Info:

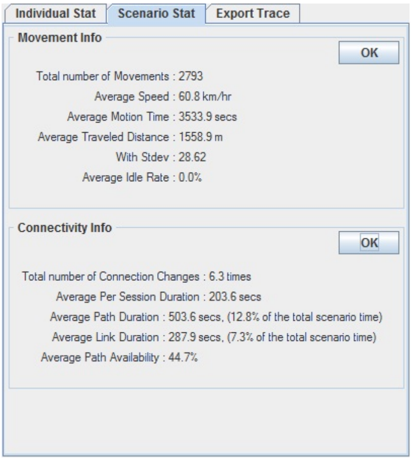
134 connection changes, had connection to 13 nodes (68.4%)

PeerId	Connect For	Disconnect...	Direct	Indirected
1	138	3843	0	138
2	1012	2969	404	608
3	177	3804	0	177
4	2096	1885	2096	0

Connecting Info To a Peer Node

Type in a node Id and click a button 2

From	To	State	Type
0	138	ON	Direct
138	178	OFF	-
178	395	ON	Direct
395	399	OFF	-



Movement Info

Total number of Movements : 2793

Average Speed : 60.8 km/hr

Average Motion Time : 3533.9 secs

Average Traveled Distance : 1558.9 m

With Stdev : 28.62

Average Idle Rate : 0.0%

Connectivity Info

Total number of Connection Changes : 6.3 times

Average Per Session Duration : 203.6 secs

Average Path Duration : 503.6 secs, (12.8% of the total scenario time)

Average Link Duration : 287.9 secs, (7.3% of the total scenario time)

Average Path Availability : 44.7%

Fig. 3. Individual Statistics

Fig. 4. Scenario Statistics

1. *Movement metrics:* number of legs, leg distance, leg speed, leg duration, etc.
2. *Connectivity metrics:* number of connections, connection duration, connection changes, connection availability, etc.
3. *Coverage metrics:* node distribution, coverage, etc.

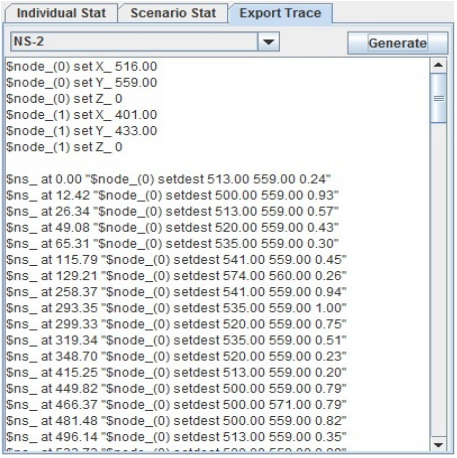
These metrices are extracted from the mobility trace by the *MetricsGenerator* using time-step based technique to collect statistics.

The above metrics, if applicable and meaningful, are computed for minimum, maximum, average, total, standard deviation, rate, ratio, etc., and also at individual, group, or system level as shown in Fig. 3 and Fig. 4. For example, connectivity may be analyzed between two nodes, between one and a group of nodes, or among all nodes. Movement metrics are easy to understand and straightforward to compute. Connectivity and coverage metrics require some elucidation.

Communication is a fundamental problem for most applications in any networked system and achieving effective communication between the mobile nodes is challenging due to the dynamics involved in the network. In mobile networks, the nodes with transmission range form a dynamic graph called *connectivity graph*. The performance of most of the communication protocols in this context is heavily influenced by the connectivity and the coverage of this graph.

6.1 Coverage Analysis

Coverage is influenced by both mobility and transmission range of the nodes. Node distribution and the ratio of the simulation area covered by transmission range to the total area are useful metrics to be analyzed for coverage. Since RLMobiGen is a discrete time based simulator, the metrics are computed over



```
$node_(0) set X_ 516.00
$node_(0) set Y_ 559.00
$node_(0) set Z_ 0
$node_(1) set X_ 401.00
$node_(1) set Y_ 433.00
$node_(1) set Z_ 0

$ns_ at 0.00 "$node_(0) setdest 513.00 559.00 0.24"
$ns_ at 12.42 "$node_(0) setdest 500.00 559.00 0.93"
$ns_ at 26.34 "$node_(0) setdest 513.00 559.00 0.57"
$ns_ at 49.08 "$node_(0) setdest 520.00 559.00 0.43"
$ns_ at 65.31 "$node_(0) setdest 535.00 559.00 0.30"
$ns_ at 115.79 "$node_(0) setdest 541.00 559.00 0.45"
$ns_ at 129.21 "$node_(0) setdest 574.00 560.00 0.26"
$ns_ at 258.37 "$node_(0) setdest 541.00 559.00 0.94"
$ns_ at 293.35 "$node_(0) setdest 535.00 559.00 1.00"
$ns_ at 299.33 "$node_(0) setdest 520.00 559.00 0.75"
$ns_ at 319.34 "$node_(0) setdest 535.00 559.00 0.51"
$ns_ at 348.70 "$node_(0) setdest 520.00 559.00 0.23"
$ns_ at 415.25 "$node_(0) setdest 513.00 559.00 0.20"
$ns_ at 449.82 "$node_(0) setdest 500.00 559.00 0.79"
$ns_ at 466.37 "$node_(0) setdest 500.00 571.00 0.79"
$ns_ at 481.48 "$node_(0) setdest 500.00 559.00 0.82"
$ns_ at 496.14 "$node_(0) setdest 513.00 559.00 0.35"
```

Fig. 5. NS2 Format

discrete times. Here the objective is to study how the nodes spread in the simulation region during simulation. To compute the node spread, the region is divided into small cells of equal size (10 by 10) and the nodes are counted inside each cell. Initially, 1000 nodes are uniformly placed inside the region. After the nodes move for 2000 seconds, a snapshot of the positions of the nodes is taken and the number of nodes at each cell is counted for their spatial distribution. Transmission range can vary from simulation to simulation.

6.2 Connectivity Analysis

For connectivity analysis, we mainly study two metrics: the connection changes and the session duration. We formally define these terms next.

- A **link** is said to exist or be ON between two nodes i and j if they are within each other's transmission range. Link can be uni-directional or bi-directional based on the transmission range of two nodes. Link is a function over time t and it is denoted by $link(i, j, t)$.
- A **path** is said to exist or be ON between two nodes i and j if there is a sequence of nodes and the links between consecutive nodes in the sequence are ON. Path is also a function over time t and it is denoted by $path(i, j, t)$.
- The interval between ON state and immediate OFF state of a path between the nodes i and j is a **session**. Session is also a boolean function on time t and it is denoted by $session(i, j, t)$.

Path duration is an important metric for testing communication protocols. For example, some protocols referred as connection-oriented protocols require the path between source and destination to be ON throughout the communication.

7 Related Works

An activity based mobility model is introduced in [30]. In this model, simulation region is divided into multiple cells and each cell represents a unique location. Node selects an activity (or trip purpose). Each activity has an associated time of day, duration and location. Given the current activity, the current time period and the node type, the next activity is randomly selected from the corresponding entries in activity transition matrix. Once the activity is selected, its duration is chosen from the activity duration matrix. Node then travels to a new location, determined from the type of the activity. After reaching at the destination cell, node stays there for the duration of the activity and the process repeats.

A generalized model is introduced in [22] and referred as *Graph-based mobility model*. In this model, the graph represents the locations that the user might visit and the edges model the connection between these locations (e.g. streets or train connections). Initially each mobile node is placed at random vertex in the graph and selects another vertex randomly as its destination and then moves towards it using shortest possible path. After reaching at the destination, node pauses for some randomly selected period and then selects another destination and the process repeats.

Another simple realistic mobility model is introduced using random waypoint model in [23] and referred as *Restricted Random Waypoint Model* (also called as localized random waypoint model). To bring realism to random waypoint model, towns and highways are introduced. Towns are sub geographic regions that are connected with highways inside a simulation region. Node moves with the random waypoint mobility model inside towns for majority of the time. However, after a certain number of movements in the same town, a node moves to another town over a highway.

GEMM proposed in [16] brings realism in to random way point model by introducing attraction points, activities, roles and group behavior. Given the role type, activity is chosen based on its trigger time and hence the attraction point is selected based on its popularity level and the activity triggered. Nodes may move between attraction points or any other random location (similar to random waypoint).

To simulate the geometry of city streets, *city section mobility model* [28], is proposed. It models a section of the city having a downtown area with freeway. Initially, each mobile node is placed randomly on some point in a street, and then randomly chooses a new location on some other street as the destination. It then travels towards the destination through shortest path between two points keeping the safe driving characteristics such as speed limit and safe distance between two nodes. Upon reaching the destination, the node pauses at its current location for a fixed period and repeats the process until the simulation ends.

To synthesize the real-world mobility, couple of models are introduced in [20] and referred as *Freeway mobility model* and *Manhattan mobility model*. The movement of node is restricted to pathways in the simulation field. In freeway mobility model, the map consists of several freeways and each freeway has lanes in both directions. The velocity of mobile node is temporally dependent on its previous velocity. Also, the velocity of the following node cannot exceed the velocity of preceding node - also known as spatial dependency. In Manhattan mobility model, the map consists of horizontal and vertical lines representing downtown area. The movement of node is restricted to horizontal and vertical pathways in the simulation field. At an intersection, the mobile node can turn left, right or go straight with a probability of 0.25, 0.25 and 0.5 respectively. Unlike freeway mobility model where node cannot change the lane, Manhattan mobility model gives a node some freedom to change its direction. Except the above difference, the Manhattan model is the same as Freeway model because the Manhattan mobility model also has high spatial and temporal dependence.

Another mobility model is introduced in [21] and referred as *Obstacle Mobility Model*. A realistic movement model is created through the incorporation of *obstacles* and pathways using *Voronoi diagram* [29] of obstacle vertices in the simulation field. These obstacles are exploited to restrict node movement as well as obstruct wireless transmissions. In this model each node selects a destination point randomly and then moves to that point using the shortest route from its current location. After reaching the destination, node pauses for some time period. The process repeats. The obstacles also impact the way radio propagates. However, since the location of obstacles and destination of each motion phase is randomly chosen, a certain level of randomness still exists for this model.

STRAW [36] is specifically designed for VANET, uses a vehicular mobility model on US road topology, which constraints the node movement to streets of real US cities. It is limited in providing the whole world map data and also the generated mobility trace can only be exported for SWAN [37].

Another mobility model is designed for VANET known as SUMO [25]. It also extracts the real world road topology using OSM database. Its main features include multi-lane streets with lane changing capability, collision detection and intersection based rules. Though it is a powerful tool to generate traffic network simulation, it does not have option to export mobility trace, which limits its use in network protocol studies. MOVE [8] is built on top of SUMO, which generates mobility models for vehicular network simulations. The mobility trace can be exported for network simulation tool such as NS-2 or GloMoSim.

VanetMobiSim [10] is an extension of another tool called CanuMobiSim which generates mobility trace from a user specified XML configuration of a mobility pattern. VanetMobiSim can import maps from the TIGER [40] database as well as can generate random maps using Voronoi tessellation. Its main features include vehicle acceleration, deceleration, multi-lane streets with lane changing functionality, but does not provide a support of multiple vehicle types. However the generated mobility trace can be exported for different mobile networks tools including NS-2, GloMoSim, and QualNet.

Another mobility model is proposed with the intend to model a real world environment, specifically designed to study urban pedestrian mobility known as MobiReal [1]. It is a network simulator that can simulate realistic mobility of nodes using probabilistic destination selection. User creates street graph and routes through GUI using mouse clicks, which is time consuming. It was initially designed to simulate MANETs and was later extended to include VANETs by incorporating the traffic simulator NETSTREAM [12] developed by TOYOTA. Since NETSTREAM is a propriety software, user can not access and modify this part of the simulator which limits its wide usage.

CityMob generates the mobility based on three models; Simple Model, Manhattan Model, and Downtown Model [4]. Road network is generated randomly using Manhattan grid model and does not provide the facility to create user defined roads or road extraction through GIS data. All streets are two-way, with lanes in both directions and node moves with random speed, within an user defined range of values. Generated mobility trace can be exported in NS-2 format.

FreeSim is specifically designed for VANET that allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure [31]. The traffic data used by the simulator can be user generated or be converted from real-time data gathered by a transportation organization. However FreeSim does not support multiple node types and it does not have option to export mobility trace, which limits its use in network protocol studies. which is a serious shortcoming.

The IMPORTANT framework [20] contains metrics to capture mobility characteristics and evaluate the impact of the performance of routing protocols but does not have visualization or animation capabilities.

GMSF [7] is a tool to simulate and analyze node mobility in vehicular ad hoc networks. The road topology is extracted from official Swiss national map (Landeskarte), which constraints the node movement to swiss streets. The mobility trace can be generated using Random Waypoint, Manhattan, GIS, and MMTS models. Only roads which are accessible by vehicles are imported into the road topology. The mobility trace can be exported in multiple formats.

SUMO, MOVE, STRAW, and VanetMobiSim all have good software features. However, only VanetMobiSim provides excellent trace support [2]. Generic simulation tools like NS-2 [34], OPNET [35], QualNet [39] support only limited mobility models such as random waypoint model and its variations. Similar software to generate various random mobility models is proposed in [5]. Our software is more generic and applicable to a wide range of systems including a class of high level VANETs. Also, our tool is more comprehensive in creating, visualizing, and analyzing the mobility models.

8 Conclusion

In this paper, we presented unified approach to generate realistic mobility models and presented a software that we designed to do the same. Using the proposed software, several realistic mobility models with varying characteristic can be generated and analyzed. Our software offers several tools to carefully visualize and analyze the generated mobility trace before it is used in actual simulation. Such a deeper insights on mobility trace would be extremely beneficial for understanding the simulation behavior of the system and will be helpful to interpret the results in more meaningful way.

Although our software is fairly comprehensive, it can be improved in several directions. Particularly, it can be improved in the statistical analysis domain.

References

1. Maeda, K., Uchiyama, A., Umedu, T., Yamaguchi, H., Yasumoto, K., Higashino, T.: Urban Pedestrian Mobility for Mobile Wireless Network Simulation. Ad Hoc Network 7(1), 153–170 (2009)
2. Martinez, F., Toh, C., Cano, J., Calafate, C., Manzoni, P.: A Survey and Comparative Study of Simulators for Vehicular Ad hoc Networks (VANETs). In: Wireless Communication and Mobile Computing (2009)
3. Stojmenovic, I.: Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions. IEEE Communications Magazine, 102–107 (2008)
4. Martinez, F., Cano, J., Calafate, C., Manzoni, P.: CityMob: A Mobility Model Pattern Generator for VANETs. In: Proc. of the IEEE International Conference on Communications (ICC), pp. 370–374 (2008)
5. Aravind, A., Cui, X.: RMobiGen: A Trace Generation, Visualization, and Performance Analysis Tool for Random Mobility Models. In: Proc. of the High Performance Computing and Simulation Symposium (ACM/SIGSIM, SpringSim), pp. 403–412 (2008)
6. Aravind, A., Tahir, H., Baldeep: RealMobiSim: realistic mobility simulator and analyzer. In: Proc. of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, pp. 186–189 (2008)
7. Sommer, P.: Design and Analysis of Realistic Mobility Models for Wireless Mesh Networks, Msc. Thesis, Communication Systems Group, Computer Engineering and Networks Laboratory (TIK), Institute of Technology Zurich (2007)
8. Karanadi, F., Mo, Z., Lan, K.-C.: Rapid Generation of Realistic Mobility Models for VANET. In: IEEE Proc. of the wireless Communications and Networking Conference, pp. 2506–2511 (2007)
9. Andel, T.R., Yasinsac, A.: On the Credibility of Manet Simulations. IEEE Computer, 48–54 (2006)
10. Harri, J., Filali, F., Bonnet, C., Fiore, M.: Vanetmobisim: Generating Realistic Mobility Patterns for VANET. In: Proc. of the 3rd International Workshop on Vehicular Ad Hoc Networks (ACM VANET), pp. 96–97 (2006)
11. Harri, J., Filali, F., Bonnet, C., Fiore, M.: Mobility Models for Vehicular Ad-hoc Networks: A Survey and Taxonomy, Research Report RR-06-168, Department of Mobile Communications, France (2006)
12. Mori, H., Kitaoka, H., Teramoto, E.: Traffic Simulation for Predicting Traffic Situations at expo 2005. R&D Review of Toyota CRDL 41(4), 45–51 (2006)
13. Bai, F., Helmy, A.: A Survey of Mobility Models in Wireless Ad-hoc Networks. In: Wireless Ad Hoc and Sensor Networks. Springer, Heidelberg (2006)
14. Kurkowski, S., Camp, T., Colagrosso, M.: Manet Simulation Studies: The Incredibles. ACM Mobile Computing Communication Review 9(4), 50–61 (2005)
15. Chu, T., Nikolaidis, I.: Node Density and Connectivity Properties of the Random Waypoint Model. Computer Communications 27, 914–922 (2004)
16. Feeley, M., Hutchinson, N., Ray, S.: Realistic Mobility for Mobile Ad-hoc Network Simulation, Ad-Hoc, Mobile, and Wireless Networks, pp. 324–329 (2004)

17. Kotz, D., Newport, C., Gray, R.S., Liu, J., Yuan, Y., Elliott, C.: Experimental Evaluation of Wireless Simulation Assumptions. In: Proc. of the ACM MSWiM, pp. 78–82 (2004)
18. Ray, S.: Realistic Mobility for Manet Simulation, MSc. Thesis, University of British Columbia, Vancouver, Canada (2004)
19. Saha, A.K., Johnson, D.B.: Modeling Mobility for Vehicular Ad-hoc Networks. In: Proc. of the ACM Workshop on Vehicular Ad Hoc Networks, VANET (2004)
20. Bai, F., Sadagopan, N., Helmy, A.: The IMPORTANT Framework For Analyzing The Impact of Mobility on Performance of Routing Protocols for Ad-hoc Networks. In: Proc. of IEEE INFOCOM, pp. 825–835 (2003)
21. Jardosh, A.P., Belding-Royer, E.M., Almeroth, K.C., Suri, S.: Towards Realistic Mobility Models for Mobile Ad-hoc Networks. In: Proc. of the 9th Annual International Conference on Mobile Computing and Networking (2003)
22. Tian, J., Hahner, J., Becker, C., Stepanov, I., Rothermel, K.: Graph Based Mobility Model for Mobile Ad-hoc Network Simulation. In: Proc. of the 35th Annual Simulation Symposium (2002)
23. Blazevic, L., Giordano, S., Le-Boudec, J.Y.: Self Organized Terminode Routing. Journal of Cluster Computing 5(2), 205–218 (2002)
24. Camp, T., Boleng, J., Davies, V.: A survey of Mobility Models for Ad-hoc Network Research. In: Wireless Communication and Mobile Computing, pp. 438–502 (2002)
25. Krajzewicz, D., Hertkorn, G., Rossel, C., Wagner, P.: SUMO (Simulation of Urban Mobility) - an open-source traffic simulation. In: Al-Akaidi, A. (ed.) 4th Middle East Symposium on Simulation and Modeling, pp. 183–187 (2002)
26. Cavin, D., Sasson, Y., Schiper, A.: On the Accuracy of MANET Simulators. In: ACM Proc. of POMC, pp. 38–43 (2002)
27. Pawlikowski, K., Joshua Jeong, H.-D., Ruth Lee, J.-S.: On Credibility of Simulation Studies of Telecommunication Networks. IEEE Communications Magazine, 132–139 (2001)
28. Davies, V.: Evaluating Mobility Models Within an Ad-hoc Network, Master's Thesis, Colrado School of Mines (2000)
29. Berg, M.D., Kreveld, M.V., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Heidelberg (2000)
30. Scourias, J., Kunz, T.: An Activity-based Mobility Model and Location Management Simulation Framework. In: Proc. of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 61–68 (1999)
31. FreeSim - Freeway Simulator, http://www.freewaysimulator.com/
32. GloMoSim - Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/
33. Geo Base, http://www.geobase.ca/geobase/en/index.html
34. The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html
35. OPNET Simulator, http://www.opnet.com/
36. STRAW - Street Random Waypoint mobility model, http://www.aqualab.cs.northwestern.edu/projects/STRAW/
37. SWAN - Simulator for Wireless Ad-hoc Networks, http://www.eg.bucknell.edu/swan
38. Open Street Map, http://www.openstreetmap.org/
39. QualNet Network Simulator, http://www.scalable-networks.com/
40. Tiger Database, http://www.census.gov/geo/www/tiger/

An Efficient Clusterhead Placement for Hybrid Sensor Networks

Abdellah Chehri and Hussein Mouftah

School of Information Technology and Engineering (SITE)
University of Ottawa, 800 King Edward Avenue, Ontario,K1N 6N5, Canada
{achehri,mouftah}@uottawa.ca
www.site.uottawa.ca/

Abstract. Wireless sensor networks (WSN) have attracted considerable interest in recent years as a convenient, flexible and low cost alternative to wired sensor in many contexts. Deployment of sensor nodes is a critical task as deployment should be efficient to increase network lifetime. To address this problem, a hierarchical architecture is employed. In this paper, we propose an incremental deployment algorithm. By examining the node position, and the sensing cover area, the proposed algorithm indicates which position should deployed clusterheads and how many new clusterhead are necessary to cover the desired monitoring area. A comparison between the proposed algorithm with grid-based and augmenting placement are analyzed.

Keywords: Wireless sensor networks, topology control, optimization, clusterhead placement.

1 Introduction

Wireless sensor network is composed of large number of small, inexpensive, battery operated sensor devices. Their goal is to collect information remotely form environment. Afterward, they forward the collected data, via multiple hops, to a central node (i.e., sink) [1], [2]. However, one problem with this architecture is that many nodes are often employed to forward data. Since sensor nodes have a limited energy, this architecture is not scalable. To address this problem, a hierarchical architecture has been proposed [3], [4].

In hierarchical architecture, besides sensors and a sink, there are some nodes which act as clusterhead. On the one hand, these nodes have a great calculation capacity, great memory and high routing capabilities; on the other hand they are more costly. Each clusterhead is in charge of a set of cluster of nodes. These sub-networks are obtained by making a spatial division of the whole network. By aggregating, the clusterhead refines the transmitted data from sensor nodes. Then, it post-processed and routed the data to the sink. Such architecture has the advantages of being more practical, scalable, economical and capable of extending the lifetime of the network [5].

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 123–134, 2010.
© Springer-Verlag Berlin Heidelberg 2010

In this work, and contrary to existing hierarchical architectures works [1], [2], [6], we take into consideration multi-hop propriety. In fact, we'll exploit the multi-hop nature of WSN, where the tiny nodes can connect to the clusterhead through multiple hops. Note that, although we consider sensor networks in this work, the concepts developed in this work can be extended to any hierarchical mesh network in general.

Our main focus in this paper is the topology design of sensor network. Thus, given that sensor resources are limited, it is necessary to determine the efficient placement where the clusterhead should be adds. This efficient placement was represented as an optimization problem which can be defined by a set of inputs and output variables, and constraints.

Before discussing this clusterhead placement in detail, we shall first give the related works on topology design. The system architecture is formally described in Section III. The methodology and two mains algorithms for clusterhead placement are presented in Section IV. Simulation results are included in Section V which illustrates the performance of the proposed algorithm. Finally, we conclude the paper in Section VI.

2 Related Works

Many algorithms have been proposed in wireless network topology design. For example, an integer linear programming (ILP) solution for network coverage while minimizing the sensor's number has been investigated in [7]. The authors formulate the sensor placement problem in terms of cost minimization under coverage constraints. Then, they develop an integer programming (ILP) model to solve the sensor deployment problem.

In [5], the nodes are added progressively to maintain global network connectivity. The node placement algorithm has been modeled as a minimum Steiner tree with bounded edge length problem.

Hu in [8], has firstly, investigated on topology according to Delaunay triangulation. Then, he used an algorithm in which the degree of each node was adjusted through neighbor negotiation to ensure that each node has similar number of neighbors.

Sensor placement that induces regular topologies such as circular, star, and hexagonal has also been analyzed in [9].

In addition to the connectivity, others works are concerned only on network lifetime. In [10], the network lifetime has been analyzed through a joint problem of energy provisioning and relay node placement. In this work, the authors used an efficient polynomial-time heuristic algorithm.

A fuzzy logic approach for cluster-head election has proposed in [11]. The method was based on three descriptors, (1) energy, (2) concentration and (3) centrality. Their simulation shows that depending upon network configuration, a substantial increase in network lifetime can be accomplished as compared to probabilistically selecting the nodes as cluster-heads using only local information.

The authors in [2] formulate a minimum-power wireless sensor network deployment problem. The objective is to determine the locations of a given number of relays and the corresponding link flows in order to minimize the total power consumption.

Furthermore, the topology optimization has been also investigated in wireless mesh network [12], [13]. The technique proposed by Aoun et al., was consisting in placing a minimum number of gateways such that quality-of-service (QoS) requirements are satisfied. The authors uses a polynomial time near-optimal algorithm which recursively computes minimum weighted dominating sets (DS), while consistently preserving QoS requirements across iterations [12].

In [13], Wong et al. addressed the gateway placement problem in two separate settings: either minimizing communication delay or minimizing communication cost.

This paper tackles the problem of adding some advanced nodes to maintain global connectivity in WSNs when transmission range of all tiny sensors was restricted. Our objective is to minimize the number of these required sensors, as they contribute to the overall cost of a WSN. Our study is motivated by an important class of static sensor networks, in which the deployment of theses sensors are known; by a manual deployment or by any localization techniques [14].

3 System Description

3.1 Network Architecture

Wireless sensor networks have been considered in the literature many different models; however, it can be organized into two main classes: *flat* and *hierarchical* topology. In flat architecture, all nodes have the same processing capabilities, same transmission rage. The nodes are frequently used as relay in order to increase connectivity and support routing between the different nodes.

In the hierarchical architecture, there is a presence of few nodes which are assumed to have higher capabilities such as longer transmission range, higher energy resources and higher processing capabilities. In addition, these nodes are required to constitute connected subnetworks between each other. The two-tiered network architecture is motivated by recent advances in signal processing and data fusion, which can exploit redundancy in information collected among sensors. This hybrid model will be considered as shown in Figure (1).

3.2 Network Model

Let N denotes the set of tiny nodes that are deployed in a two dimensional Euclidean plane $\Re$. All node have a fixed and common transmission range ρ. The adjacency matrix of $G(V, E)$ is a matrix with rows and columns labeled by the graph vertices V, with a 1 or 0 in position (m, n) according to whether v_m and v_n are directly connected or not. For the undirected graph G, the adjacency matrix is symmetric.

On the other hand, clusterhead have two sets of wireless transceivers working on two different wireless channels, one for the communication with the tiny

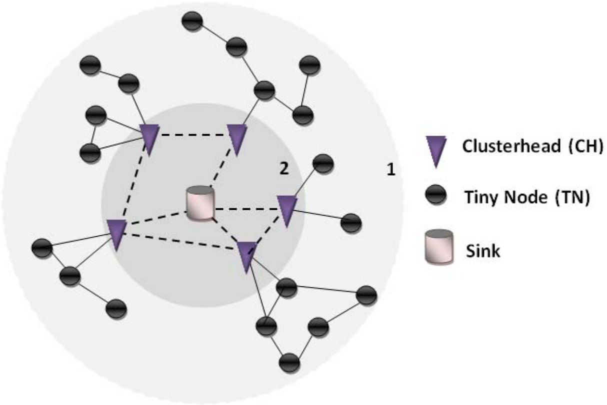


Fig. 1. Two-tiered hybrid wireless sensor network architecture

nodes, and the other for the communication with other clusterhead. We assume that a link is established between two nodes as long as the distance between the nodes is smaller than the corresponding transmission range.

3.3 Clusterhead Placement

As we have mentioned earlier, the main variable of the deployment is the clusterhead position. Suppose that we have N sensor node handled with M cluster head. There is C_N^M possible configurations of the hybrid network. For example, For $N = 100$ and $M = 10$, we have $1.7 \cdot 10^{13}$ possible configurations. It is so infeasible to explore all solutions in order to find the optimal hybrid topology.

3.4 Optimization Variables

Table 1 lists the notation used in this paper.

3.5 Optimization Goals

Like all optimization problems, a number of constraints must be verified during the resolution process. In the case of the clusterhead placement; we face several constraints which are either technological constraints or traffic-imposed constraints. We enumerate in the following sub-sections these different constraints.

The Capacity Constraints. The capacity constraint which should be verified by each optimization problem is related to capacity limitation of the clusterhead. In fact, the capacity (C_j) of clusterhead must be one of the discrete capacity offered by the clusterhead; and the sum of links capacity should not exceed the bandwidth of the clusterhead.

Now, let C_j denote the capacity in terms of data rate that the j^{th} clusterhead can handle. We need to allocate this capacity to different children tiny nodes.

Table 1. Main Algorithm parameters

Input
1) $A = \{1, 2, ..., M\}$: Set of the clusterhead candidate to be deployed.
1) $B = \{1, 2, ..., N\}$: Set of tiny nodes.
2) (x_i, y_i): The physical position of the tiny node, $i \in B$.
3) ρ_i: The maximum transmission range of a tiny node, $i \in B$.
4) R: The maximum transmission range of a clusterhead.
5) h_{max}: The maximum number of hops allowed for a node.
6) D_{max}: The maximum number of clusterhead.
7) λ_i: The data rate generated by node, $\forall i \in B$.
7) C_j: The maximum capacity that can be handled by clusterhead $j \in A$.
Output
1) $(\hat{x}_j, \hat{y}_j)$ The optimal position of the clusterhead, $j \in A$
$\% = M/N$ percentage of clusterhead in the network.

Let $\phi_{i,j}$ denote the number of node that are i hop away from j^{th} clusterhead. Also for simplicity, we assume that traffic generated by each node ($\forall i \in B$) is equal to λ_i.

Note that the traffic generated from tiny nodes further away from the j^{th} clusterhead will be relayed by other tiny nodes more closer to the clusterhead. Thus the effective traffic ($\mathit{\Lambda}_i$) form all i^{th} hop include the traffic generated by all tiny node through the route.

$$\mathit{\Lambda}_j = \sum_{i=1}^{h} \phi_{i,j} \cdot \lambda_i \quad \forall j \in A, i \in B. \tag{1}$$

The capacity constraint which guarantees the feasibility of the solution that the flow generated in each communication links should not exceed its capacity. Then,

$$\mathit{\Lambda}_j \leq C_j \quad \forall j \in A \tag{2}$$

The Cost Constraint. The topology obtained should have a total cost D less that the maximal cost D_{max} ($D \leq D_{max}$). Where D_{max} is the maximal budget allowed and which is an input of the design problem.

QoS Requirements. The QoS requirements are given per end-to-end packet transmission in terms of maximum allowed delay T_D and BER (P_{req}). The probability for a successful end-to-end transmission over h hops can be found as the product of the probability for successful transmission for each hop, assuming bit error events are independent between the hops (when a decode-and-forward scheme is considered).

The following product must be satisfied:

$$(1 - P_i) = \prod_{i=1}^{h} (1 - P_i) \geq (1 - P_{req}) \tag{3}$$

where P_{req}, is the required BER (parameter design). The total transmission time is calculated as the sum of the transmission for all nodes (through the route). For end-to-end delay constraint (T_d), a valid configuration must satisfy:

$$T_{tot} = \sum_{i=1}^{h} T_{on}.i \geq T_D \tag{4}$$

T_D is a parameter design. T_{on} is the time required for the node to transmit the packet.

Unfortunately, P_i and T_{on} are difficult to estimate accurately. They depend on many parameters such as, the modulation, the multiuser interferences, and even the hardware's characteristics of node. However, these two parameters are depending on the multihop h. So, the QoS constrains, can be resumed to the number of hop maximal (h_{max}) between the tiny node and clusterhead. This value should be small to expect that the QoS constrains will be respected.

Connectivity Constraint. The connectivity constraint guarantees that all clusterhead can reach the sink. This ensures that the information collected by the tiny node can reach the sink. This problem is NP-hard, which can be proven through a straight forward reduction [15]. Due to space limit, the proof for NP-hardness is not included in this paper and can be found in a [15].

4 Methodology

We now develop a model to define clusterhead placement. Three constraints namely; the quality of service (QoS) constraint, the connectivity constraint and the network cost constraint, had been considered. The QoS constraint guarantees that all traffic can be handled by clusterhead, which implies that all tiny nodes can reach at least one clusterhead with a maximum hop h_{max}. Furthermore, the connectivity constraint guarantees that all clusterhead can reach the sink. In addition, the network cost constraint assumes that the number of clusterhead doesn't exceed a fixed value.

The immediate solution for finding the optimal position of the clusterhead is to study all possibilities and then to determine the optimal position where the clusterhead should be add. Of course, this position is which optimizes the goal of the optimization problem. As we have mentioned previously, it is very expensive in processing time to find the optimal topology by the examination of all possibilities. As solution which is more efficient consists of finding the optimal position that respect all constraints without processing all possibilities. Therefore, the estimated position of clusterhead is suboptimal solution.

The proposed planning algorithm starts by the evaluating the performance of the initial topology and then it executes two main steps which are:

- Finding the optimal position where the clusterhead can be added in the networks;
- Performance evaluation of the current hybrid topology.

The first step consists for looking for the optimal position where we should add the clusterhead in order to improve the network performance. In the second step, and according to several performance criteria, the algorithm acts to evaluate the performance metrics of only the new hybrid cluster. The resultant clusterhead position will be considered as the optimal temporary only if we ensure that all constraints were respected. The process stops when the optimal topology is reached or the number of cluster head added is equal to the maximum number.

The step of performance evaluation of current hybrid cluster is the more critical step. It is critical considering that after this step we decide if the placement of the clusterhead is optimal one or not. After evaluating performance of the current hybrid topology during an iteration of the algorithm, we should add a new clusterhead.

$$\text{minimize} \left(\sum_{\forall j} \theta_j \right) \quad \forall j \in A \tag{5}$$

subject to

$$\sum_{i=1}^{h} \phi_{i,j} \cdot \lambda_i \leq C \;\; \forall j \in A, i \in B \tag{6a}$$

$$h_i \leq h_{max} \forall i \in B \tag{6b}$$

$$\sum_{\forall j} \theta_j \leq D_{max} \;\; \forall j \in A \tag{6c}$$

$$\sum_{\forall j} \theta_i = 1 \;\; \forall i \in B \tag{6d}$$

$$G_i \cap G_j \neq \oslash \;\; \forall i, j \in A, i \neq j \tag{6e}$$

The traffic constraint in (6a) specifies that the capacity allocated to j^{th} clusterhead should be respected. Inequality (6b) ensures that only node closer than (h_{max}) are selected by the clusterhead. Constraint (6c) requires that the total deployment cost of clusterhead is limited by the total cost (D_{max}). Condition (6d) denotes that each tiny node should be assigned only to one clusterhead. The connectivity constraint in (6e), specifies that the clusterhead have to form a connected network, where $G_{\forall j}$ is coverage of the j^{th} clusterhead.

4.1 Procedure

In this section, we describe three algorithms to solve the clusterhead placement problem for randomly deployed nodes.

Proposed Approach. Our proposed schema is based on recursive greedy algorithm. The basic idea of the algorithm is to partition the network nodes into a minimum number of disjoint clusters, and place a clusterhead in each cluster. We use the Greedy Dominating Independent Set (DIS) [16] heuristic to determine a set of clusterheads, which are used as possible clusterhead locations. As the greedy algorithm is a heuristic algorithm, it cannot guarantee to find the optimal solution during a single iteration. Thus, we obtain multiple solutions by running the greedy algorithm a preset number of times and choose the best amongst them as the final solution.

Algorithm 1. Efficient Clusterhead Placement

```
Require: Number of nodes ($N$), Node Positions ($x_{i,j}$), Maximum number of hop
    ($h_{max}$), Maximum number of clusterhead ($D_{max}$), $\rho$, $R$, $\lambda$, $C$.
 1: Initialize by obtaining clusters.
 2: Place first clusterhead at the center of the biggest cluster $\Gamma_{(k=1)}$
 3: $\theta$ =1
 4: while ($\theta < D_{max}$) do
 5:    Remove node with ($h_i > h_{max}$) away from the clusterhead (constraint 6b).
 6:    if $\left(\sum_{i=1}^{h} \phi_{i,j} \cdot \lambda_i \leq C, \quad \forall i\right)$ then
 7:       $\theta \leftarrow \theta + 1$
 8:       Save clusterhead position
 9:    end if
10:    Cluster the remaining nodes $\Gamma_{\forall k}$.
11:    Place the additional clusterhead at cluster center which serves the most nodes
       and is within distance $R$ from at least clusterhead.
12:    Goto (5)
13:    if $G_i \cap G_j \neq \oslash \quad \forall i \neq j$ then
14:       Find node $j$ with max $\left(\sum_{j=1}^{h} K_{i,j} \cdot \lambda_i \leq C\right)$ closet to any $\theta_i$.
15:       Place additional $\theta$ at distance $R$ from $\theta_i$ towards node $j$.
16:       $\theta \leftarrow \theta + 1$
17:    end if
18: end while
19: return $\hat{\theta} = [(\hat{x}_1, \hat{y}_1), \cdots, (\hat{x}_M, \hat{y}_M)]$, % = M/N    best configuration
```

Augmenting Placement. The augmenting placement represents another alternative for clustering. This approach involves placing the clusterhead independently until such that all nodes in the network are covered while respecting all constraints. The idea of the augmenting placement algorithm is similar to the greedy algorithm placement with respect to its internal procedure; however, it does not make greedy decisions regarding the next placement of additional clusterhead. Any placement providing subsequent coverage to uncovered nodes is considered [12].

Grid-Based Placement. Grid-based deployment is a practical and efficient coverage-oriented deployment approach for moderate to large scale WSN applications due to its simplicity and scalability. The goal is to deploy sensors on a sensor field to guarantee a particular quality of service, if the properties of the terrain are predetermined. The terrain is generally divided into grids and sensors are carefully deployed at the grid points [17].

5 Results

In this section, the performances of the three algorithms were described. We assume that the tiny nodes are deployed randomly over a two-dimensional 100×100 square area.

5.1 Impact of Clusterhead Radio Range

We start by examining the effect of clusterhead radio range R. The performance of each algorithm has been evaluated for different values of R. The figure 2 has been obtained for each algorithm after average of 100 independent simulations. The maximum capacity of clusterhead was fixed C_j= 20 units, while the data rate generated by the tiny node was fixed to $\lambda_i = 1$. The radio range of tiny node is equal $\rho = 10$. During each simulation, 100 nodes are randomly deployed in the area.

As we expected, when we increase the value of R, the number of clusterhead required to meet all constraints decrease. This is because a great of R leads a huge overlap of wireless coverage, and hence, a few clusterhead are needed. Also, by comparing the performance of the different algorithms, we observe that our proposed approach gives the best configuration. In contrast, and for low clusterhead transmission radio, the Grid-based clustering schemes perform much worse. For low radio coverage, our approach required 1.3 to 2.45 less number of clusterhead compared to Augmenting placement and Grid-based algorithm, respectively. However, for high coverage ($R > 30$), all algorithms gives almost the same performance.

5.2 Impact of Network Size

Next, we study the impact of network size on the placement algorithms. We randomly deploy different number of nodes (from 30 to 200 nodes). However, the transmission range and the data rate generated by each node remains the same ($\rho = 10$, $\lambda = 1$ unit).

Figure 3 shows the number of required clusterhead using the different placement algorithms. As we would expect, an increase in the number of nodes leads to a larger number of clusterheads. Moreover, our proposed approach continues to perform well compared to the other algorithms. We can also see that the Grid-based algorithm lead the same number of clusterhead for low network size ($N < 100$). For high network size, this number should be increased in under to ensure the capacity criteria constraint.

5.3 Impact of Number of Hop

This study was conducted by varying only the number of hop allowed to the nodes; the other constraints are maintained constant.

One can easily notice that the effect of hop number is important for deployment strategy. The proposed approach still performs better than the Augmenting

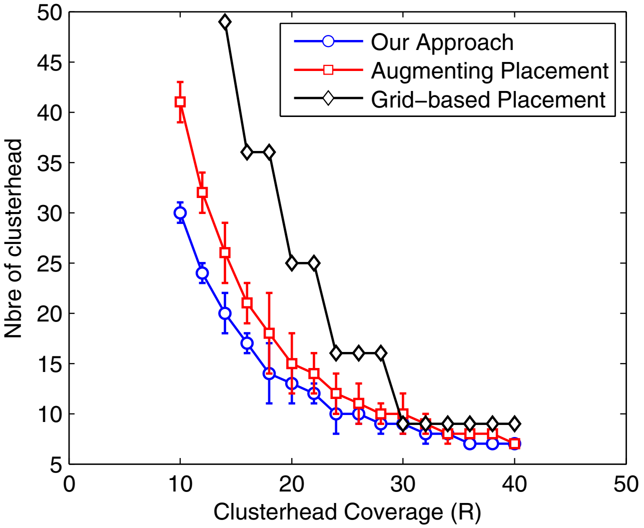


Fig. 2. Required clusterhead number vs. radio range, N = 100, ρ =1, C = 20, λ =1, h_{max} =3

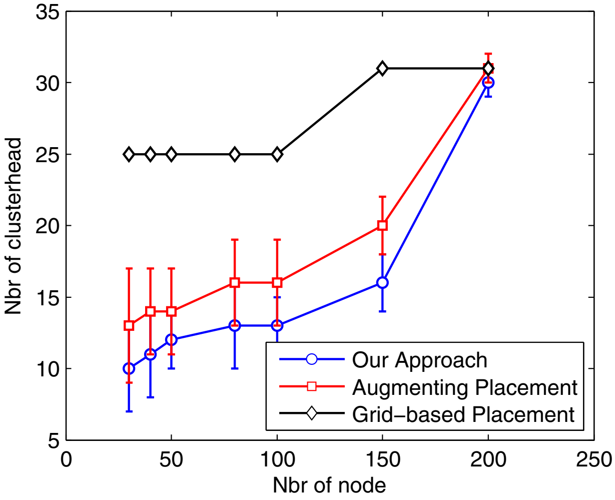


Fig. 3. Required clusterhead number vs. network size, R = 20, ρ =1, C = 20, λ =1, h_{max} =3

placement and the Grid-based deployment. The Grid-based algorithm required the same number of clusterhead whatever the number of hop. However, this remark is true only for a great hop number (more than 3 hops). For example, for h = 4, the proposed algorithm required 9 clusterheads; while the Augmenting

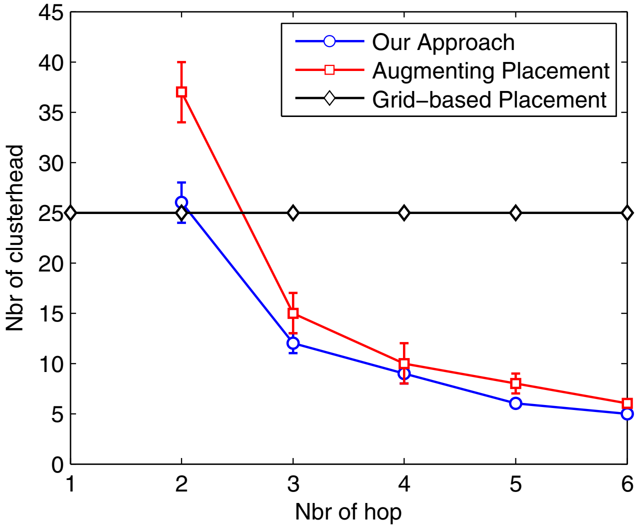


Fig. 4. Required clusterhead number vs. allowed hop number, $R = 20$, ρ =1, $C = 20$, λ =1, $N = 100$.

placement and Grid-based algorithms require 11 and 25 clusterheads, respectively. Nevertheless, and for low hop $h \in \{1, 2\}$, surprisingly, the simplest method (Grid-based) gives the best performance.

6 Conclusion

Wireless sensor network topology design has attracted significant research interest. This interest is expected to grow further with the proliferation of WSN application. In this paper, we have analyzed the problem of efficient clusterhead placement for hierarchical WSN architecture.

To address this problem, we have proposed an optimization method to determine the positions of clusterhead which minimize the mean cost of network while respecting some constraints. We have seen that the proposed approach, compared to two known methods, gives the best solutions over varieties of scenarios.

References

1. Raghavendra, C., Sivalingam, K., Znati, T.: Wireless Sensor Networks. Springer, Boston (2004)
2. Li, W., Cassandras, C.G.: A minimum-power wireless sensor network self-deployment scheme. In: IEEE Wireless Communications and Networking Conference, vol. 3, pp. 1897–1902. IEEE Press, Los Alamitos (2005)

3. Holger, K., Willig, A.: Protocols and architecture for Wireless sensor networks. John Wiley and Sons, Chichester (2005)
4. Lewis, F.L.: Wireless sensor networks. In: Cook, D.J., Das, S.K. (eds.) Smart Environments: Technologies, Protocols, and Applications. John Wiley, Chichester (2006)
5. Cheng, X., Du, D., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. In: Wireless Networks, vol. 14(3), pp. 347–355. IEEE Press, Los Alamitos (2008)
6. Gupta, G., Younis, M.: Fault-tolerant clustering of wireless sensor networks. In: Proceeding of IEEE WCNC, pp. 1579–1584. IEEE Press, Los Alamitos (2003)
7. Chakrabarty, K., Iyengar, S., Qi, H., Cho, E.: Grid coverage for surveillance and target location in distributed sensor networks. IEEE Transactions on Computers 51, 1448–1453 (2002)
8. Hu, L.: Topology control for multi hop packet radio networks. IEEE Transactions on Communications 41(10), 1474–1481 (1993)
9. Biagioniand, E.S., Sasaki, G.: Wireless sensor placement for reliable and efficient data collection. In: IEEE HICSS 2003. IEEE Press, Hawaii (2003)
10. Hou, H., Shi, Y., Sherali, S., Midkiff, S.F.: On energy provisioning and relay node placement for wireless sensor network. IEEE Transactions on Wireless Communications 4(5), 2579–2590 (2005)
11. Gupta, I., Riordon, D., Sampalli, S.: Cluster-head election using fuzzy logic for wireless sensor networks. In: Proceedings of the 3rd Annual Communication Networks and Services Research Conference, pp. 255–260. IEEE Press, Halifax (2005)
12. Aoun, B., Boutaba, R., Iraqi, Y., Kenward, G.: Gateway placement optimization in wireless mesh networks With QoS constraints. IEEE Journal on Selected Areas in Communications 24(11), 2127–2136 (2006)
13. Wong, J., Jafari, R., Potkonjak, M.: Gateway placement for latency and energy efficient data aggregation. In: Proc. IEEE Int. Conf. Local Computer Network, pp. 490–497. IEEE Press, Los Alamitos (2004)
14. Savvides, A., Srivastava, M., Girod, L., Estrin, D.: Localization in Sensor Networks. In: Wireless Sensor Networks, ch. 15. Kluwer Academic Publishers, Dordrecht (2004)
15. Suomela, J.: Computational complexity of relay placement in sensor networks. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2006. LNCS, vol. 3831, pp. 521–529. Springer, Heidelberg (2006)
16. Bejarano, Y.: Efficient Integration of multihop wireless and wired networks with, QoS constraints. IEEE/ACM Trans. Networks 12(6), 1064–1078 (2004)
17. Takahara, G., Xu, K., Hassanein, H.: Efficient Coverage Planning for Grid-Based Wireless Sensor Networks. In: Proceedings of the IEEE International Conference on Communications (ICC). IEEE Press, Glasgow (2004)

RaceTrack: An Approximation Algorithm for the Mobile Sink Routing Problem

Yuan Yuan[⋆,⋆⋆] and Yuxing Peng

Department of Computer Science
National University of Defense Technology
kenyuan44@gmail.com, yuxingpeng@nudt.edu.cn

Abstract. In large-scale monitoring applications, randomly deployed wireless sensor networks may not be fully connected. Using mobile sink for data collection is one of the feasible solutions. For energy saving, it is necessary to plan a shortest route for the mobile sink. Mobile sink routing problem can be regarded as a special case of TSP with neighborhoods (TSPN) problem. In this paper, we propose a novel approximation algorithm called RaceTrack. This algorithm forms a "racetrack" based on the TSP route, which is constructed from the locations of the deployed sensor nodes. By using inner lane heuristic and concave bend heuristic of auto racing, and a shortcut finding step, we optimize the obtained TSP route within $O(n)$ computation time. Through formal proofs and large-scale simulations, we verified that our RaceTrack algorithm can achieve a good approximation ratio.

1 Introduction

With the availability of cheap sensor nodes and the progress of wireless technology, sensor networks have been widely deployed in many large-scale applications, such as environment monitoring [5] and ecosystem surveillance [13]. However, sensor nodes randomly scattered in a large monitoring region may not form a fully connected networks, unless the number of nodes achieves a certain huge value [9]. Thus, how to collect data from sparse sensor networks becomes a great challenge.

One feasible solution to this challenge is controlling a mobile sink(MS) to visit each node. When the MS enters into a node's radio range, it can fetch a quantity of sensed data. After a trip through all the nodes, the MS returns the data to the base station. Nevertheless, MS aided data collection is also up against the energy saving problem, such that we need to plan a shortest route for the MS to travel through all the sensor nodes, which we call as*MS routing problem*.

Intuitively, this problem can be simply reduced to a Traveling Salesman problem (TSP). Many literatures about the controlled mobility in sensor networks

[⋆] The first author would like to thank Dr. Kui Wu for his suggestions.

[⋆⋆] The work was supported by the National High-tech R&D Program of China (2009AA01Z142) and by the National Basic Research Program of China (2005CB321801).

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 135–148, 2010.
© Springer-Verlag Berlin Heidelberg 2010

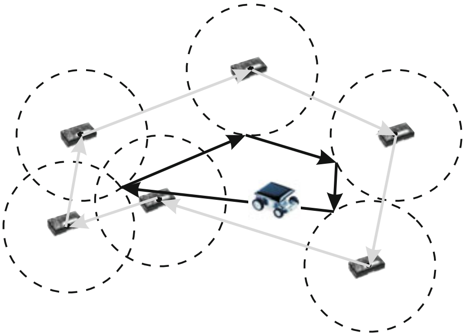


Fig. 1. MS routes: the gray one denotes the original TSP route without considering the radio range, while the black one utilizes the radio range

use this reduction, such as RD-VT algorithm [6] and PBS algorithm [15]. These algorithms all require that the MS should arrive in the exact positions where sensor nodes are located for communication. However, they may not benefit from radio ranges. As shown in Fig. 1, by utilizing radio ranges, the MS route could be further optimized.

Considering radio ranges, MS routing problem can be regarded as a special case of TSP with neighborhoods (TSPN). In TSPN, the salesman only has to reach the customs' neighborhoods to deliver his products. Clearly, TSPN is the generic form of TSP and hence is NP-hard. TSPN was first studied by E. Arkin and R. Hassin [3]. These years, many approximation algorithms are proposed to handle TSPN with different shapes of neighborhoods or with different relationships of neighborhoods, such as line segments, convex polygons, fat objects, and possibly overlapped regions or totally disjoint regions. Most of these algorithms apply TSP algorithm as an integral part. Excluding the computation time of TSP algorithm, the existing methods of TSPN with overlapped disk neighborhoods have a computation time all above $O(n^2)$ and even worse. However, for the MS routing problem in large-scale applications, since thousands of sensor nodes are deployed, the algorithms with heavy computation time may not be desired.

In this paper, we design a novel fast approximation algorithm, called "RaceTrack", for the MS routing problem with possibly overlapped disk radio ranges. Excluding the time of TSP algorithm, It can derive a good approximation solution within $O(n)$ optimization time, which is faster than the existing work. The major characteristic of our RaceTrack algorithm is that we transform the original problem into an auto racing problem. It first constructs a non-intersecting TSP route from the exact locations of the sensor nodes deployed, then based on the non self-intersecting property and loop property of the obtained TSP route, it forms a racetrack where the MS may travel through. Then using inner lane heuristic and concave bend heuristic in auto racing, RaceTrack quickly calculates the optimized substitute points for the vertices in the obtained TSP route and reduces the route length. Finally, it finds shortcuts to further optimize the route. The contributions of this work are as follows:

- We propose a novel approximation algorithm for the MS routing problem. It transforms the TSPN problem with possibly overlapped disk neighborhoods into an auto racing problem for the shortest route finding.
- We use 2 heuristics of auto racing and a shortcuts finding step to optimize the TSP route within $O(n)$ computation time, which is faster than the existing algorithms;
- We formally prove that the approximation ratio of RaceTrack algorithm can be described as $|T_S| < (1+\varepsilon)(|T_{MS}| + 2nr)$

The rest of the paper is as follows: Section 2 surveys the related work. Section 3 proposes the formal definition of the MS routing problem. Section 4 describes the details of RaceTrack algorithm and and Section 5 analyzes it. The performance evaluation is in Section 6 and the conclusion is in Section 7.

2 Related Work

The existing TSPN algorithms can be classified into two categories: TSP-posterior and TSP-anterior.

Most of the related algorithms belong to the formal category. TSP-posterior algorithms choose substitute points from the neighborhoods first, and then apply TSP algorithm to the set of points to make an optimum route. Gudmundsson[7] et al. present an $O(\log(n))$-approximation algorithm for overlapped polygons (n is the number of neighborhoods). It first computes a minimum covering box which covers all the polygons before using TSP algorithm. Dumitrescu and Mitchell[1] propose an (11.5)-approximation algorithm. It first finds the maximum independent set of all the overlapped disks, then constructs a TSP route from the center points of the independent disk set. The route combined with the boundary of each disk in the independent set is the final result. Elbassioni's algorithm [8] iteratively chooses substitute point p_i from neighborhood i to minimize $dist(p_i, \{p_1, p_2, \cdots, p_{i-1}\})$ (α is fatness, $\alpha = 4$ when neighborhood is disk) for disjoint fat neighborhoods, which achieves a $(9.1\alpha + 1)$-approximation ratio.

In contrast, the TSP-anterior algorithms construct a TSP route first and then optimize the route. For disjoint disk radio ranges in sensor networks, Yuan[2] et al. first construct a TSP route from the center points of these disks, then formulate a continuous optimization problem. Although a nearly optimum solution can be achieved, the algorithm consumes heavy computation time for the substitute points searching. Sugihara[11] et al. translate the MS routing problem with overlapped disk radio ranges into a so-called Label Covering problem. They search the shortcuts between two vertices of the obtained TSP route by a dynamic programming, whose computation time is $O(n^3)$. However, their algorithm does not fully benefit from the radio ranges, the results are loose.

So far as we known, RaceTrack algorithm is the first one that use geometric properties of TSP route and heuristics of auto racing to handle TSPN problem. We stress that RaceTrack can quickly find a TSPN solution with desired approximation ratio.

3 Problem Definition

Suppose that the MS can freely move in the monitoring region and a number of sensor nodes are uniformly randomly deployed in this region. After deployment, the exact positions of the sensor nodes can be obtained by using localization methods. Disk model [14] is often used to describe the wireless radio range. Thus, in the MS routing problem, the neighborhoods can be regarded as a number of randomly placed disks and some of them may overlap each other. We specify the MS routing problem formally as follows:

Definition 1
Given: *a square monitoring region of size* $R \times R$*, a MS start point* S*, and* n *equal size disk neighborhoods which represent the randomly deployed sensor nodes with radio ranges. Disk* i *is identified by its center point* $V_i(1 \leq i \leq n)$ *and its radius* $r(r > 0)$*.*

Goal: *find a shortest route* T_{MS}*, which starts and ends with point* S*, and travels through all the disks, i.e.* $\forall i(1 \leq i \leq n)$*,* $dist(V_i, T_{MS}) \leq r$*.*

4 RaceTrack Approximation Algorithm

RaceTrack approximation algorithm consists of four steps: first, it constructs a TSP route without self-intersecting from the center points of the disks and forms a racetrack. Second, it calculates an inner lane of this racetrack. Third, it chooses better paths at the concave bends of this racetrack. Finally, it finds shortcuts to further optimize the route.

4.1 Step 1: TSP Route Construction

By using exact TSP algorithms or approximation algorithms with untangling improvement, we construct a TSP route T_V without self-intersecting from the set of $n+1$ vertices $\{S, V_1, V_2, \cdots, V_n\}$. Assume that $T_V[i]$ $(0 \leq i \leq n)$ stores the i-th vertex of route T_V and $T_V[0] = S$. Clearly, T_V is a loose approximation of T_{MS}, hence we could use the vertex permutation of T_V as an approximation of that of T_{MS}. Based on this intuition, we can easily verify the quality of the optimized route in each next step.

After construct a TSP route, we then present an important fact of the optimum TSP route in Lemma 1:

Lemma 1. *The optimum TSP route is not self-intersecting.*

Proof. Given two intersecting edges found in an optimum TSP route, there exists a quadrangle formed by the four vertices corresponding to the two intersecting edges. Thus, the two intersecting edges could be regarded as the diagonals of this quadrangle. If we replace the two diagonals with two opposite sides, the connectivity of the TSP route is sustained but the route length could be further reduced. This contradicts the definition of the optimum TSP route. Therefore, optimum TSP route is not self-intersecting.

Based on Lemma 1 and the definition of TSP route, we could define *TSP polygon* as:

Definition 2. *TSP polygon is the simple closed polygon bounded by the TSP route without self-intersecting.*

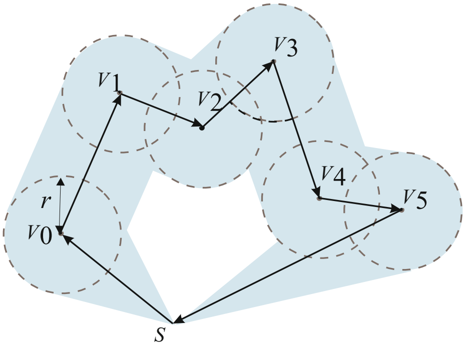


Fig. 2. A racetrack area swept by a disk with radius r, whose center moves along the TSP route

As shown in Fig. 2, we use a disk with radius r to sweep along the obtained TSP route, which forms a racetrack area where the MS may travel through. Thus, we transform the MS routing problem into an auto racing problem for finding a shortest route, along which the MS could access of all the disks within this racetrack.

4.2 Step 2: Inner Lane Substitute Points Choosing

Since T_V is a loose approximation of T_{MS}, thus we try to find a substitute point for each vertex of T_V, which could form a shorter route without permutation changing.

According to the observation of auto racing, we have the first heuristic as follows:

heuristic 1: *In non self-intersecting racetrack, the inner lane is often shorter than the outer lane.*

Based on heuristic 1, the appropriate substitute points could be found on the boundary of each disk inside the TSP polygon. We call these substitute points *inner substitute point*(ISP). Fig. 3 describes our ideas: we use C to denote the centroid of the $n+1$ vertices, which can be calculated as:

$$C = \frac{1}{n+1}(\sum_{i=1}^{n} V_i + S). \tag{1}$$

Then, based on C, we perform a triangulation for the TSP polygon. That is, for each disk, draw a line segment V_iC. Assume that line segment V_iC intersects the boundary of disk i at point R_i. I_i denotes the ISP of V_i. We have the following important result:

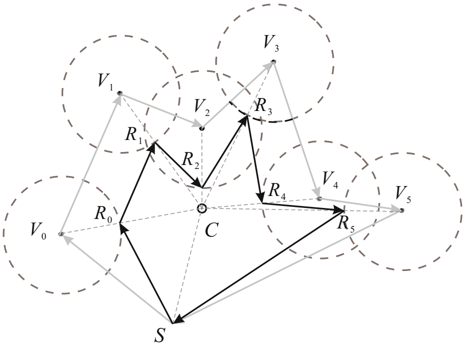


Fig. 3. Choosing inner substitute points

Theorem 1. *If ISP I_i for disk i are chosen based on the following rules:*

1. *if $dist(V_i, C) \geq r$, $I_i = R_i$;*
2. *if $dist(V_i, C) < r$, $I_i = C$,*

Then $\forall i, j (1 \leq i, j \leq n)$, $dist(V_i, V_j) \geq dist(I_i, I_j)$.

Proof. We describe the whole problem in a polar coordinate system, where the pole is at C. For any disk, say disk i, the polar coordinate of its center V_i can be represented as (d_i, α_i), where d_i is the alias of $dist(V_i, C)$ and α_i denotes the directed angle from axis x to line V_iC. According to the rules, if $d_i \geq r$, the polar coordinate of ISP I_i is $(d_i - r, \alpha_i)$, otherwise, the coordinate is $(0, 0)$.

In polar coordinate system, the distance between two points (d_i, α_i) and (d_j, α_j) equals to:

$$dist(V_i, V_j) = \sqrt{d_i^2 + d_j^2 - 2d_id_j \cos(\alpha_i - \alpha_j)}. \tag{2}$$

The formal proof is divided into three cases:

Case 1: Given two disks i and j, assume that both ISPs are derived from rule 1, which implies that $d_i \geq r$ and $d_j \geq r$. Since $dist(V_i, V_j) > 0$ and $dist(I_i, I_j) \geq 0$, so $dist(V_i, V_j) + dist(I_i, I_j) > 0$. Based on this, we have

$$dist(V_i, V_j) - dist(I_i, I_j) \geq 0 \Leftrightarrow dist(V_i, V_j)^2 - dist(I_i, I_j)^2 \geq 0. \tag{3}$$

According to (3), the difference between $dist(V_i, V_j)$ and $dist(I_i, I_j)$ can be analyzed as follows:

$$\begin{aligned} & dist(V_i, V_j)^2 - dist(I_i, I_j)^2 \\ & = d_i^2 + d_j^2 - 2d_i d_j \cos(\alpha_i - \alpha_j) - (d_i - r)^2 - (d_j - r)^2 \\ & \quad + 2(d_i - r)(d_j - r)\cos(\alpha_i - \alpha_j) \\ & = -2r(r - (d_i + d_j))(1 - \cos(\alpha_i - \alpha_j)). \end{aligned} \tag{4}$$

From (4), since $d_i \geq r$ and $d_j \geq r$, so $0 < r \leq (d_i + d_j)$. And due to $-1 \leq \cos(\alpha_i - \alpha_j) \leq 1$, the value of (4) is no less than zero. Thus we obtain $dist(V_i, V_j) - dist(I_i, I_j) \geq 0$, where the equal mark achieves only when $\alpha_i - \alpha_j = 0$. When $\alpha_i - \alpha_j = \pm\pi$, $dist(V_i, V_j) - dist(I_i, I_j)$ reaches its maximum value of $2r$.

Case 2: Given two disks i and j, assume that ISP I_i and ISP I_j are derived from rule 1 and rule 2, respectively, which implies that $d_i \geq r$ and $d_j < r$. The difference between $dist(V_i, V_j)$ and $dist(I_i, I_j)$ can be analyzed as follows:

$$\begin{aligned} dist(V_i, V_j) & = \sqrt{d_i^2 + d_j^2 - 2d_i d_j \cos(\alpha_i - \alpha_j)} \\ & \geq \sqrt{d_i^2 + d_j^2 - 2d_i d_j} \\ & = d_i - d_j \\ & > d_i - r = dist(I_i, I_j). \end{aligned} \tag{5}$$

Case 3: Given two disks i and j, assume that both ISP I_i and I_j are derived from rule 2, thus $dist(I_i, I_j) = 0$ and $dist(V_i, V_j) > 0$.

Summarize the above three cases together, Theorem 1 holds naturally.

Then according to the vertex permutation of route T_V, we could construct a route, T_I, from the set of ISPs. Based on Theorem 1, the route length, $|T_I|$, satisfies the following corollary:

Corollary 1. *$\forall k (0 \leq k \leq n)$, $dist(T_I[k], T_I[k+1]) \leq dist(T_V[k], T_V[k+1])$ and $|T_I| \leq |T_V|$, where the $n+1$-th vertex of each route is the start point.*

Corollary 1 indicates that T_I is a better solution than T_V for the MS routing problem.

4.3 Step 3: Concave Bend Substitute Points Choosing

After the ISP choosing step, we find that the route can still be optimized. Another observation of auto racing is as follows:

heuristic 2: *In non self-intersecting racetrack, if there exists some concave bends, the inner boundary may not be the shortest route.*

As illustrated in Fig. 4, on this segment of the racetrack, there are two convex bends and three concave bends, where the concave bends are at $\angle I_j I_k I_l$ and $\angle I_l I_m I_n$, respectively. We could choose another two better substitute points B_k and B_m for disk k and disk m. By replacing I_k and I_m with the two points, the

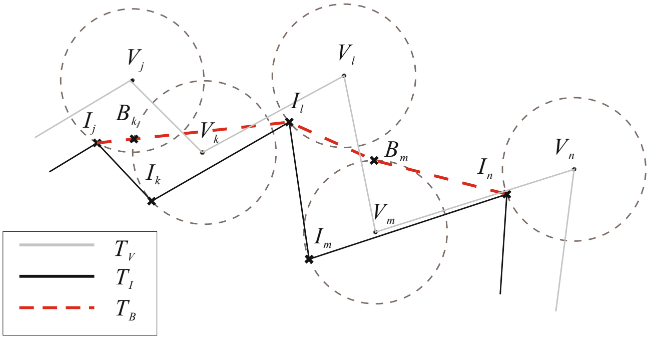


Fig. 4. Concave bends and concave bend substitute points

route length could be further optimized. We call these better substitute points *concave bend substitute point*(BSP). Assume that T_B denotes the new route formed by the BSPs.

To find these BSPs in the racetrack, we may first use a known algorithm [10] to determine the convex-concave vertices of T_I, then analyze each concave vertex. If there are concave vertices lying between two successive convex vertices[1] in T_I, we could apply nonlinear optimization techniques to calculate BSPs for these concave vertices. Assume $T_I[i]$ and $T_I[j]$ $(i+1 \leq j)$ are two successive convex vertices of T_I. This implies that there are $j-i-1$ concave vertices between $T_I[i]$ and $T_I[j]$. We use variables x_{i+1} to x_{j-1} to denote these BSPs. For tractable computation, in T_B, we reserve the corresponding ISP for each convex bend, e.g. $T_B[i] = T_I[i]$ and $T_B[j] = T_I[j]$. Therefore, an optimization problem is formulated as follows:

$$\begin{aligned} \text{minimize} : z = dist(T_I[i], x_{i+1}) + \textstyle\sum_{k=i+1}^{j-2} dist(x_k, x_{k+1}) \\ + dist(x_{j-1}, T_I[j]); \end{aligned}$$

$$s.t.\ dist(x_k, T_V[k]) \leq r \quad (i+1 \leq k \leq j-1).$$

Note that if $j-i=1$, the single BSP can be exactly obtained in the following process: by regarding the two successive convex vertices $T_I[i]$ and $T_I[j]$ as two focuses of an ellipse, thus the focal radius of this ellipse is determined as half of the length of the line segment $T_I[i]T_I[j]$. We only have to introduce one parameter variable so as to construct the equation of this ellipse. Combining the ellipse equation with the equation of the circle centered at vertex $T_V[i+1]$, we can calculate the value of the parameter variable when the equations have only one solution. Thus, we may derive $T_B[i+1]$. However, if $j-i$ becomes large, this problem is NP-hard. We design a greedy approximation algorithm to calculate all the BSPs. Let $Con[i]$ record the convexity of vertex $T_I[i]$ derived

[1] Successive convex vertices are specified two convex vertices among which there are on other convex vertices.

from a known convex-concave determining algorithm and idx denote the first concave vertex after a convex vertex. We use $Ellipse(a, b, c, r)$ to stand for the above process of the single BSP calculation, where a and b are two successive convex vertices, c is the one concave vertex between a and b, and r is the radius of radio range. The greedy algorithm is described in Fig. 5.

```
Initialize $T_B[0] \leftarrow S$; $idx \leftarrow 0$ $Con[i] \leftarrow 0$;
Determine the convexity of each vertex in $T_I$ and record the results in $Con[i]$;
**for** $i = 1$ to $n$ **do**
   **if** $Con[i] =$ convex **then**
      $T_B[i] = T_I[i]$;
      **if** $idx \neq 0$ **then**
         **for** $k = idx$ to $i - 1$ **do**
            $T_B[k] = Ellipse(T_B[k-1], T_B[i], T_V[k], r)$;
         **end for**
      **end if**
   **else if** $Con[i] =$ concave & $Con[i-1] =$ convex **then**
      $idx = i$;
   **end if**
**end for**
```

Fig. 5. Concave bend substitute points choosing algorithm

4.4 Step 4: Shortcut Finding

Due to the overlapped radio ranges, there may exist shortcuts between two BSPs. As shown in Fig. 6, the dash-dot line is a shortcut which allows the MS to access disk i, j, k and l in a straight line. We design a dynamic programming algorithm

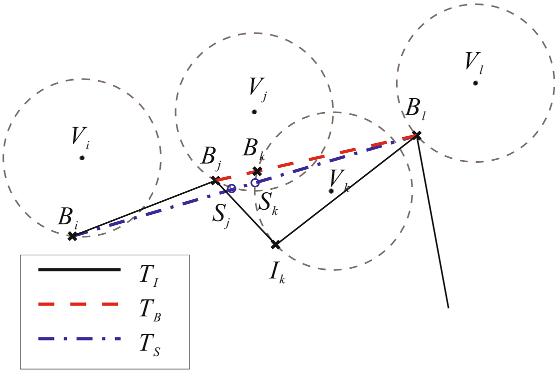


Fig. 6. Finding shortcuts

without backtracking to find out shortcuts. Assume that T_S denotes the final route after this step. $T_S[i] = 0$ means there exists a shortcut which traverses disk i and thus the point within the range of the disk i becomes meaningless. *last* stands for the latest convex vertex scanned by the algorithm. The details are described in Fig.7.

```
Initialize $T_S[0] \leftarrow S$; $i = 1$ ; $last = 0$;
while $last \leq n$ do
  for $j = i + 2$ to $n + 1$ do
    if the line segment $T_S[i]T_B[j]$ is within radio range of vertex $T_V[j-1]$ then
      $T_S[j-1] \leftarrow 0$;
      $last = j$;
    else
      $i = j - 1$;
      $T_S[i] = T_B[i]$;
      $last = j$;
      break;
    end if
  end for
end while
```

Fig. 7. Shortcut finding algorithm

5 Algorithm Analysis

In this section, we analyze the computation time and the approximation ratio of RaceTrack algorithm.

5.1 Computation Time

To analyze the computation time of RaceTrack algorithm, we should consider all its four steps. In Step 1, we handle the TSP problem by using a known PTAS [12] with the untangling improvement [4], which may have a computation time of $O(n^2)$. Since the computation time of the existing work all include the computation time of a TSP algorithm and the analysis is beyond the scope of this paper, we use C_{TSP} to denote it; In Step 2, we only have to calculate the intersecting point R_i for each disk i, thus the computation time is $O(n)$; In Step 3, we first borrow the convex-concave determining algorithm from [10], whose computation time is $O(n)$, then by traveling through all the vertices, we can compute the BPSs by using *Ellipse* function. This also consumes $O(n)$-computation time. In Step 4, although the dynamic programming algorithm has two loops, the end condition is determined by variable *last*, thus the computation time is still $O(n)$. Summarize all the above analysis, we have the optimization time of RaceTrack algorithm is $O(n)$ and the whole computation time is $C_{TSP} + O(n)$.

5.2 Approximation Ratio

Theorem 2. *The approximation ratio of RaceTrack algorithm can be described as* $|T_S| < (1+\epsilon)(|T_{MS}| + 2nr)$

Proof. We use the PTAS of Euclidean TSP[12] with the untangling improvement [4] to obtain a TSP route without self-intersecting. This PTAS achieves an approximation ratio of $1+\epsilon$, so we have:

$$|T_V| \leq (1+\varepsilon)|T_{TSP}|. \tag{6}$$

Based on the definition of the MS routing problem, the optimum solution T_{MS} is within radio range r from the center point of each disk. When the MS travels along T_{MS} and has a detour to the center point of each disk, this route is obviously an approximation solution of T_{TSP}, so

$$|T_{TSP}| < |T_{MS}| + 2nr. \tag{7}$$

According to our analysis in each step of RaceTrack algorithm, the route length may be further reduced in each step, we have

$$|T_S| \leq |T_B| \leq |T_I| \leq |T_V|. \tag{8}$$

Put Equation (6), (7) and (8) together, we proved Theorem 2.

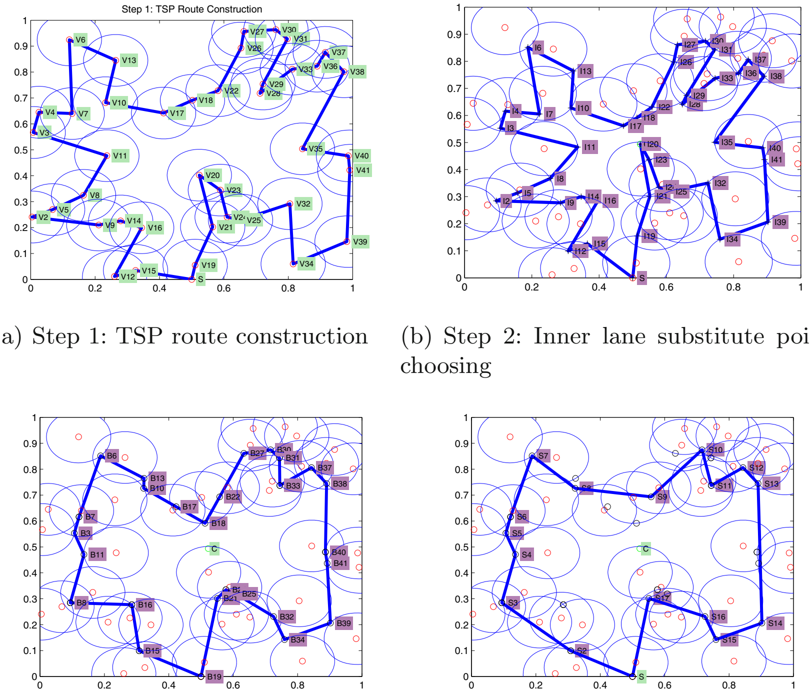
6 Performance Evaluation

In this section, we first use a case study to illustrate RaceTrack algorithm step by step, then compare it with other two algorithm in large-scale simulations.

In the case study, given a square monitoring region with size of $1000m \times 1000m$ and a fixed start point S at $(500, 0)$. We uniformly randomly deploy 40 wireless sensor nodes into this region, assume that the radio radius is $100m$. As shown in Fig. 8(a), our algorithm first constructs a TSP route without self-intersecting from the sensor nodes' locations; Fig. 8(b) shows that it then chooses ISPs for the vertices in T_V and forms route T_I; In Fig. 8(c), BSPs are chosen at each concave bend and form route T_B. Finally, in Fig. 8(d), it finds shortcuts to further cut down the route length. It is easy to see the improvement of the solution from each step.

To further verify the validity of RaceTrack algorithm, we implement two representative TSPN algorithms [Dum03][1] and [Sug08][11] in our simulations, since they both consider TSPN with possibly overlapped disk neighborhoods and others not. Comparisons between RaceTrack algorithm and these two algorithms are conducted. The metric route length is evaluated under two varied parameters: the number of sensor nodes and the radio radius.

By setting the radius a fixed value of $50m$, we first compare the route length under different number of sensor nodes. As shown in Fig. 9(a), with the number



(a) Step 1: TSP route construction

(b) Step 2: Inner lane substitute points choosing

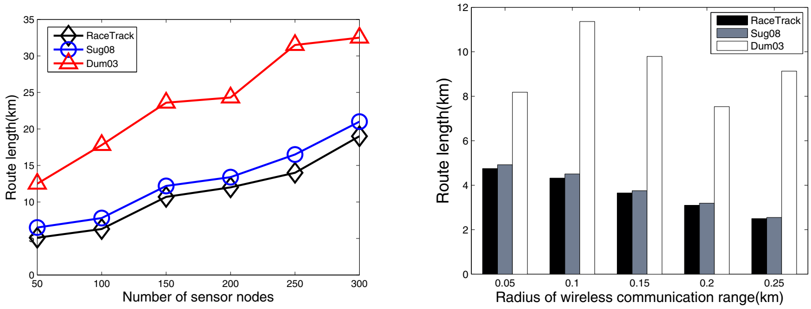
(c) Step 3: Concave bend substitute points choosing

(d) Step 4: Shortcut finding

Fig. 8. A case study for RaceTrack algorithm

of sensor nodes increasing, all the route lengths rise steadily. However, the results of [Dum03] are much higher than the other two, because they contain the perimeters of the disks in the maximum independent set. The results of [Sug08] and RaceTrack are close to each other, since both of [Sug08] and RaceTrack are TSP-anterior algorithm. The difference is that RaceTrack considers the steps of substitute points choosing, while [Sug08] not.

Then, we fix the number of sensor nodes at 40 and analyze the route length under varied radio ranges. As illustrated in Fig. 9(b), the results of [Sug08]and RaceTrack decrease as the radio radius becomes larger, since the overlapped radio ranges yield many shortcuts and the MS can access more nodes at one position. However, the results of [Dum03] fluctuate when the radius increases. A reasonable explanation is that the approximation ratio of [Dum03] is also determined by the maximum independent set algorithm. With the increase of the radius, the advantage of RaceTrack becomes smaller, this is because the playground bounded by the racetrack is shrunk, thus the route length optimization much more relies on the shortcut finding step.



(a) Number of sensor nodes VS. route length

(b) Radius VS. route length

Fig. 9. Approximation ratio comparison

7 Conclusion

RaceTrack algorithm transforms the MS routing problem of the data collection in wireless sensor networks into a auto racing problem. By using inner lane heuristic, concave bends heuristic and a shortcut finding step, it obtains an approximation shortest route in the racetrack within $O(n)$ computation time. In our future work, we may extend our algorithm with a more realistic radio model[], and consider other related factors to the energy consumption of the MS, such as, speed and direction changing. We may continue using new heuristics in auto racing to handle these extensions.

References

1. Dumitrescu, A., Mitchell, J.: Approximation algorithms for tsp with neighborhoods in the plane. Journal of Algorithms 48(1), 135–159 (2003)
2. Yuan, B., Orlowska, M., Sadiq, S.: On the optimal robot routing problem in wireless sensor networks. IEEE Trans. on Knowledge and Data Engineering 19(9), 1252–1261 (2007)
3. Arkin, E., Hassin, R.: Approximation algorithms for the geometric covering salesman problem. Discrete Applied Math. 55(3), 197–218 (1994)
4. Lawler, E., Lenstra, J., Rinnooy Kan, A., Shmoys, D.: The traveling Salesman Problem. John Wiley, Chichester (1985)
5. Tolle, G., Polastre, J., Szewczyk, R., Culler, D.: A macroscope in the redwoods. In: Proceedings of SenSys (November 2005)
6. Xing, G., Wang, T., Jia, W., Li, M.: Rendezvous design algorithms for wireless sensor networks with a mobile base station. In: Proceedings of MobiHoc (2008)
7. Gudmundsson, J., Levcopoulos, C.: A fast approximation algorithm for TSP with neighborhoods and red-blue separation. In: Asano, T., Imai, H., Lee, D.T., Nakano, S.-i., Tokuyama, T. (eds.) COCOON 1999. LNCS, vol. 1627, pp. 473–482. Springer, Heidelberg (1999)

8. Elbassioni, K., Fishkin, A., Mustafa, N., Sitters, R.: Approximation algorithms for euclidean group TSP. In: Proceedings of ICALP (2005)
9. Gupta, P., Kumar, P.: Critical power for asymptotic connectivity in wireless networks. In: A Volume in Honor of Fleming, W.H., McEneaney, W.M., Yin, G., Zhang, Q. (eds.) Stochastic Analysis, Control, Optimization and Applications (1998)
10. Liu, R.: A simple and fast algorithm for detecting the convexity and concavity of vertices for an arbitrary polygon. Journal of Software 13(7), 1309–1312 (2002)
11. Sugihara, R., Gupta, R.: Improving the data delivery latency in sensor networks with controlled mobility. In: Proceedings of DCOSS (2008)
12. Arora, S.: Nearly linear time approximation schemes for euclidean tsp and other geometric problems. Journal of ACM 45(5), 1–30 (1998)
13. Gandhi, S., Suri, S., Welzl, E.: Catching elephants with mice: sparse sampling for monitoring sensor networks. In: Proceedings of SenSys (November 2007)
14. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of Sensys (2003)
15. Gu, Y., Bozdag, D., Brewer, R.W., Ekici, E.: Critical power for asymptotic connectivity in wireless networks. Journal of Computer Networks 50, 3449–3465 (2006)

Data Aggregation Integrity Based on Homomorphic Primitives in Sensor Networks

Zhijun Li and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
leezj@engmail.uwaterloo.ca, ggong@calliope.uwaterloo.ca

Abstract. Designing message integrity schemes for data aggregation is an imperative problem for securing wireless sensor networks. In this paper, we propose three secure aggregation schemes that provide provably secure message integrity with different trade-offs between computation cost, communication payload, and security assumptions. The first one is a homomorphic MAC, which is a purely symmetric approach, and is the most computation- and communication-efficient, but requires all data-collecting nodes to share one global key with the base station. The other two make use of (public key based) homomorphic hashing, combined with aggregate MAC and identity-based aggregate signature (IBAS) respectively. The scheme with aggregate MAC allows the base station to share a distinct key with every node, while the scheme with a paring-based IBAS enables all intermediate nodes beside the base station to verify the authenticity of aggregated messages.

1 Introduction

From the very beginning of wireless sensor networks (WSNs) development [13,2,30,14,3], it has been widely accepted that in-network data aggregation plays a critical role in the practicability and appealing of WSNs. In a typical sensor network, hundreds and thousands of low-cost sensor nodes scatter in a targeted area, collect environmental information, and collaboratively transmit data back to a base station. In many cases, users of sensor network applications are only interested in aggregated results after in-network processing, rather than detailed readings from individual nodes. On the other hand, data aggregation during message transmission is a natural way of preserving sensor nodes precious energy. Due to infeasibility of recharging nodes batteries in most circumstances, energy becomes the most valuable resource for sensor nodes. Among all nodes operations, data transmission consumes the most energy [2,3]. Moreover, in the absence of data aggregation, sensor nodes near the base station will suffer from heavy message transmission overhead, and then die of power exhaustion much sooner than other nodes, breaking down the whole network's functionality. Subsequently, data aggregation attracted a great deal of attention and many a data aggregation scheme has been proposed in recent years. Interested readers may refer to [30,14] for systematic surveys on this topic.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 149–162, 2010.
© Springer-Verlag Berlin Heidelberg 2010

When sensor nodes are deployed in a hostile environment, security measurements should be taken into consideration for network protocols. Attacks to wireless sensor networks not only come from outsider adversaries, but also can be conducted by compromised, previously legitimate nodes. Thus applicable secure protocols should prevent malicious inside nodes from damaging the whole network's functionality, or at least constrain their impacts to a reasonable level. Unfortunately, data aggregation, which requires intermediate nodes to process and change messages, and security objectives, one of which is preventing malicious manipulation, conflict with each other in this regard. As a result, designing secure and practical data aggregation schemes, which are critical to many sensor network applications, imposes an interesting and formidable challenge.

Resembling general security cases in other fields, message integrity might be one of the most important security objectives in sensor networks, and it should addressed by specific protocols. Generally, there are three kinds of message verification approaches for data aggregation: retroactive detection, abnormality-based detection, and cryptographic integrity primitives. Generally speaking, retroactive detection approaches, which involve substantial communication/interaction among the base station and sensor nodes to verify messages integrity, are not satisfactory because their costly performance penalty directly violates the intent of data aggregation. One may argue that the predictable data distribution can be used as a gauge to analyze and detect the abnormality of aggregated results; but the false rates are generally too high to be practical, and thus it is not a dependable solution. Consequently, schemes based on solid cryptographic primitives are usually desirable. Unfortunately, conventional cryptographic integrity primitives, such as message authentication code (MAC) and signature, are not compatible with data aggregation scenarios.

Contributions. Based on new cryptographic homomorphic primitives [1,24,17], we propose three secure aggregation schemes that provide provably secure message integrity. The first one is a homomorphic MAC scheme for data aggregation, which is a revised version of the homomorphic MAC proposal on secure network coding application in [1]. This homomorphic MAC scheme, other than revisions to fit data aggregation scenarios, achieves a little bit performance improvement, as we observe and then remove an unnecessary step in the original scheme. The homomorphic MAC scheme is computation- and communication-efficient, but with one inherent restriction: all data-collecting nodes share one global key with the base station. The assumption that all those nodes are tamper-proof might be too strong to be realistic in many sensor network applications. In order to overcome this drawback, we further propose two secure aggregation schemes based on homomorphic hashing [24,17], at the expense of increasing communication and computation costs. One is to combine homomorphic hashing with aggregate MAC [23], in which every node shares a different key with the base station, while the other is with identity-based aggregated signature [19], which enables intermediate nodes to verify the authenticity of messages. The proposed three protocols present different trade-offs between computation, communication, security and can fit a wide variety of application areas.

Organization. The remainder of the paper is organized as follows. The related work is introduced in Section 2. Then we state the data aggregation network setting along with the security objective, discuss homomorphic primitives and define homomorphic MAC as well as homomorphic hashing in Section 3. Afterward, secure aggregation integrity schemes based on homomorphic MAC and homomorphic hashing are proposed and discussed in Section 4 and Section 5 respectively. Finally, Section 6 concludes the paper.

2 Related Work

Hu and Evans [20] described a secure hop-by-hop data aggregation scheme, in which every node shares with the base station a different key, from which temporary session MAC keys will be derived, and by adopting hash-chain-based delayed message authentication, such as μTESLA [27], intermediate modes, after the base station reveals session MAC keys, will be able to verify the integrity of messages that they buffered. This scheme suffers from communication penalties, as the introduction of μTESLA for distributing session MAC keys incurs considerable communication cost. More disturbingly, in order to detect one inside malicious node that manipulates other nodes input, intermediate nodes have to obtain and buffer all their grandchildren's messages and corresponding MACs, that is, two-hop messages buffer only being able to detect *one* misbehavior node. Although Jadia and Muthuria [21] extended the Hu-Evans scheme by all two nodes in the two-hop communication range sharing pairwise keys and then the scheme eliminates the usage of μTESLA, the fact that both schemes are only capable of preventing a single inside malicious node at an appreciable communication cost makes them impractical.

Yang *et al.* [32] presented a secure hop-by-hop data aggregation protocol for sensor networks named SDAP, using the principles of divide-and-conquer and commit-and-attest, which is a typical example of retroactive detection approach. In SDAP, a probabilistic grouping technique is utilized to dynamically partition the nodes in a tree topology into subtrees. A commitment-based hop-by-hop aggregation is conducted in each subtree to generate a group aggregate, and accordingly the base station identifies the suspicious subtrees based on the set of group aggregates. Finally, each subtree under suspect participates in an attestation procedure to prove the correctness of its group aggregate. Those complicated algorithms cause significant transmission overhead, and may cancel off all communication benefits from data aggregation.

Przydatek, Song, and Perrig [29] proposed secure information aggregation (SIA) to identify forged aggregation values from malicious nodes. In the SIA scheme, a special node named aggregator computes an aggregation result over raw data together with a commitment to the data based on a Merkle-hash tree and sends them back to a remote user, which later challenges the aggregator to verify the aggregation. Later Chan, Perrig, and Song built on the aggregate-commit-prove framework in [29] but extended their single aggregator model to a fully distributed setting. Frikken and Dougherty [16] further improved the

Chan-Perrig-Song scheme. Moreover, Chan and Perrig [10] derived several security primitives form this kind of algorithms.

3 Preliminary

3.1 Network Setting and Security Objective

We consider a sensor network that consists of n sensor nodes which are highly sensitive of energy consumption, and a base station that is only concerned about the statistical results, mainly mean and variance. Thus a data aggregation mechanism is implemented in the sensor network.

Since loose time synchronization among sensor nodes is indispensable for efficient message aggregation and the sensor network is under attacks, it is assumed that there is a secure time synchronization scheme [28] available in the sensor network. We do not explore a specific secure time synchronization selection because it is independent and relatively irrelevant. At a designated time, the sensor network outputs a *report*, which is an overall aggregated result for a task and is *uniquely* identified by a report identifier rid. The report identifier may be the task description combined with the reporting time. It is clear that all sensor nodes should have an agreement on the report identifier specification and know how to correctly generate rid. Otherwise, nodes cannot distinguish messages of different kinds and data aggregation cannot be properly performed.

There are three kinds of roles in the sensor network: a *contributor* that collects environmental readings and generate a *raw message*, an *aggregator* that aggregates all messages that it received plus possibly its own raw message and then produces an *aggregated message*, an *verifier* that verifies the authentication of messages it received. A node may play some of or all the three roles, while the base station is definitely a verifier.

The data are aggregated though the network, and the base station eventually retrieves an aggregated result, i.e. the report. In order to produce the mean of a measurement, it suffices for the base station to retrieve the sum of the samples and the number of contributors. If the variance is desired, the contributors should also provide the squares of their readings and the aggregators accordingly merge the squares. By the mean, the number of contributors and the sum of the squares, one can readily calculate the variance as a basic statistical equation. In other words, we only need to consider an additive aggregation. For the sake of simplicity, we assume that the sensor network is organized as a tree structure rooted on the base station, though our proposed schemes fit into any kind of data additive aggregation architecture.

In addition, to support advanced aggregation requirements, the concept of *weight* is introduced. Specifically, we allow that the measurements of different nodes have different weights for their contributions to the final report. In most cases, node weights are uniform; when different weights are required, we assume that aggregators and the base station are aware of the weights of messages contributors, either via an established agreement, or from explicit indications attached to messages.

The primary objective of our proposals is to provide the message integrity for data aggregation in a cryptographic manner, thus an authentication segment that facilitates verification shall be appended to a message. Generally speaking, it is impossible for a verifier to validate the integrity of an aggregated message without the knowledge of its contributors. This is because if contributors use different keys, the verifier certainly needs knows who those contributors are before using those keys in the verification stage; if a global key is employed and a verifier cannot retrieve contributors of messages, an adversary may easily construct a malicious message to pass the integrity verification by aggregating a single message from one contributor many times, say b times, which is indistinguishable with an aggregated message resulting from b legitimate contributors. In other words, data origin authentication is an inherent requirement for data aggregation integrity.

The simplest way of indicating data origin is to attach the list of contributors to a message. To avoid the communication cost in this approach, we may utilize a mechanism that allows a verifier to implicitly obtain the contributor list, such as derivation from the network topology. This is pretty realistic for the base station as the ultimate verifier. In a case that a verifier is capable of identify all potential contributors, of which only a small fraction do not really participate in a message contribution, a list of exclusive nodes rather than the contributors may be appended to the message. Due to the space limitation, we do not elaborate the techniques of efficiently providing the contributor list for verifiers. Henceforth, we simply assume that an aggregator knows the appropriate weights to aggregate messages, and a verifier of a message can obtain its contributors and corresponding weights. When we discuss a scheme's communication cost, we do not consider the payload from contributor lists and weights, because, as we argued, there might be mechanisms to avoid it, or it is inevitable for message authentication.

As a typical application scenario of this network setting, a sensor network is employed to routinely detect environmental information, such as temperature, humidity, radiation. Every node senses data in a hourly interval, and submits the results on a daily basis. For example, at two o'clock every day, starting from all leaf nodes, messages are transmitted and aggregated over a spanning tree.

3.2 Homomorphic Primitives

Homomorphic property in cryptographic operations may be very useful in a variety of applications, and thus stimulates research on homomorphic primitives, namely homomorphic encryption, homomorphic MAC, homomorphic hashing, and homomorphic signature. Homomorphic encryption [15], in which a user without a decryption key can perform algebraic operations on ciphertext to achieve designated operation results on the corresponding plaintext, has been studied for decades, and recently, an outstanding result, fully homomorphic encryption [18], was proposed, which allows arbitrary operations on ciphertext (and so on plaintext). Even though the only two fully homomorphic encryption schemes [18,12] by now have not provided competitive performance for most applications, they do reveal a perspective on a powerful, widely demanded technique and we expect that practical schemes will eventually emerge. Those homomorphic encryption schemes shall provide a solid foundation for data confidentiality of aggregated messages. As for homomorphic signature, current schemes [22,8] are mainly aimed at one-sender many-recipients secure multi-cast scenarios, with costly computation overhead (compare to symmetric primitives), thus they may not be suitable for secure data aggregation integrity of WSNs. In contrast, homomorphic MAC and homomorphic hashing can be effectively used to construct message integrity schemes of supporting additive aggregation with weights. Formally, homomorphic MAC and homomorphic hashing are defined as follows.

Definition 1 (Homomorphic MAC [1]). *A homomorphic MAC should satisfy the following properties:*

1. Homomorphism. *Given two (message, tag) pairs $(\boldsymbol{m}_1, t_1)$ and $(\boldsymbol{m}_2, t_2)$, anyone can create a valid tag t_a for an aggregated message $\boldsymbol{m}_a = w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ for any scales w_1, w_2 as weights. Typically, $t_a = w_1 t_1 + w_2 t_2$.*
2. Security against Chosen Message Attack. *Even under a chosen message attack, in which an adversary is allowed to query tags of polynomial number of messages, it is still infeasible for the adversary to create a valid tag for a message other than an linear combination of some previously queried messages.*

A homomorphic MAC consists of three probabilistic, polynomial-time algorithms (Sign*,* $\mathsf{Aggregate}$*,* Verify*)*

- $t_u = \mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$*: node u with ID* id_u*, as a contributor of a raw message $\boldsymbol{m}_u$ regarding report* rid*, computes a tag t_u for $\boldsymbol{m}_u$ using k as the key.*
- $t = \mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$*: an aggregator implements the homomorphic property for message-tag pairs in the absence of key k, that is, generates a tag t for the aggregated message $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i$*
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$*: a verifier verifies the integrity of message $\boldsymbol{m}$ regarding report* rid *by key k and tag t.*

The homomorphic MAC scheme is first defined and proposed in [1], intended to provide secure network coding. The definition above is equivalent to that in [1], with emphasis on the data aggregation.

Definition 2 (Homomorphic Hashing [24,17]). *A homomorphic hash function H is a hash function satisfying:*

1. Homomorphism. *For any two messages $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and scalars w_1, w_2, it holds that $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$.* [1]

[1] Intuitively, the homomorphic equation should be $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = w_1 H(\boldsymbol{m}_1) + w_2 H(\boldsymbol{m}_2)$. In fact, that just uses a different notation on group operation and essentially they are equivalent.

2. Collision Resistance. *There is no probabilistic polynomial-time (PPT) adversary capable of forging* $(\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, w_1, w_2)$ *satisfying both* $\boldsymbol{m}_3 \neq w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ *and* $H(\boldsymbol{m}_3) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$.

The homomorphic hashing can be used in many applications, such as secure network coding [17], secure peer-to-peer content distribution using erasure codes [24].

4 Secure Aggregation with Homomorphic MAC

Our first proposal is a specific homomorphic MAC scheme that fully complies with Definition 1. Basically, the scheme is a revised version of the homomorphic MAC for network coding proposed by Agrawal and Boneh (AB Scheme) [1].

4.1 Scheme Description

To formally present our schemes, message $\boldsymbol{m}$ is formed as d segments of l bits. Let $q = 2^l$, then the message space is $\mathbb{F}_q^d$. In other words, message $\boldsymbol{m}$ can be represented as a vector of d segments: $(m_1, m_2, \cdots, m_d)$, where $m_i \in \mathbb{F}_q, i = 1, 2, \cdots, d$. As the additive operation is over finite field $\mathbb{F}_q$, q should be greater than the bound of the desired data sum. We stress that this is also an inherent requirement in the data aggregation.

To generate and verify tags, all contributors and verifiers share one global MAC key that consists of (k_1, k_2). Naturally, those nodes should be tamper-proof to protect the protocol security. Let $\mathcal{K}_1$ and $\mathcal{K}_2$ denote the key spaces of k_1 and k_2 respectively, $\mathcal{I}$ denote the space of node identities, and $\mathcal{R}$ denote the space of report identifiers. Two pseudo random functions are required: $R_1 : \mathcal{K}_1 \rightarrow \mathbb{F}_q^d$ and $R_2 : (\mathcal{K}_2 \times \mathcal{R} \times \mathcal{I}) \rightarrow \mathbb{F}_q$.

The three algorithms (Sign, Aggregate, Verify) are given as follows.

- $\mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$, by node u as a contributor
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b_u = R_2(k_2, \mathsf{rid}, \mathsf{id}_u) \in \mathbb{F}_q$.
  3. $t_u = \boldsymbol{a} \circ \boldsymbol{m}_u + b_u \in \mathbb{F}_q$, where $\circ$ stands for the inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{m}_i$ over finite field $\mathbb{F}_q$, that is, $\boldsymbol{a} \circ \boldsymbol{m}_u$ is equal to $a_1 m_{u,1} + a_2 m_{u,2} + \cdots + a_d m_{u,d} \mod q$.
- $\mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$, by an aggregator
  1. $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i \in \mathbb{F}_q^d$, in which the additive operation is over $\mathbb{F}_q$.
  2. $t = \sum_{i=1}^{j} w_i t_i \in \mathbb{F}_q$.
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$, by a verifier with the knowledge of contributor identities and weights
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b = \sum_{i=1}^{j} [w_i \cdot R_2(k_2, \mathsf{rid}, \mathsf{id}_i)] \in \mathbb{F}_q$.
  3. if $\boldsymbol{a} \circ \boldsymbol{m} + b = t$ outputs "ACCEPT"; otherwise outputs "REJECT".

4.2 Discussion and Comparison

By the same reductionist proof of Theorem 2 in [1], this scheme is probably secure against chosen message attack based on the pseudo-randomness of R_1 and R_2. Since the tag size is l-bit, in order to achieve 80-bit security level, l should not be less than 80.

To support secure network coding, the space $\mathcal{I}$ in the AB scheme [1] is $\mathbb{F}_q^c$, albeit id_i is a vector base identifier, rather than a node id, and c is the number of vector base. Since every message in the network coding should include a vector in $\mathbb{F}_q^c$ to indicate the combination coefficients of c vector bases, which are analogue to weights in the data aggregation, usually $q = 2^8$ is recommended (as in the AB Scheme) to save communication cost while maintaining high success decoding probability for random network coding. Such a small q, however, undermines the security level, as the tag size would be 8-bit and an adversary can fake a message's tag at least with probability 1/256. Fortunately, the data aggregation does not suffer that limitation—the weights are not randomly chosen by aggregators. Therefore, we can safely use $q \geq 2^{80}$.

In addition, the AB homomorphic MAC scheme specifies $R_1 : \mathcal{K}_1 \rightarrow \mathbb{F}_q^{d+c}$, $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^{d+c}$, and then $t_u = \boldsymbol{a} \circ (\boldsymbol{m}_u || \mathsf{id}_u) + b$. We observe that the occurrence of id_u in $(\boldsymbol{a} \circ (\boldsymbol{m}_i || \mathsf{id}_u))$ is unnecessary and then it is removed in our revision because id_u has been used in the computation of $b = R_2(k_2, \mathsf{rid}, \mathsf{id}_u)$. This modification slightly improves the computation performance and can apply to both network coding and data aggregation scenarios.

We notice that our proposed scheme has a similar structure to the data aggregation MAC scheme proposed by Castelluccia *et al.* (CCMT scheme) [9]. In their scheme, the space of message $\boldsymbol{m}$ is limited to $\mathbb{F}_q$, which means that the tag is as long as the maximal length of messages. This approach violates a principle on MAC that a MAC scheme should support arbitrary length of message and output short, fixed length of tags. Admittedly, the length of messages in our scheme has to be determined beforehand, but it is a basic requirement for data aggregation. In addition, the CCMT scheme does not supply a reductionist security proof; Theory 2 in [9] pertaining to the scheme security is more like an argument than a proof. Nonetheless, the CCMT scheme provides a necessary integrity scheme for data aggregation, and our homomorphic MAC scheme can be thought as the combination of the CCMT scheme and the AB scheme.

The security of the proposed MAC scheme relies on the pseudo-randomness of R_1 and R_2. In principle, all provably secure pseudo-random generators are public-key based[2], involving heavy computation. As a widely employed method, we may use AES [11] to implement R_1 and R_2. In this way, the proposed scheme is very computationally efficient, and the key lengths of k_1 and k_2 are 128-bit. On the other hand, a 80-bit tag would suffice to allow a verifier to check the authenticity of an aggregated message, which presents the optimal communication overhead. One inherent drawback in homomorphic MACs is that one

[2] A public-key based approach does not necessarily indicate that it involves public/private keys; instead, it implies that the approach employs typical public-key cryptosystem operations, such as exponentiation over a big group.

single MAC key is shared by all contributors and verifiers. If sensor nodes are not tamper-proof and one of them is compromised by an adversary, the whole system security is breached.

5 Two Schemes Based on Homomorphic Hashing

In order to overcome the drawback of one global MAC key in the previous scheme, we propose two schemes based on homomorphic hashing.

5.1 Construction of Homomorphic Hashing

The first step is to find a homomorphic hashing function suitable for sensor networks. At present, there are only two homomorphic hashing functions: one is based on the hardness of discrete logarithm [24], and the other is based on the intractability of integer factorization [17].

Discrete Logarithm [24]. Let $\mathbb{G}$ be a cyclic group of prime order p in which the discrete logarithm problem is hard, and the public parameters contain a description of $\mathbb{G}$ and d random generators $g_1, g_2. \cdots, g_d \in \mathbb{G}$. Then a homomorphic hashing on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}_p^d$ can be constructed by

$$H(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i}. \tag{1}$$

It is easy to verify that the homomorphic property is satisfied in this construction, and the collision resistance is guaranteed by the hardness of the discrete logarithm problem in $\mathbb{G}$.

Integer Factorization [17]. Let N be the product of two safe primes[3] so that the group $\mathbb{Q}_N$ of quadratic residues modulo n is cyclic, and let $g_1, g_2, \cdots, g_d$ be generators of $\mathbb{Q}_N$. Then a homomorphic hashing on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}_N^d$ can be constructed by

$$H_N(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i} \mod N. \tag{2}$$

Finding a collision is computationally equivalent to factoring N, which is intractable.

Comparison. The homomorphic hashing function (2) can use the form of $H_N(\boldsymbol{m}) = 2^{\boldsymbol{m}} \mod N$ by choosing a proper N such that 2 is a generator of $\mathbb{Q}_N$ and the integer value converted from any message $\boldsymbol{m}$ is less than N. Subsequently, it presents some computational advantage over hashing function (1) by

[3] A prime number p is a safe prime if $(p-1)/2$ is also a prime.

fast exponentiation. However, then its hash value size, which is the same as the size of N, exceeds the message size. This is unacceptable in the data aggregation of sensor networks. Even for the basic form (2), in order to provide 80-bit security, N is at least 1024-bit, while by using elliptic curve cryptography (ECC), the hash value size of function (1) can be approximately as low as 160-bit. Moreover, the practicability of implementing ECC in low-cost sensor nodes has been successfully demonstrated in [25,31]. Therefore, the suitable homomorphic hashing for secure data aggregation integrity in WSNs should be function (1).

5.2 Aggregation Integrity by Homomorphic Hashing

Since we choose the homomorphic hashing function (1), the message space is $\mathbb{F}_p^d$, where p is a prime number and $p \geq 2^{160}$ for 80-bit security. For a raw message $\boldsymbol{m}_i$, node i computes a raw hash value $h_i = H(\boldsymbol{m}_i)$, and uses a mechanism to sign h_i, which will be specified later, in a way that allows verifiers to verify the authenticity of h_i. When a verifier receives an aggregated message $\boldsymbol{m} = \sum_{i=1}^{j}(w_j \boldsymbol{m}_i)$ along with j pairs of (raw hashing value, weight) (h_i, w_i), it first determines whether the hashing values are valid, and then verifies the message's integrity by checking whether

$$\prod_{i=1}^{j} h_i^{w_i} \stackrel{?}{=} H(\boldsymbol{m}).$$

This scheme is proven secure in the standard model via reductionist from the discrete logarithm problem [24,8], when raw hash values are authenticated by a secure mechanism. In the following two subsections, we describe two communication-efficient mechanisms to authenticate h_i.

5.3 Authentication by Aggregate MAC

Aggregate MAC [23] presents the property that multiple MAC tags, computed by different contributors on multiple raw hash values, can be aggregated into a single tag that can be verified by a verifier who shares a distinct key with each contributor. The construction of aggregate MAC has been long known. In fact, an aggregate MAC which is provably secure [23] can be constructed from essentially any standard message authentication code as follows.

For simplicity, we assume that the base station is the sole verifier. Let k_i be the symmetric key shared by node i and the base station, Mac be a standard deterministic MAC, for example: CBC-MAC [5], HMAC [4]. To authenticate a raw hash value h_i, node i generates a tag: $t_i = \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i)$. Any aggregator can aggregate j tags by simply computing the XOR of all the tag values: $t = \bigoplus_{i=1}^{j} t_i$. Then the base station uses the aggregate tag t to verify the authenticity of all raw hash values by checking whether

$$t \stackrel{?}{=} \bigoplus_{i=1}^{j} \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i).$$

5.4 Authentication by Identity-Based Aggregate Signature

Aggregate MACs, like all other symmetric-key MACs, demand verifiers to comprehend contributors keys. In many circumstances, it would be much appreciated that all intermediate nodes can verify the authenticity of raw hash values (and then aggregated messages). In terms of communication cost, the best scheme providing such a property is an identity-based aggregate signature (IBAS), in which different raw hash values produced by many different contributors, whose public keys are their identities, can be authenticated by one single aggregate signature.

As far as we know, there are three IBAS schemes which are provably secure: GR scheme [19], BN scheme [6], and BGOY scheme [7]. The BN scheme [6] requires interactions of all signers, and the BGOY scheme demands a sequential signature aggregation procedure; thus both are not suitable for secure aggregation in WSNs. One presumably too strong assumption in the GR scheme [6] is that all signers must use a same unique string when signing, which, fortunately, is not a problem at all in the WSN secure aggregation application, because an unique rid for every report is known to all nodes.

GR Paring-Based IBAS Scheme [19]. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of some large prime order q that efficiently support a bilinear mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. That is, $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$. The GR IBAS scheme works as follows.

- *Setup*: To set up the scheme, a private key generator (PKG)
  1. generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
  2. chooses an arbitrary generator $P \in \mathbb{G}_1$.
  3. picks a random $s \in \mathbb{Z}/q\mathbb{Z}$ as the master key of PKG and sets $Q = sP$.
  4. chooses three cryptographic hash functions $H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$ and $H_3 : \{0,1\}^* \to \mathbb{Z}/q\mathbb{Z}$.
- *Private key generation*: Node i receives from the PKG the values of $sP_{i,\alpha}$ as its private key for $\alpha \in \{0,1\}$, where $P_{i,\alpha} = H_1(\mathsf{id}_i, \alpha) \in \mathbb{G}_1$.
- *Signing*: To sign $\boldsymbol{h}_i$, node i
  1. computes $P_{\mathsf{rid}} = H_2(\mathsf{rid}) \in \mathbb{G}_1$.
  2. computes $c_i = H_3(h_i, \mathsf{id}_i, \mathsf{rid}) \in \mathbb{Z}/q\mathbb{Z}$.
  3. generates random $r_i \in \mathbb{Z}/q\mathbb{Z}$.
  4. computes signature (S_i, T_i), where $S_i = r_i P_{\mathsf{rid}} + sP_{i,0} + c_i sP_{i,1}$ and $T_i = r_i P$.
- *Signature Aggregation*: Signatures (S_i, T_i) for $1 \le i \le j$ can be aggregated into (S, T), where $S = \sum_{i=1}^{j} S_i$, and $T = \sum_{i=1}^{j} T_i$.
- *Verification*: Any node can verify the signature by checking whether

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}(T, P_{\mathsf{rid}})\ \hat{e}(Q, \sum_{i=0}^{j} P_{i,0} + \sum_{i=0}^{j} c_i P_{i,1}).$$

This scheme is proven secure in the random oracle model, on the assumption of hardness of computational Diffie-Hellman problem.

Generally speaking, paring is a highly computation-intense operation and more costly than ordinary public key based operations. Consider the fact that identity-based schemes eliminate the cost of transmitting nodes public keys and most of practical identity-based encryptions are paring-based, the use of the GR paring-based IBAS scheme in the secure WSN data aggregation is justifiable. In addition, TinyPBC [26] which implements and measures paring operations give an affirmative answer to the question of whether paring is feasible in the WSNs, albeit their paring implementation is understandably slow.

5.5 Discussion

To verify the integrity of an aggregated message, a verifier should retrieve the raw hash values of the contributors, which constitutes considerate communication payload and is an instinctive downside for homomorphic-hashing-based approaches. When the message size in a application does not exceed the homomorphic hashing result size (160-bit typically), then the homomorphic hashing is redundant, and directly applying aggregate MAC/signature to raw messages is preferred. If the message size is substantially greater than the hashing value size, which is quite common for WSNs, then using homomorphic hashing would significantly reduce the communication cost, as in the application scenario described in Section 3.1.

For 80-bit security, the signature of the GR scheme is roughly 320-bit, while a typical aggregate MAC tag is 80-bit. The third scheme, which combines homomorphic hashing function (1) with the GR identity-based aggregate signature, provides the most promising security for data aggregation integrity. Since all intermediate nodes are capable of verifying the integrity of (raw or aggregated) messages in that scheme, a node (or an adversary) that tries to inject invalid messages into the sensor network can be easily caught. It is worth to notice that this is achieved at the computational cost of paring operation by intermediate nodes. By contrast, the second scheme (with aggregate MAC) does not require paring operation and is useful in practice. If detecting an invalid (message, tag) pair in the second scheme, the base station can require the corresponding child to submit its aggregation record, and then interacts with grandchildren until reaching leaf nodes. In this way, the base station can determine which nodes should be responsible for faking messages and then expels them from the network.

6 Conclusion

In this paper, we present three secure aggregation schemes that provide provably secure message integrity with different trade-offs between computation cost, communication payload, and security assumptions. The first proposal is a concrete homomorphic MAC scheme for WSN data aggregation integrity, and the other two are combining homomorphic hashing with aggregate MAC and identity-based aggregate signature respectively. We detail on the selections and constructions of those three cryptographic primitives and discuss their practicability on wireless sensor networks.

Acknowledgment

The research is supported by the NSERC Strategic Project Grants.

References

1. Agrawal, S., Boneh, D.: Homomorphic MACs: MAC-Based Integrity for Network Coding. In: ACNS 2009. LNCS, vol. 5536, pp. 292–305. Springer, Heidelberg (2009)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
3. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Hu, Y.F.: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications 30(7), 1655–1695 (2007)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
5. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences 61(3), 362–399 (2000)
6. Bellare, M., Neven, G.: Identity-Based Multi-signatures from RSA. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 145–162. Springer, Heidelberg (2006)
7. Boldyreva, A., Gentry, C., O'Neill, A., Yum, D.H.: Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 276–285. ACM, Alexandria (2007)
8. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a Linear Subspace: Signature Schemes for Network Coding. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 68–87. Springer, Heidelberg (2009)
9. Castelluccia, C., Chan, A.C.F., Mykletun, E., Tsudik, G.: Efficient and provably secure aggregation of encrypted data in wireless sensor networks. ACM Trans. Sen. Netw. 5(3), 1–36 (2009)
10. Chan, H., Perrig, A.: Efficient security primitives derived from a secure aggregation algorithm. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. ACM, Alexandria (2008)
11. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
12. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
13. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next Century Challenges: Scalable Coordination in Sensor Networks. In: Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263–270. IEEE Computer Society, Seattle (1999)
14. Fasolo, E., Rossi, M., Widmer, J., Zorzi, M.: In-network aggregation techniques for wireless sensor networks: a survey. IEEE Wireless Communications 14(2), 70–87 (2007)
15. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security 2007(1), 1–15 (2007)

16. Frikken, K.B., Dougherty IV, J.A.: An efficient integrity-preserving scheme for hierarchical sensor aggregation. In: Proceedings of the first ACM Conference on Wireless Network Security, pp. 68–76. ACM, Alexandria (2008)
17. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure Network Coding Over the Integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 142–160. Springer, Heidelberg (2010)
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 169–178. ACM, Bethesda (2009)
19. Gentry, C., Ramzan, Z.: Identity-Based Aggregate Signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 257–273. Springer, Heidelberg (2006)
20. Hu, L., Evans, D.: Secure aggregation for wireless networks. In: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT 2003 Workshops), pp. 384–391 (2003)
21. Jadia, P., Mathuria, A.: Efficient Secure Aggregation in Sensor Networks. In: Bougé, L., Prasanna, V.K. (eds.) HiPC 2004. LNCS, vol. 3296, pp. 40–49. Springer, Heidelberg (2004)
22. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic Signature Schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–245. Springer, Heidelberg (2002)
23. Katz, J., Lindell, A.: Aggregate Message Authentication Codes. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 155–169. Springer, Heidelberg (2008)
24. Krohn, M.N., Freedman, M.J., Mazières, D.: On-the-fly verification of rateless erasure codes for efficient content distribution. In: IEEE Symposium on Security and Privacy 2004, pp. 226–240 (2004)
25. Liu, A., Ning, P.: TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In: International Conference on Information Processing in Sensor Networks (IPSN 2008), pp. 245–256 (2008)
26. Oliveira, L.B., Scott, M., Lopez, J., Dahab, R.: TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. In: 5th International Conference on Networked Sensing Systems, INSS 2008, pp. 173–180 (2008)
27. Perrig, A., Szewczyk, R., Culler, V.W.D., Tygar, J.D.: SPINS: Security protocols for sensor networks. In: Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM), pp. 189–199. IEEE, Rome (2001)
28. Poovendran, R., Wang, C., Roy, S.: Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks. Springer, Heidelberg (2007)
29. Przydatek, B., Song, D., Perrig, A.: SIA: Secure Information Aggregation in Sensor Networks. In: Proceedings of the First International Conference on Embedded Networked Sensor Systems, Los Angeles, California, USA, pp. 255–265 (2003)
30. Rajagopalan, R., Varshney, P.K.: Data-aggregation techniques in sensor networks: a survey. IEEE Communications Surveys & Tutorials 8(4), 48–63 (2006)
31. Szczechowiak, P., Oliveira, L., Scott, M., Collier, M., Dahab, R.: NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 305–320. Springer, Heidelberg (2008)
32. Yang, Y., Wang, X., Zhu, S., Cao, G.: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks. In: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 356–367 (2006)

iCCA-MAP versus MCL and Dual MCL: Comparison of Mobile Node Localization Algorithms

Shafagh Alikhani, Thomas Kunz, and Marc St-Hilaire

Department of Systems and Computer Engineering, Carleton University, Canada
{shafagha,tkunz}@sce.carleton.ca, marc_st_hilaire@carleton.ca

Abstract. Accurately locating a moving node in a wireless sensor network, in real time, is a difficult yet essential process. In this paper, we compare the localization performance of different mobile node localization algorithms: iCCA-MAP, MCL, and Dual MCL. The localization errors as well as the effect of increasing the percentage of anchor nodes and varying the speed of the mobile node in the network are compared. iCCA-MAP applies an iterative and efficient nonlinear data mapping technique in order to localize the position of a mobile node within a wireless sensor network. MCL and Dual MCL, which is the logical inverse of MCL, use particle filtering combined with probabilistic models of robot perception and motion. Simulation results show that iCCA-MAP outperforms MCL and Dual MCL by having a lower localization error with the minimum number of anchor nodes required. Simulation results also show that varying the mobile node's speed does not impact the performance of iCCA-MAP, while MCL and Dual MCL's performance is impacted.

Keywords: Wireless Sensor Networks, Node Localization, Mobile Nodes, Sensor Nodes, Monte Carlo Localization (MCL), Dual MCL, Curvilinear Component Analysis (CCA).

1 Introduction

Recent advancements in electronics and wireless communication technology has led the way to the development of tiny, low-power, low-cost sensor nodes which have the ability to sense physical phenomena, process data, and communicate with one another. A large number of these wireless sensor nodes are deployed across a geographical region to form a wireless sensor network (WSN). These WSNs create smart environments by providing access to information regarding the environment through collecting, processing, analyzing, and disseminating data whenever required. In order to use WSNs in inaccessible terrains or disaster relief operations, random deployment of the sensor nodes is required. As a result, the position of these nodes will not be predetermined and thus the nodes must have the ability to collaborate with each other to form self-organized networks in order to perform tasks including, but not limited to, determining their location [1].

The location of the nodes in a WSN is of great importance, given that without location information we would not know where in the network the collected data is

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 163–176, 2010.
© Springer-Verlag Berlin Heidelberg 2010

coming from. As a result, the data would become less meaningful and we would therefore not be making effective use of our WSN. Knowing a node's location is also required for many network protocols and middleware services that rely on location information, such as geographic routing protocols [12], context-based routing protocols [6], location-aware services [9], and enhanced security protection mechanisms [25].

In the literature, substantial research on location estimation of stationary nodes in WSNs has been presented [3, 4, 8, 13, 17, 18]. These algorithms typically localize normal sensor nodes with the help of anchor nodes. An anchor node is a special node which knows its location either by statically configuring the location information or by equipping these nodes with GPS receivers or other dedicated localization hardware. For localization in the 2-D space, a minimum of three anchor nodes is required, and many algorithms can increase their localization accuracy with the number of anchor nodes in the network. However, not as much research pertaining to mobility in WSNs has been conducted. Recently however, the subject of mobility in WSNs has gained much interest due to the increasing number of applications that require mobile sensor nodes. Animal tracking, logistics applications, and elderly healthcare home monitoring are but a few of such applications.

Studies conducted on introducing mobility in WSNs have demonstrated an overall improvement not only by increasing the overall network lifetime, but also by improving the data capacity of the network as well as addressing delay and latency problems [14, 24]. As a result, many researchers have started to investigate the concept of mobility in WSNs. A number of algorithms that can estimate the location of mobile nodes within WSNs have been developed [2, 5, 11, 15, 19, 23, 24].

In this paper, the accuracy of location estimates obtained from iCCA-MAP (iterative Curvilinear Component Analysis MAP) [2] while varying the number of anchor nodes used and the mobile node’s speed is compared to that of MCL (Monte Carlo Localization) [11] and Dual MCL [23], two competitive localization algorithms. The remainder of this paper is organized as follows. Section 2 gives an overview of previous work done in the field of mobile wireless sensor networks. The iCCA-MAP algorithm is detailed in Section 3. Section 4 presents simulation results to compare iCCA-MAP, MCL, and Dual MCL. Finally, Section 5 concludes the paper.

2 Related Work

In this paper, we are interested in range-free localization techniques (i.e., algorithms that deduce node locations from connectivity information only) since range-based localization algorithms (which use ranging techniques such as TDoA or RSSI to measure the distance between nodes) often require additional node hardware and suffer from noise in the ranging measurements. Stationary (range-free) sensor node localization has received tremendous attention from the wireless sensor network community [4, 8, 13, 18]. However, mobile node localization as well as the use of mobile anchor nodes for assistance in localization is gaining more interest as the number of applications requiring mobility increases. In this section, algorithms involving sensor node mobility in WSNs are reviewed. Algorithms involving mobility can be divided into three categories:

1. Algorithms requiring mobile anchor nodes which aid in localizing stationary wireless sensor nodes [10, 16, 21, 22];
2. Algorithms proposed for localization of mobile sensor nodes [19, 26];
3. Algorithms proposed for localization of mobile sensor nodes for which mobile anchor nodes are used [5, 11, 15, 23].

Since we are interested in mobile nodes, this section will only review algorithms proposed for categories 2 and 3, with an emphasis on the competitive comparison algorithms selected in this paper, MCL [11] and Dual MCL [23]. Then, an overview of CCA-MAP [13], upon which our algorithm is built, will be provided as well.

2.1 Localization Algorithms for Mobile Sensor Nodes

One of the early works on locating mobile nodes with the aid of mobile anchor nodes in the context of range-free localization is by Hu and Evans [11], who introduce the sequential MCL method. Their algorithm is based on adopting the Monte Carlo Localization developed for localization in robotics. In the MCL algorithm, the posterior distribution of possible locations, as the valid solutions to the underlying problem, is represented by a set of weighted samples. Because nodes have no knowledge of their location, initially a set of *N* random locations in the deployment area are chosen. The algorithm consists of two phases, namely the prediction and the filtering phase. In the prediction phase, the node calculates its possible locations based on the previous possible locations and its maximum velocity. Anchor nodes then transmit their location information and based on the nodes' observation of the anchor node locations, samples inconsistent with observations are filtered out. There are two types of observations: direct anchor observations and indirect anchor observations. In a direct anchor observation, if the node hears the anchor, it must lie within a circle of radius *r* (the transmission range) of the anchor's location. An indirect anchor observation is when a node does not hear an anchor but one of its neighbors does, thus indicating that the node must lie within distance *r* and 2*r* of the anchor's location. This process is repeated until a satisfactory estimate of the nodes' locations is made.

In [23], Stevens-Navarro *et al.* propose two variations of the MCL algorithm: Dual MCL and Mixture MCL. Dual MCL is the logical inverse of the original MCL algorithm, and Mixture MCL is a combination of the original MCL and Dual MCL. In Dual MCL, the sampling process is inverted, making its prediction step more elaborated than that of original MCL. In the prediction step of Dual MCL, samples are generated from the deployment area and are validated based on anchor nodes being heard by nodes or by their neighbouring nodes. In the Dual MCL filtering step, at every time interval, the validated predicted samples are filtered based on the previous location of the node and the maximum velocity the node can travel. The Mixture MCL algorithm combines the Dual MCL and the original MCL algorithm by generating samples using both methods and mixing these samples together using a mixing rate. The results generated using Mixture MCL is not as accurate as that of Dual MCL, but in terms of the trade off between computational time and estimated location accuracy, Mixture MCL outperforms Dual MCL. However, as estimated location accuracy is of importance to this paper, we will focus on Dual MCL for comparison purposes. The authors report that for both Dual MCL and MCL, a high percentage of nodes are required to be anchor nodes.

EMAP [15] is an extended variation of the MAP algorithm [22] used for mobile nodes. The MAP algorithm uses mobile anchors in a static sensor network, whereas EMAP extends MAP to deal with mobile sensor nodes in WSNs. In EMAP, it is assumed that the mobile nodes know their Euclidean distance and moving direction. Once the location of a sensor node is modified, its beacon points must also be repositioned as the sensor nodes change their moving direction or obtain the last beacon point. An inherent attribute of WSNs is their random deployment and, in all likelihood, random movement in mobile WSNs, therefore the assumption that the mobile nodes know their distance and moving direction is a major drawback of this algorithm.

Enhanced Color-theory-based Dynamic Localization (E-CDL) proposed in [5] is based on the CDL (Color-theory-based Dynamic Localization) algorithm proposed in [19] whose location accuracy depends on the accuracy of the average hop distance derivation. Here the authors employ mobile anchor nodes rather than stationary ones in order to enhance the accuracy of the measurements and to decrease the possibility of sensor node isolation in the multihop environment. The anchor nodes are placed in the four corners of the square field and move a distance of a radio range r in every time slot. Another improvement to the CDL scheme is the introduction of two new methods for calculating the average hop distance measurements; the first method calculates the expected value of the next hop distance based on the next hop being located between $0.5r$ and r, and the second proposed method adjusts the average hop distance based on the ratio of the Euclidean distance to the shortest path length.

MCL and Dual MCL have been chosen as the comparison algorithms for iCCA-MAP. The reasons MCL and Dual MCL have been chosen are that these algorithms do not require additional hardware, similar to iCCA-MAP, provide rather accurate localization estimates, and can be used in scenarios where both mobile and stationary nodes are deployed. Comparisons made are for scenarios where a sensor node is mobile and the anchor nodes are static, since this is the scenario iCCA-MAP has been designed for. While Dual MCL has been reported to have superior localization performance [23], we opted to include both algorithms in our study. One reason is that the superior performance of Dual MCL, as reported in [23], comes at the cost of a significantly higher computational complexity. In addition, MCL has been used as a base algorithm in many recent papers such as [20].

2.2 The CCA-MAP Algorithm

The original CCA-MAP algorithm is a localization solution which requires only a minimum number of anchor nodes in order to facilitate a rapid deployment process and can achieve a high level of position accuracy with or without the assistance of range measurements. CCA-MAP uses a self-organizing neural network originally proposed for non-linear data mapping, called Curvilinear Component Analysis (CCA) [7] to build local maps for every node in the network. Each node uses only local information to compute its own local map. The local maps are then patched together to form a global map. This global map has node coordinates that indicate relative node placements, but these coordinates are not tied into any external coordinate system. Using anchor nodes, in a final step, this relative global map can then be translated via the Procrustes method [13] into an absolute global map, where node coordinates reflect the node positions based on the coordinates used to localize the anchor nodes.

In the CCA-MAP algorithm, neighbours within *h* hops are included in building the local map for each node. The minimum hop distance between nodes of the local map is computed and used as the approximate distance matrix. Each node then applies the CCA algorithm generating the relative coordinates for every node in its local map. The local maps are then merged. The merged map transforms to an absolute map based on positions of the anchor nodes. For the starting map, the local map of a randomly selected node is used. Then, the neighbour node whose local map shares the most nodes with the current map is selected to merge its local map into the current map. Using the coordinates of their common nodes, two maps are merged. A linear transformation is applied for merging a new local map into the current map. In CCA-MAP, local maps can be merged in parallel in different parts of the network. During map merging anchor nodes are not required. However, when at least three anchor nodes are found in the patched map, coordinates of the anchor nodes can be used to compute the absolute coordinates of the nodes in that map.

3 The iCCA-MAP Algorithm

The iCCA-MAP (iterative CCA-MAP) algorithm has been initially proposed in [2]. In fact, iCCA-MAP is a simplified version of the original CCA-MAP algorithm since the local map is only computed for the mobile node rather than every node in the network. Then, the newly calculated local map is patched into the relative global map. A simplifying assumption we made is that the mobile node is known. In an implementation of this algorithm, we would at best be able to determine relative mobility. Therefore, we may have to run our modified algorithm on multiple nodes.

In the first step of the algorithm, the CCA-MAP algorithm is used for obtaining the initial location of every node in the network including the mobile node. The iCCA-MAP algorithm differs from CCA-MAP in that it iteratively calculates the local map of the mobile node only and patches this new local map into the existing global map to obtain its location at every time step it is invoked. By only calculating the local map of the mobile node, the complexity of local map computation in iCCA-MAP decreases from $O(k^2n)$ to $O(k^2)$, where *k* is the average number of 2-hop neighbours and *n* is the total number of nodes in the network. As shown in [2], this reduction results in much lower computational time for iCCA-MAP as opposed to CCA-MAP, for networks with larger number of nodes (assuming constant density).

To merge the local map of the mobile node to the global map, the relative global map obtained from running CCA-MAP is saved, the mobile node is removed from the original global map, and then the mobile node's local map is merged with the global map using the common nodes. Merging the global map, obtained from the initial run of the CCA-MAP algorithm, and the local map of the mobile node, decreases this step's complexity from $O(k^3n)$ to $O(k^3)$, since now only one map is being merged rather than the local map of every node in the network.

In the last step, using at least three anchor nodes for 2-dimensional space and four for 3-dimensional space, the merged map (global map and mobile node local map) is transformed to an absolute map based on the absolute positions of the anchor nodes. For *a* anchor nodes, the complexity of this step is $O(a^3+n)$, the same as CCA-MAP. All steps except step one of the algorithm are executed repeatedly in order to provide the coordinates of the mobile node as it moves.

In a centralized deployment, where we assume a sequential execution of the steps of the algorithm, the total complexity of iCCA-MAP is as follows:

$$O(k^2)+ O(k^3)+ O(a^3+n) \approx O(n) . \quad (3.1)$$

The overall complexity of iCCA-MAP is $O(n)$, assuming that, as the network size grows, the density (and therefore *k*) stays constant. Also, we are usually interested in only a fixed number of anchor nodes, which results in *a* being constant as well. The total complexity of CCA-MAP is as follows:

$$O(k^2n)+ O(k^3n)+ O(a^3+n) \approx O(n) . \quad (3.2)$$

The total overall complexity of iCCA-MAP is asymptotically the same as that of CCA-MAP (again assuming that *k* and *a* are constant). However, as indicated by the k^3 factor, the constant may be significantly higher. As mentioned in [2], iCCA-MAP is indeed significantly faster than CCA-MAP, allowing the algorithm to track the location of a mobile node in near real-time, resulting in better localization performance at a fraction of the computational costs.

4 Simulation Results

In WSNs that contain mobile nodes, it is essential to use an algorithm that is accurate and efficient, thus estimation accuracy and computational time are the most important performance metrics. In this paper, we evaluate the localization accuracy of iCCA-MAP by comparing it to the well-known mobile node localization algorithm MCL and an improved variation of it, namely Dual MCL.

This section begins with a description of the experimental setup. Then, the average localization error of all three algorithms is compared with respect to the network size for different percentage of anchor nodes and different node speeds.

4.1 Experimental Setup

For the simulation of iCCA-MAP and CCA-MAP, Matlab R2009b, Version 7.9.0.529 for Windows was used. Windows 7 Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz with 12.0 GB memory was used as the computing platform for iCCA-MAP. While for the simulation of MCL and Dual MCL, Fedora Eclipse Platform, Version 3.3.2 with Java Version 1.6.0 was used on a Linux workstation with a 2.66 GHz CPU and 3.25GB memory.

The MCL simulation code (written in Java) was provided by its authors, Hu and Evans [11]. However, in order to allow both iCCA-MAP and MCL implementations to read in the same network topologies and mobility scenarios as well as to produce the same error statistics, changes to the original MCL code were made. Using the modified MCL code, Dual MCL was implemented according to the description provided by Navarro *et al.* as outlined in [23]. As the implementation languages for the various localization algorithms differ, we cannot fairly compare computation times across the various algorithms. Similarly, as each implementation models the networking aspects differently, comparing message costs is not trivial either. Rather, in this paper, we focus on the algorithms' localization performance as a function of network size, number of anchor nodes, and mobile node movement speed.

All simulations were done on a random square network configuration. The radio range (r) was set to 8 for all nodes and anchors. In all cases, the average node density was kept constant by increasing the network area as the number of nodes increased. The average node density for the networks was approximately 12. To generate a path for the mobile node, we used the Random Waypoint mobility model.

When comparing iCCA-MAP to MCL and Dual MCL, network sizes of 50, 100, 150, 200, 250, and 300 nodes were used, and for each network size, 10 random networks were generated. Four sets of simulations were performed where the number of anchor nodes was varied and the localization error measured. In the first set, the number of anchor nodes is the minimum required number, namely three and in the second set, 20% of the total nodes have been designated as anchor nodes. In the first two simulation sets, the speed of the mobile node was kept constant and in the second two simulation sets the speed was varied. A total simulation time of 100 seconds was used for all simulation sets. The mobility parameters were set as follows:

- Minimum speed: 0 m/s;
- Maximum speed: 2 m/s, 8 m/s;
- Pause time: 0 seconds.

These two speeds were chosen to simulate walking and running speeds. It was also of interest to see the effect speed has on the performance of the algorithms. When normalized by radius r, which is set to 8 for all simulations, the corresponding speeds become 0.25 *r/unit time* and 1 *r/unit time* respectively. Results presented in Sections 4.2 and 4.3 are pertaining to speed of 1 *r/unit time*. Section 4.4 compares the effects of the two speeds on iCCA-MAP, MCL, and Dual MCL.

4.2 Average Localization Errors of iCCA-MAP, MCL, and Dual MCL

In this section, we evaluate the performance of iCCA-MAP with respect to localization accuracy. In the first set of simulations, three anchor nodes were used and the location errors of iCCA-MAP, MCL, and Dual MCL were calculated using the Euclidean distance between the real location of the mobile node and the estimated location, normalized by r, as follows:

$$Localization\ Error = \frac{\sqrt{(X_{real} - X_{estimated})^2 + (Y_{real} - Y_{estimated})^2}}{r} \qquad (4.1)$$

Throughout the simulation, we sampled the actual node location and the node location as determined by the localization algorithm under study every second. The reported results are the average localization error over the 100 samples for each run and the 10 runs for each network size.

Simulation results for average localization error of the three algorithms are depicted in Fig. 1. As can be seen, the average localization error of iCCA-MAP is lower than that of MCL and Dual MCL. The error bars depicted in the figure illustrate the 95% confidence interval. As the confidence intervals of iCCA-MAP, MCL and Dual MCL do not overlap, it can be concluded that the differences in their localization performance are statistically significant. The only exception is the 50 node network where all 3 algorithms provide similar results. In addition, for iCCA-MAP, the confidence intervals for different network sizes overlap, signifying that the average localization error does not change with network size. The performances of MCL and Dual MCL, on the other hand, deteriorate as the network size increases. Dual MCL outperforms MCL as expected and corresponds to results reported in [23].

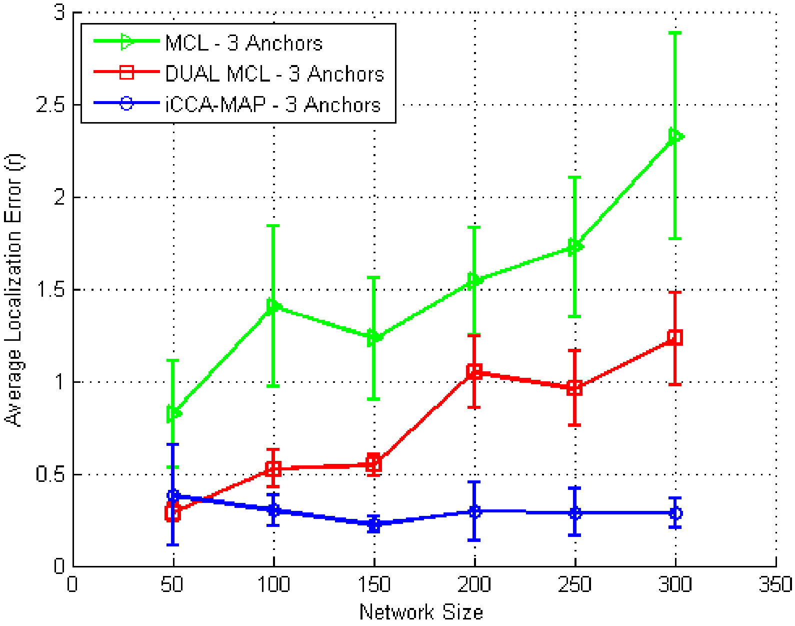


Fig. 1. Average localization error of iCCA-MAP versus MCL and Dual MCL for network sizes of 50, 100, 150, 200, 250, and 300 nodes with 3 anchor nodes and speed of 8m/s

The biggest advantage of iCCA-MAP over MCL and Dual MCL is its ability to provide accurate estimates of node location with the minimum number of anchors required, namely three for the 2-dimensional space. Increasing the number of anchors results in a higher cost and energy consumption for nodes since anchor nodes usually require a GPS receiver to be mounted on them. While CCA-MAP and iCCA-MAP have been shown to not benefit significantly from additional anchors, MCL and Dual MCL both are very sensitive to the number of anchors used. In the next section, we therefore explored the impact of adding more anchors to the network.

4.3 Varying the Number of Anchor Nodes

In this section, we evaluate the performance of iCCA-MAP, MCL, and Dual MCL with respect to varying the number of anchor nodes. Again, randomly deployed networks of 50, 100, 150, 200, 250, and 300 nodes were used for evaluating the effectiveness of increasing the percentage of anchors in the networks at a maximum mobile node speed of 8 m/s.

Typically, as the number of anchor nodes increases in a WSN, the accuracy of the localization estimates also increases, as there are more nodes that have exact information about their location, and thus can provide precise location information to the regular nodes. In iCCA-MAP, the transformation (rotation, scaling, and translation) of the global map in order to obtain its absolute position, will be computed using more nodes. For MCL and Dual MCL, the increase in anchor nodes translates into a higher number of observations for non-anchor (or regular) nodes, thus improving their location estimation.

Fig. 2 depicts simulation results for iCCA-MAP, MCL, and Dual MCL when three anchor nodes and 20% anchor nodes are used in the network. We can see that an increase in the percentage of anchor nodes improves the performance of all three localization algorithms. iCCA-MAP's performance is less sensitive to the number of anchor nodes, and the localization accuracy it provides with the minimum number of anchor nodes, namely three, still outperforms MCL's performance even with a higher percentage of its nodes as anchor nodes. However, Dual MCL provides slightly better results when the number of anchors nodes in the network has been increased to 20%.

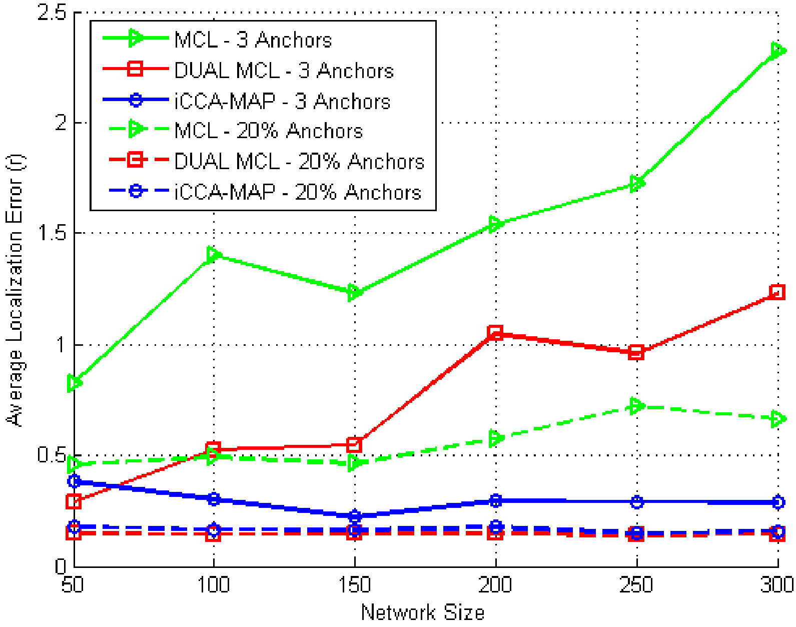


Fig. 2. Average localization error for iCCA-MAP versus MCL and Dual MCL for three anchor nodes and 20% anchor nodes and node speed of 8m/s

Although both MCL and Dual MCL show significant improvements in localization estimates with the increase in the number of anchor nodes, they do not outperform iCCA-MAP when only three anchor nodes are present in the network. As reported in [23], due to the increase in time required to obtain results for Dual MCL, the Mixture MCL algorithm has been proposed, which has a lower computational time as well as lower location accuracy. Therefore, it could be concluded that for the Dual MCL algorithm, better accuracy is obtained by sacrificing computational time which is unfavourable for mobile nodes in WSN. In [2], it was shown that obtaining accurate location estimates in near real-time is of high importance in real world conditions. Using the obtained results in this section and the results provided by [23] regarding the computational time of Dual MCL, it can be concluded that iCCA-MAP is a more accurate, efficient, and cost-effective algorithm than MCL and Dual MCL.

4.4 Varying the Speed of the Mobile Sensor Node

In this section, simulation results for mobile node speed of 2 m/s equivalent to 0.25 *r/unit time* are presented. The results are compared to results pertaining to node speed of 8 m/s equivalent to 1 *r/unit time* and the effect of varying the speed of the mobile node is analyzed and discussed.

Fig. 3 depicts the average localization error of iCCA-MAP, MCL and Dual MCL for network sizes of 50, 100, 150, 200, 250, and 300 with three anchor nodes for both speeds of 2 m/s and 8 m/s. As seen in Fig. 3, for iCCA-MAP and MCL there is a

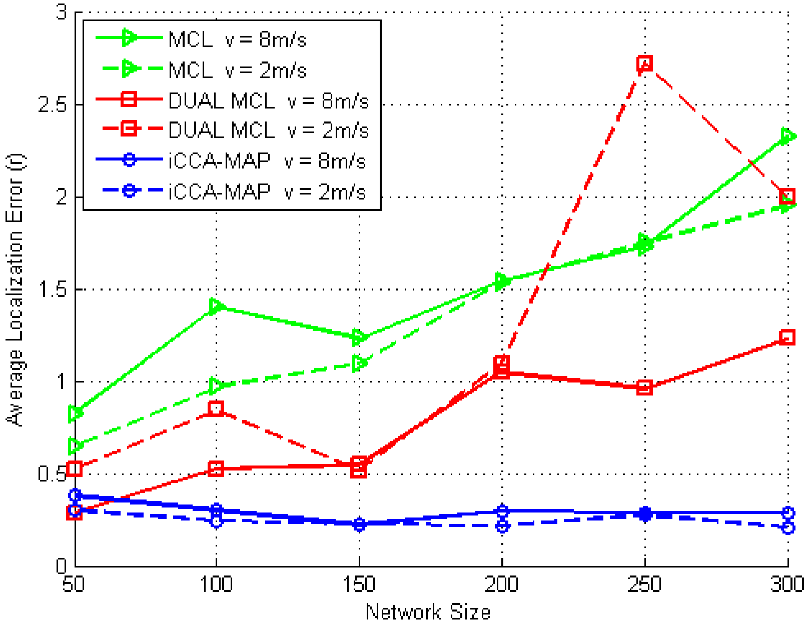


Fig. 3. Average localization error of iCCA-MAP versus MCL and Dual MCL for network sizes of 50 - 300 nodes with 3 anchor nodes and speeds of 2m/s and 8m/s

slight increase in localization error as the speed increases from 2 m/s to 8 m/s for most network sizes. Error bars for 95% confidence interval have been omitted in the figure for the purpose of clarity. However, in most cases, the error bars either completely overlap or there is quite a bit of overlapping, indicating that the improved performance may not be statistically significant. In the case of Dual MCL, as seen in Fig. 3, for most network sizes the average localization error decreases with the increase of speed. This occurs because at higher speeds, the Dual MCL filtering step, which filters samples based on the mobile node's previous location and its maximum velocity per unit time, allows for the possibility of collecting more samples that meet the condition and thus can provide a better estimate of the node's location. At low speeds, since the area in which the node can reside based on the node's previous location and its maximum velocity is very small, the chances of finding enough sample points is rather low. In that case, the node's previous location is used as an estimate of its current position and if the node's previous location is not available (i.e., node has not been initialized yet), a location near the center of the simulation area is assigned to the node. The fact that only three anchor nodes are present in the network also contributes to the problem of finding enough sample points, since the number of observations will be low when the number of anchor nodes is low.

Fig. 4 depicts the average localization error of iCCA-MAP, MCL and Dual MCL for speeds of 2 m/s and 8m/s when 20% of the nodes in the network are anchor nodes. As depicted in Fig. 4, for most network sizes, MCL's performance deteriorates as the speed of the mobile node increases. This can be explained by the fact that samples

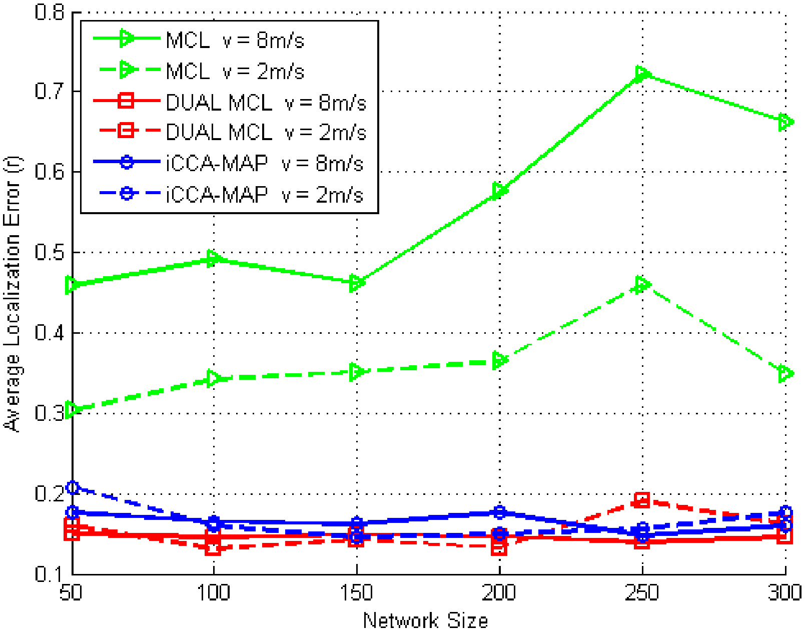


Fig. 4. Average localization error of iCCA-MAP versus MCL and Dual MCL for network sizes of 50 - 300 nodes with 20% anchor nodes and node speeds of 2m/s and 8m/s

predicted in MCL are based on the previous predicted location of the node and the node's maximum speed per unit time. As the mobile node's speed increases, the area in which the predictions can be obtained also increases, resulting in less accurate estimations. In the case of iCCA-MAP and Dual MCL, no definite conclusions can be made with respect to varying speeds as there exists no obvious trend and there is quite a bit of overlap of the error bars (which have been omitted for clarity of the figure), illustrating low statistical significance.

Based on the presented results, we conclude that MCL and Dual MCL are impacted by mobility depending on anchor node density. For low anchor node density, Dual MCL is adversely affected by low speed, whereas for a high anchor node density MCL suffers from higher mobility. iCCA-MAP, on the other hand, performs consistently well independent of speed and the number of anchor nodes.

5 Conclusions

The localization error of iCCA-MAP has been compared to that of MCL and Dual MCL. Simulation results show that iCCA-MAP outperforms the MCL and Dual MCL algorithms in finding the location of a mobile node in a WSN with respect to localization error using the minimum number of anchor nodes. When the number of anchor nodes is increased to 20% of the total nodes in the network, both MCL and Dual MCL show significant improvement, as they require a high anchor node density in order to perform well. iCCA-MAP, however, demonstrates very slight improvement with the increase in the number of anchor nodes. Results for iCCA-MAP and Dual MCL are very similar when there are 20% anchor nodes present in the network. The other parameter varied in these sets of simulations was speed, which seemed to have negligible effect on iCCA-MAP. MCL's performance deteriorates with increase of speed when there are 20% anchor nodes in the network. Higher speeds improve the performance of Dual MCL when anchor node density is very low (i.e., three anchors in the network).

The results in this paper are based on a fixed average nodal density of 12. MCL has been reported as being relatively insensitive to node density [11], the localization performance of Dual MCL improves slightly with nodal density [23]. Similarly, the core CCA-MAP algorithm benefits from increased nodal density [13]. We expect that the relative performance differences would hold at different nodal densities, but further studies are required to confirm this.

Similarly, we have not directly compared computation or communication costs of the three localization algorithms. However, based on our observations in running the simulations, and consistent with the results reported in [23], Dual MCL, while providing better localization performance than MCL, also requires much longer execution times. All localization algorithms require that nodes determine neighbourhood information through the periodic exchange of HELLO messages. However, a more in-depth comparison and analysis is yet to be done, and would require to implement all localization algorithms on the same platform.

The core algorithms could also be improved further. For example, what is the optimal frequency for running iCCA-MAP? The best choice involves a trade-off among localization accuracy, power consumption, the amount of resources available to the node for attending to application-related tasks, and the number of messages sent across the air interface.

Using simulations, we are also interested in determining the maximum percentage of mobile nodes that can be localized using iCCA-MAP, as iCCA-MAP is not suited for localizing an all-mobile WSN. This is because in iCCA-MAP, the mobile node is first removed from the stationary global map and then its local map is patched into that same map. If all the nodes become mobile, the global map would seize to exist as all nodes would have to be removed from it.

Obtaining iCCA-MAP localization results for different network topologies such as grid square networks, C-shaped networks, loop networks, and pipeline networks is another task for future work. The obtained results can be compared to results from other algorithms in order to draw conclusions as to which network topologies iCCA-MAP is better suited for.

Another important future work is to implement iCCA-MAP on a small-scale testbed in order to compare simulation results with real world results. We would also like to compare iCCA-MAP to other variants of MCL and Dual MCL such as [20].

References

[1] Akyildiz, I.F., Sankarasubramaniam, Y., Su, W., Cayirci, E.: A Survey on Sensor Networks. IEEE Communications Magazine 40(8), 104–112 (2002)

[2] Alikhani, S., St-Hilaire, M., Kunz, T.: iCCA-MAP: A New Mobile Node Localization Algorithm. In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 382–387 (October 2009)

[3] Bahl, P., Padmanabhan, V.: RADAR: An in-building RF-based User Location and Tracking System. In: Proceedings of the IEEE International Conference on Computer Communications, pp. 775–784 (March 2000)

[4] Boukerche, A., Oliveira, E.H.A.B.F., Nakamura, F., Loureiro, A.A.F.: A Novel Lightweight Algorithm for Time-Space Localization in Wireless Sensor Networks. In: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, pp. 336–343 (October 2007)

[5] Chang, T.-C., Wang, K., Hsieh, Y.-L.: Enhanced Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks. In: Proceedings of the IEEE Wireless Communications and Networking Conference, pp. 3064–3069 (March 2007)

[6] Cugola, G., Migliavacca, M.: A Context and Content-Based Routing Protocol for Mobile Sensor Networks. In: Proceedings of the 6th European Conference on Wireless Sensor Networks, pp. 69–85 (February 2009)

[7] Demartines, P., Hérault, J.: Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets. IEEE Transactions on Neural Networks 8(1), 148–154 (1997)

[8] Doherty, L., Pister, K., El Ghaoui, L.: Convex Position Estimation in Wireless Sensor Networks. In: Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1655–1663 (April 2001)

[9] Friedman, R., Kliot, G.: Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey. Israel Institute of Technology, Technical Report CS-2006-10 (2006)

[10] Galstyan, A., Krishnamachari, B., Lerman, K., Pattem, S.: Distributed Online Localization in Sensor Networks using a Moving Target. In: Proceedings of the 3rd International Symposium on Information Processing In Sensor Networks, pp. 61–70 (April 2004)

[11] Hu, L., Evans, D.: Localization for Mobile Sensor Networks. In: Proceeding of the International Conference on Mobile Computing and Networking, pp. 45–57 (October 2004)

[12] Kalosha, H., Nayak, A., Rührup, S., Stojmenovic, I.: Select-and-Protest-Based Beaconless Georouting with Guaranteed Delivery in Wireless Sensor Networks. In: Proceedings of the 27th Annual IEEE Conference on Computer Communications (INFOCOM), pp. 346–350 (April 2008)

[13] Li, L., Kunz, T.: Cooperative Node Localization Using Non-linear Data Projection. ACM Transactions on Sensor Networks 5(1), 1–26 (2009)

[14] Liu, B., Brass, P., Dousse, O., Nain, P., Towsley, D.: Mobility Improves Coverage of Sensor Networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc), pp. 300–308 (May 2005)

[15] Ou, C.-H.: Range-Free Node Localization for Mobile Wireless Sensor Networks. In: Proceedings of the 3rd International Symposium on Wireless Pervasive Computing, pp. 535–539 (2008)

[16] Priyantha, N.B., Balakrishnan, H., Demaine, E.D., Teller, S.: Mobile-Assisted Localization in Wireless Sensor Networks. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), pp. 172–183 (March 2005)

[17] Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: Proceedings of the International Conference on Mobile Computing and Networking (ACM MOBICOM), pp. 32–43 (August 2000)

[18] Shang, Y., Ruml, W.: Improved MDS-Based Localization. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), pp. 2640–2651 (March 2004)

[19] Shee, S.-H., Wang, K., Hsieh, Y.-L.: Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks. In: Proceedings of the Workshop on Wireless, Ad Hoc, Sensor Networks, pp. 73–78 (August 2005)

[20] Sheu, J.-P., Hu, W.K., Lin, J.-C.: Distributed Localization Scheme for Mobile Sensor Networks. IEEE Transactions on Mobile Computing 9(4), 516–526 (2010)

[21] Sichitiu, M., Ramadurai, V.: Localization of Wireless Sensor Networks with a Mobile Beacon. In: Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), pp. 174–183 (October 2004)

[22] 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)

[23] Stevens-Navarro, E., Vivekanandan, V., Wong, V.W.S.: Dual and Mixture Monte Carlo Localization Algorithms for Mobile Wireless Sensor Networks. In: Proceedings of IEEE Wireless Communications and Networking Conference, pp. 4024–4028 (March 2007)

[24] Wang, W., Srinivasan, V., Chua, K.-C.: Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. In: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, pp. 270–283 (August 2005)

[25] Xiao, Y., Sethi, S., Chen, H.-H., Sun, B.: Security Services and Enhancements in the IEEE 802.15.4 Wireless Sensor Networks. In: Proceedings of the IEEE Global Telecommunications Conference, pp. 1–5 (November 2005)

[26] Yu, G., Yu, F.: A Localization Algorithm for Mobile Wireless Sensor Networks. In: Proceedings of the IEEE International Conference on Integration Technology, pp. 623–627 (March 2007)

ADNL-Angle: Accurate Distributed Node Localization for Wireless Sensor Networks with Angle of Arrival Information

Julien Champ and Vincent Boudet

LIRMM - CNRS - University of Montpellier 2
{champ,boudet}@lirmm.fr

Abstract. This paper deals with the collaborative multi-hop localization problem in static Wireless Sensor Networks with Angle of Arrival information. We present ADNL-Angle, a new, original and efficient localization algorithm. Our method uses a small set of sensors which know their positions, it also uses the 2-hop neighborhood knowledge and angular information between neighbor nodes computed thanks to AoA technology. Each node with enough anchors in its neighborhood locally runs a kind of force-based algorithm so as to localize itself, and then forward its position to enable other nodes to locate themselves. We also propose an additional delaying mechanism to increase localization accuracy. We provide extensive simulation results showing the accuracy and the robustness of ADNL-Angle even with noisy angular measurements, few anchor nodes and realistic topologies.

1 Introduction

As a consequence of recent advancements in miniaturization and wireless communications, a new kind of network has come to the fore: Wireless Sensor Networks (WSN). In those networks, nodes (sensors) can gather information from their environment, such as temperature, gas leak, etc. They can also communicate, thanks to their wireless communication device, with other nodes in their transmission range. WSN recently attracted a lot of attention because of their wide range of applications [1].

As in WSN many applications are monitoring tasks, it is often needed to annotate sensed data with geographical information. Geographical positions of nodes can also be used to communicate in such networks, for example in geographic routing [3]. Thus, each sensor has to know or assess its own position as accurately as possible: this is the localization problem. The easiest way to solve this problem is to equip all nodes with a GPS (Global Positioning System) device. But a GPS comes with many drawbacks: it often does not work in indoor environment, moreover it is expensive to equip all sensors with such a device, and as most of the time sensors are static, it is clearly not cost-efficient to provide such equipment for only a one-time localization.

Sensor nodes are powered by limited batteries and are only able to communicate with their neighbors, so they need to collaborate so as to estimate their positions. Many algorithms have been proposed in the literature according to various hypothesis, such as the presence of anchor nodes, centralized or distributed algorithms, distance or angular information, etc.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 177–190, 2010.
© Springer-Verlag Berlin Heidelberg 2010

In this paper we focus on the anchor and angle-based distributed multi-hop localization problem in static WSN. It considers a small set of nodes which know their positions (anchor-based), and nodes with the ability to obtain angular information from their neighbors (angle-based). We propose ADNL-Angle algorithm, whose main principle is to use an original angular force-based algorithm in sensors' neighborhood so as to locate sensors. Thanks to the localization process using the force-based algorithm, and thanks to the delaying mechanism before executing the localization process, our proposed localization scheme produces very accurate localization results.

This paper is organized as follows: Section 2 describes the localization problem and previous works. Section 3 is dedicated to the description of the different steps of ADNL-Angle method. In section 4 we describe parameters used to run simulations, and we present and discuss simulation results. Section 5 concludes this paper and describes our plans for future work.

2 Related Work

2.1 The Localization Problem

Let's consider n sensor nodes deployed in a given physical region, for example a square area. Each node is able to communicate with other nodes inside its communication range: its neighbors. Using their wireless communication device, sensor nodes can collaborate to perform chosen tasks. The localization problem consists in finding geographical coordinates for all nodes, as accurately as possible. However, various hypothesis can be made to solve this problem.

In the single-hop localization problem, each node is adjacent to at least three anchor nodes, contrary to the multi-hop localization problem where nodes which need to be localized can be several hop distant from anchor nodes.

Among localization algorithms in the literature, we can find two main categories: measurement-based and measurement-free approaches. In measurement-based algorithms, a specific hardware is needed to provide for example distance or angular measurements between neighbor nodes. Technologies such as RSSI (Received Signal

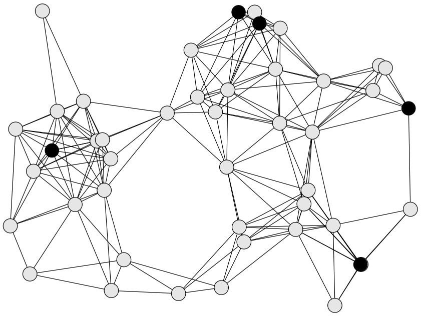


Fig. 1. Random deployment of sensor nodes with five anchors (nodes aware of their positions)

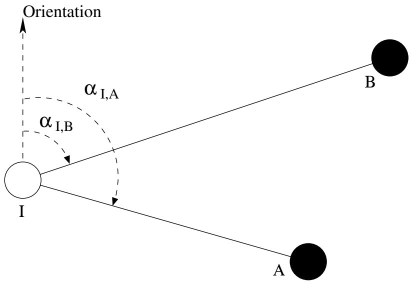


Fig. 2. Principle of AoA localization with orientation information

Strength Indicator), ToA (Time Of Arrival) or TDoA (Time-Difference of Arrival) provide an estimation of the distance between neighbor nodes, whereas AoA (Angle of Arrival) provides angular information between neighbor nodes [7]. Measurement-free approaches only consider the connectivity knowledge of the network (or of the neighborhood) to solve the localization problem. Obviously measurement-based approaches perform better regarding accuracy, but may require extra hardware.

Centralized and distributed algorithms can be found in the literature, and they obviously both have advantages and drawbacks. But, regarding scalability and energy consumption it is generally more suitable to use distributed algorithms, excepted maybe when considering small networks.

Generally in the localization problem, a small set of nodes, named anchor nodes, know their positions thanks to a GPS receiver or to manual deployment. As GPS is expensive and as it cannot be used in indoor environments it is not possible to equip all nodes with such a device. Some localization algorithms, are called anchor-free: they provide localization information, but with relative positions comparing to absolute positions provided by algorithms using anchor nodes.

Figure 1 shows a random deployment of wireless sensor nodes, with five nodes aware of their positions (anchor nodes) represented by black circles.

In this paper, we are interested in the distributed multi-hop localization problem using angular information. Thanks to an antenna array, the direction of the signal received by a sensor node can be determined, and is known as angle of arrival. Figure 2 represents two anchor nodes denoted A and B and one node with unknown location denoted I. With orientation information, using a compass for example, and thanks to AoA technology, node I is able to measure angles $\alpha_{I,A}$ and $\alpha_{I,B}$. If angle measurements are accurate, it is easy to determine node I position using anchor nodes A and B positions. When measurements are not accurate, and when sensor nodes do not have two or more anchor nodes in their neighborhood the problem becomes more difficult.

2.2 Existing Algorithms

Various methods have been proposed to solve the multi-hop localization problem, but not so many using AoA information. We describe here three interesting methods. For a more exhaustive description of existing localization algorithms, references can be found in [7,5].

APS-AoA. One of the most known localization method using angular information, named APS-AoA, is described in [6]. Its main principle is to extend the triangulation method to multi-hop networks. In APS-AoA, angular measurements are exchanged between neighbors and using geometric relations among nodes, non-located nodes which are multiple hops away from anchor nodes determine estimations of angular information regarding anchor nodes. Then, using triangulation, non-located sensors compute an estimation of their positions.

The main strength of this method is it simplicity, however results regarding positions are not particularly accurate, especially when considering potentially large measurement errors. To perform well, this method needs a higher number of anchor nodes and/or a higher connectivity than other methods.

Probabilistic. The probabilistic localization scheme, presented in [8] is, as far as we know, the most accurate distributed localization method using AoA measurements. Its main principle is to use the measurement errors model when determining sensors positions thanks to a probability density functions (pdf). Firstly, non-located sensors compute their pdf using anchor nodes messages, and then communicate this pdf. Pdf are updated using received messages and after several exchanges non-located sensors positions are determined through the pdf.

Despite its interesting results regarding accuracy, the probabilistic scheme uses several problematic assumptions. Indeed, they use the knowledge of the modelling of AoA measurement errors to generate probability density functions; moreover, they also use the knowledge of the (maximum) transmission range of sensors. We can also note that each node needs to merge several times probability density functions, and several exchanges of messages including these pdf, thus leading to important communication and computational costs. This method would be very difficult to use in practice, and would also be really less accurate than simulation results presented in [8] because of previous assumptions.

AT-Angle. In [9], authors present the AT-Angle localization method. After an initialization step where anchor nodes broadcast their positions in the whole network, each sensor collects its distances to anchor nodes. Then each node deduces an estimation of its position using sensors maximum transmission range r : a sensor node located at h hops from an anchor node deduces that it is inside the disk of radius $r \times h$ and outside the disk of radius r. Sensors positions are then improved using geometric relations between nodes and AoA measurements. A node whose estimated position error is below a given threshold becomes an estimated anchor, and then communicates its position to improve the localization accuracy.

Results regarding accuracy are interesting, however this method is particularly sensitive to measurement errors. Moreover, authors also use the knowledge of the (maximum) transmission range which would lead to many difficulties and less accuracy in practice.

3 ADNL-Angle

Let's consider a WSN, with a set of nodes with unknown geographical positions and a smaller set of anchor nodes whose positions are known. We consider sensor nodes with AoA technology which enables them to obtain angular information with neighboring nodes. Thus, we are working on the anchor and angle based distributed multi-hop localization problem.

Many different ways can be used to solve the localization problem. However, we believe that a localization method should not use strong assumptions such as the knowledge of the maximum transmission range of sensor nodes (in the Unit Disk Graph communication model), or as the angular measurement error model and its parameters.

We believe that broadcasting anchor nodes positions in the whole network is not needed unless the number of anchor is very low. Indeed, a very distant anchor does not provide a lot of information to a sensor node.

Our method consists in using topological knowledge in the k-hop neighborhood with measured angular information between neighbor nodes. In this paper, we provide simulation results using 2-hop knowledge and this seems to provide satisfying results. Moreover, limiting sensors knowledge at 2-hop is more suitable because of the needed exchange of messages to obtain this information. We can also note that 2-hop knowledge is also often needed by communication algorithms, such as Connected Dominating Sets construction algorithms for example [11].

The main principle of our localization scheme, named ADNL-Angle, can be divided in three steps :

1. Sensors collect information on their neighborhood: nodes in their 1-hop neighborhood with AoA information and anchor nodes positions. Then, they forward this data and collect information on their 2-hop neighborhood.
2. Nodes with enough anchors inside their (2-hop) neighborhood are able to compute their positions using a new force-based algorithm using AoA information, and then become estimated anchors.
3. A delaying mechanism is added to postpone the position computation on nodes which are harder to locate.

Definition 1. *An **estimated anchor** is a node whose location is initially not known and which will act as a real anchor node as soon as it computes its position.*

3.1 Computing Positions Using the Force-Based Algorithm

We consider that the 2-hop neighborhood discovery (step 1.) is done thanks to several exchanges of $HELLO$ messages with additional information between neighbor nodes.

Let's see more precisely how a sensor node determines its position in ADNL-Angle (step 2.). Each node with at least two anchors or estimated anchors in its 2-hop neighborhood can run the force-based algorithm; other nodes wait until some of their neighbors become estimated anchors.

Let's consider I, a node with enough anchors inside its neighborhood. I has partial knowledge of the network restricted to its 2-hop neighborhood, and thus I is authorized to estimate its own position. I locally runs the following algorithm so as to determine its position :

- I computes hop-distances between nodes in its neighborhood;
- I virtually places each anchor (or estimated anchor) at its known position;

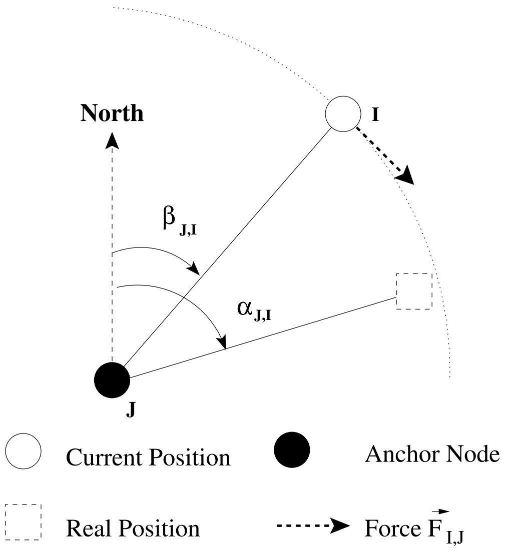


Fig. 3. Force exerted by node J on node I

- Non-located nodes are placed between their two closest anchor nodes;
- In a first time, I executes the angular force-based algorithm using only forces related to anchor nodes;
- Then, the algorithm is executed again, but this time all forces are considered between neighbor nodes, virtually moving them towards their final position;
- At the end of the algorithm, I retrieves its position and becomes an estimated anchor.

I internally builds a representation of its neighborhood, and then, the force-based algorithm tries to shift non-located nodes so as to satisfy angular relationship between neighbor nodes. For a given number of iterations, or until the target node no longer moves, sensors are virtually shifted at each iteration towards (normally) their real position.

Figure 3 illustrates this mechanism. During each iteration, node I is virtually shifted so as to satisfy angular relationship with node J. After the moving due to the force, node I virtual position is closer to its real position.

Forces applied on nodes depend on the differences between the virtual angles (in the local representation of the subgraph) denoted $\beta_{I,J}$, and the ones related to AoA measurements denoted $\alpha_{I,J}$.

Let $\vec{v}_{I,J}$ be the unit vector from I to J, and $\vec{p}_{I,J}$ the unit vector obtained with a 90^o counterclockwise rotation of $\vec{v}_{I,J}$. $\vec{F}_{I,J}$, the force applied on I by J is given by :

$$\vec{F}_{I,J} = (\alpha_{I,J} - \beta_{I,J}) \times \vec{p}_{I,J} \tag{1}$$

Let $N(I)$ be all neighboring nodes for I. At the end of each iteration, the sum of all forces applied on I by its neighbors provides the resultant force on node I :

$$\vec{F_I} = \sum_{K \in N(I)} \vec{F}_{I,K} \tag{2}$$

Let i be the current number of previously done iterations. At each iteration, non-located nodes are shifted according to forces exerted on it. The $(i+1)^{th}$ position of node I is computed as this :

$$\begin{cases} x_I^{i+1} = x_I^i - \sum\limits_{K \in N(I)} (sin(\alpha_{I,K} - \beta_{I,K}^i) \times sin(\beta_{I,K}^i)) \\ y_I^{i+1} = y_I^i + \sum\limits_{K \in N(I)} (sin(\alpha_{I,K} - \beta_{I,K}^i) \times cos(\beta_{I,K}^i)) \end{cases} \tag{3}$$

As said before, forces are exerted until a given number of iterations MAX is reached, or until positions no longer move. More formally, our force-based algorithm stops if one of the following condition is true :

$$\begin{array}{l} dist(x_I^{i-1}, y_I^{i-1}, x_I^i, y_I^i) \leq \epsilon \\ i \geq MAX \end{array} \tag{4}$$

Our observations lead us to set MAX value proportional to the number of involved nodes.

After the last iteration, the node which runs the force-based algorithm gets its position, becomes an **estimated anchor**, and communicates this position to its neighborhood. This allows non-located nodes with not enough anchors in their neighborhood to compute their positions, or to increase the knowledge of nodes which have not started the localization process.

Remark. When applying the force-based algorithm, anchor nodes and estimated anchors are not affected by virtual forces, they remain static at their estimated or accurate position. Moreover, anchor nodes do not need to apply the localization algorithm as they already know their positions.

3.2 Delaying the Computation of Localization

Using angular information, a non-located sensor node needs at least to collect the positions of two anchor nodes in order to estimate its position. However we can note that some nodes are harder to locate than other. In order to increase the accuracy of the localization process, we propose to add a delaying mecasnism in ADNL-Angle: some nodes will postpone the computation of their positions.

Various hypotheses can be made to determine how to set this delay:

- It is easier to locate nodes with many anchor nodes in their neighborhood, mainly to compensate angle measurement errors;
- Anchor nodes which are closer from a non-located node provide more information than distant ones.

We have studied various criteria and ways to combine them, and we have chosen the following formula to define the time to wait (TTW) until a node, denoted I, with at least two anchor nodes in its neighborhood starts the localization process:

$$TTW = \frac{c}{|A_1(I)| + \frac{|A_2(I)|}{w}} \qquad (5)$$

with $|A_1(I)|$ and $|A_2(I)|$ respectively the number of 1-hop and 2-hop anchor nodes in node I neighborhood. c is a constant whose value has been set empirically and regarding simulation parameters. In practice, c value depends on the time needed for a node to compute its position. The weighting coefficient w has been set empirically to 10 to decrease significantly the importance of 2-hop anchor nodes.

When I has at least two anchor nodes in its neighborhood, it computes its TTW regarding equation (5), and starts a timer T. While $T < TTW$, I waits in order to collect possibly more information; during the waiting, if I receives a message providing the position of a newly located node, I recomputes its TTW. As soon as $T \geq TTW$ node I runs the angular force-based algorithm, retrieves its position and communicates it to its neighbors. This strategy significantly improves results regarding accuracy at the expense of delay.

4 Performance Analysis

In this section we analyze the impact of several parameters on ADNL-Angle localization algorithm. We also compare our results with the probabilistic scheme [10]. We have made these simulations using WSNET simulator [4].

4.1 Simulation Parameters

In a first time, we consider two different kind of deployment strategies, both inside a 1000×1000 square area. The two different ways to deploy nodes in the area are the following : one consists in a random deployment inside the square area according to an uniform distribution, and the other one is a 10×10 noisy grid placement. Two examples of deployments are shown in Figure 4.

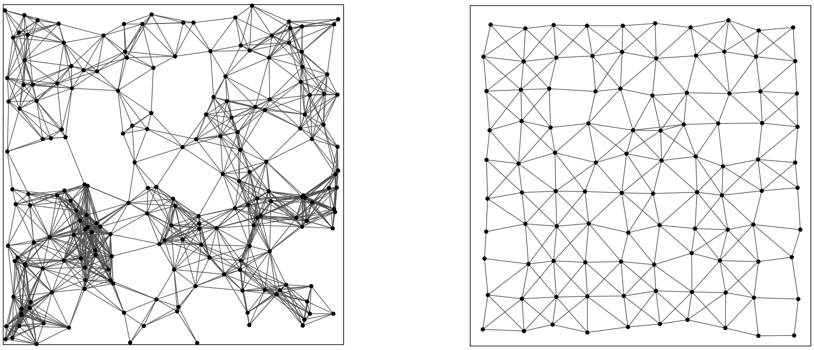
The number of nodes is fixed for each considered topology: 200 nodes for the random uniform deployment, and 100 for the grid one. The transmission range is set to obtain different connectivity levels, and is chosen from 125 to 250 with an increment of 25.

As in [6,10], we have chosen a Gaussian noise model for the AoA measurements; that is to say, if the real angle between nodes I and J (and North) is $\theta_{I,J}$, then the measured angle is provided thanks to a normal distribution:

$$\alpha_{I,J} = \mathcal{N}(\theta_{I,J}, \sigma^2) \qquad (6)$$

with σ the chosen angle measurement error parameter (the standard deviation of the normal distribution). If we set, for example, the standard deviation σ to $\frac{\pi}{8}$, then 95% of the measurements will be in the interval $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

Gaussian noise is also added to model error in grid placements, using the same method than the one in [2], and in order to obtain more realistic deployments.



a) b)

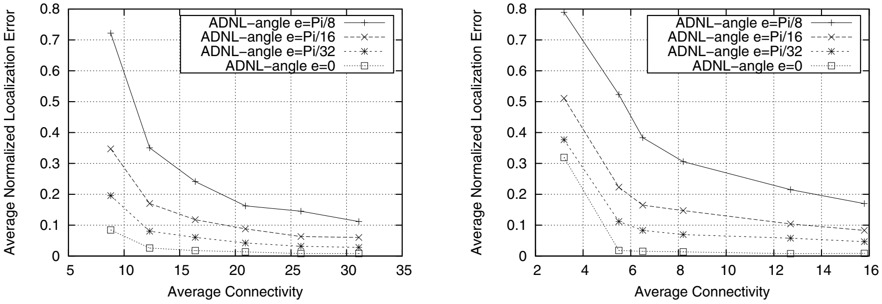
Fig. 4. Two different network topologies with transmission range set to 150. a) 200 nodes and random uniform deployment in a square area. b) 100 nodes and grid deployment in a square area with placement error.

In our simulations, and to obtain comparable results, localization accuracy is normalized using the maximum transmission range R. One hundred simulations have been made for each datum point.

4.2 Simulation Results for ADNL-Angle

We analyze in this section simulation results using previously described parameters.

The average number of neighbors has an important impact on the performance of most of localization schemes. Figures 5a) and b) show the impact of the network density when considering respectively random deployments or noisy grid placements. The different lines represent simulation results with various AoA measurement errors between 0 and $\frac{\pi}{8}$. Using previously described parameters we obtain an average degree between 8.8 and 31.1 with the first considered topology and between 3.2 and 15.8 with the second one.



a) Random deployment b) Noisy Grid placement

Fig. 5. Normalized localization error as a function of average connectivity using both considered topologies, and for various AoA measurement errors

When considering both topologies, and as it was expected, when the average connectivity is increased this leads to more accurate localization results. For example, using random deployments (Fig. 5a)), the average localization error goes from $35\%R$ to $6\%R$ when considering the measurement noise parameter $\sigma = \frac{\pi}{16}$. We can also note that when the average connectivity is increased, the average localization error tends to stabilize: excepted when considering large measurement errors, if we consider networks with density higher than 12, the gain regarding localization accuracy is not very important.

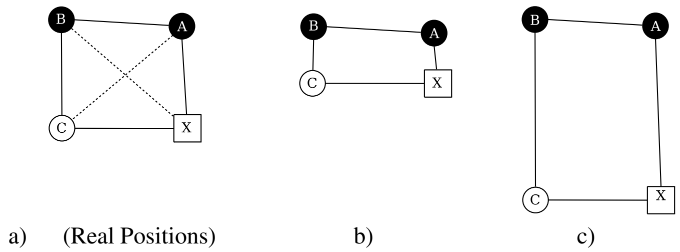
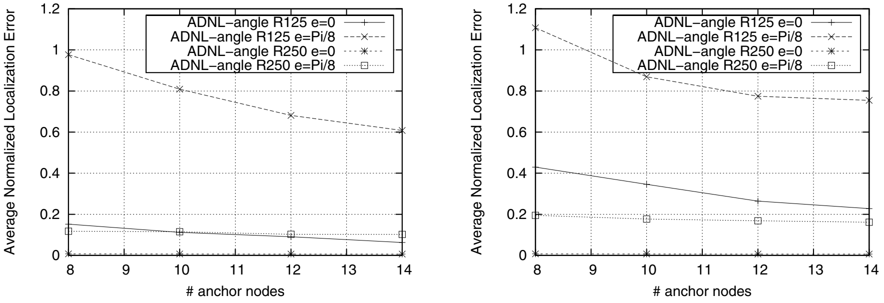


Fig. 6. Node X is hard to locate even with accurate AoA measurements. The 3 graphs above respect all angular measurements.

Nearly the same conclusions can be drawn using both considered topologies. However, with the lowest considered density using the noisy grid placement, results are particularly less accurate than when considering a higher connectivity, and this even without any measurement errors. Indeed, nodes are still able to locate themselves, but initially only a few of them know two anchors in their 2-hop neighborhood. Because of the chosen transmission range (i.e. the connectivity level), the case presented in Fig. 6a) happens very often. Even if node X knows the topology presented in the figure, the positions of anchor nodes A and B, and the angular measurements without error between all neighbor nodes, it is really difficult to determine its position. With such knowledge, all the graphs presented in Fig. 6 respect provided angular measurements. To obtain more accurate results, one possible solution is to add initially more anchor nodes. However, Fig. 5b) also shows that thanks to an increase of the transmission range the localization accuracy goes from more than $30\%R$ to nearly $1\%R$. The main explanation of this important gap, is the presence of new communication links which removes ambiguities. These links are represented by dashed lines on Figure 6a).

According to the four different curves in Figure 5a) and in Figure 5b), we can observe the impact of the Angle of Arrival measurement errors. In our simulations, we have used AoA measurements without errors ($\sigma = 0$), and with more and more large errors : $\sigma = \{\frac{\pi}{32}, \frac{\pi}{16}, \frac{\pi}{8}\}$. We can note that, as expected, larger measurement errors lead to less accurate localization results. However the impact of measurement errors is particularly reduced when density is increased; in networks with the highest considered density this impact is nearly not noticeable. Indeed, while the average localization error equals to $8\%R$ without measurement errors and equals to $72\%R$ if $\sigma = \frac{\pi}{8}$ in the lowest considered densities, it reaches then respectively $1\%R$ and $11\%R$.



a) b)

Fig. 7. Normalized localization error as a function of the number of anchor nodes using both considered topologies

Figure 7 shows the impact of additional anchor nodes on ADNL-Angle localization scheme. When transmission range is set to 250, that is to say with a high connectivity level, additional anchor nodes do not provide an increase of the accuracy when using ADNL-Angle, and this even with large transmission range. When we consider the lowest studied density (i.e. transmission range set to 125), the impact of additional anchor nodes is clearly more significant. The benefit of additional anchor nodes is further significant when considering large measurement errors. The previously described difficulties to locate nodes with the lowest density and noisy grid placement, are partially corrected with additional anchor nodes.

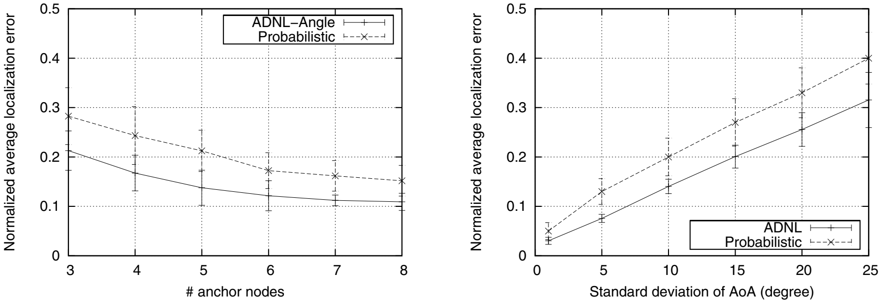
This analysis of simulation results shows that ADNL-Angle method is able to cope with measurement errors, and even to limit the impact of large measurement errors. Moreover, ADNL-Angle benefits from additional anchor nodes when large measurement errors are considered and/or when considering low connectivity levels.

4.3 ADNL-Angle vs. Probabilistic

We compare in this section our proposed scheme ADNL-Angle, with the probabilistic scheme presented in [10]. Simulation are still made with WSNet but with the same parameters than in [10]. We compare our work to the probabilistic scheme because it is as far as we know the most accurate localization method with AoA information.

In Figure 8, we present simulation results in network with an average connectivity equals to 10. Figure 8a) compares the accuracies of the ADNL-Angle and the probabilistic schemes with various number of anchor nodes with the angular measurement error parameter σ equals to $\frac{\pi}{18}$. ADNL-Angle outperforms the probabilistic scheme in these simulations. Whatever the considered number of anchor nodes is, the gap between both methods is around $5\%R$, both methods benefit equally from additional anchor nodes.

In Fig. 8b), we study the impact of AoA measurement errors on both methods. As expected, when the measurement errors are larger, both methods obtain less accurate results. ADNL-Angle also outperforms the probabilistic scheme in these simulations.



a) b)

Fig. 8. Comparison of ADNL-Angle with the probabilistic scheme. a) Average localization accuracy regarding the number of anchors. b) Average localization accuracy as a function of AoA measurement errors.

Moreover, the gap between both localization methods slightly tends to be increased when the measurement noise becomes higher.

Beyond these results showing the accuracy of ADNL-Angle, our method also uses really less strong assumptions. Indeed, contrary to the probabilistic scheme, ADNL-Angle does not use the knowledge of the maximum transmission range which would not be usable in practice. Moreover, in the probabilistic localization scheme, the algorithm uses the measurement error model to localize sensor nodes. As this model is clearly environment and hardware dependent, it is not suitable to use it to determine sensors positions. These strong assumptions make us believe that in practice results for the probabilistic localization scheme will probably be particularly less accurate.

Regarding computational cost(combination of several probability density function), and to the number of exchanged messages, the probabilistic scheme is more costly. In ADNL-Angle, the number of exchanged messages is $2n + (n - m)$ with n the number of nodes in the network, and m the number anchor nodes : $2n$ for the neighborhood discovery step, and $(n - m)$ to communicate newly located sensors positions. It is also important to note that, as said previously, information which are communicated can also be used to construct for example communication backbones. In the probabilistic scheme the number of exchanged message is higher : in their simulation results they suggest that each anchor node broadcasts a message with a TTL equals to 4. The use of the probabilistic scheme leads to an important number of exchanged messages and a lot of data in anchors packets.

For all these reasons, we believe that ADNL-Angle is more suitable to solve the localization problem, and this, regarding accuracy and scalability.

5 Conclusion

In this paper, which deals with the multi-hop distributed localization problem in WSN with angle of arrival information, we present ADNL-Angle, an original and accurate localization method.

After a neighborhood discovery step, nodes with enough anchors in their 2-hop neighborhood compute their positions according to an original angular force-based algorithm. Newly located nodes become estimated anchors and then propagate their new position so as to enable other nodes to locate themselves. ADNL-Angle also uses a delaying process which postpones the position computation on nodes which are most difficult to locate, providing accurate localization information for almost all sensor nodes.

The simulation results show the impact of several parameters such as AoA measurement errors, the number of anchor nodes and various network topologies. Results show the accuracy of ADNL-Angle even with a small number of anchor nodes or potentially large AoA measurement errors. Our localization scheme also outperforms methods such as the APS-AoA method or the accurate probabilistic scheme, and this without strong assumptions such as the knowledge of the maximum transmission range or of the distribution of measurement errors.

Some additional work could be made to emphasize advantages and drawbacks of our method, such as analyzing the behavior of our method in more realistic networks. We are testing a method to reduce the number of nodes involved during the angular force-based step: it consists in removing useless nodes, or nodes which may interfere with an accurate localization. Preliminary results show that this method leads to less computations on nodes and slightly more accurate results.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Dept, Y.S., Shang, Y.: Improved mds-based localization. In: INFOCOM, pp. 2640–2651 (2004)
3. Giordano, S., Stojmenovic, I., Blazevic, L.: Position based routing algorithms for ad hoc networks: a taxonomy. In: Ad Hoc Wireless Networking, pp. 103–136. Kluwer, Dordrecht (2003)
4. Ben Hamida, E., Chelius, G., Gorce, J.M.: On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks Using Simulations. In: 11th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2008), IEEE/ACM (2008)
5. Mao, G., Fidan, B., Anderson, B.D.O.: Wireless sensor network localization techniques. Computer Networks 51(10), 2529–2553 (2007)
6. Niculescu, D., Nath, B.: Ad hoc positioning system (APS) using AOA. In: INFOCOM, vol. 22, pp. 1734–1743 (2003)
7. Patwari, N., Ash, J.N., Kyperountas, S., Hero, A.O., Moses, R.L., Correal, N.S.: Locating the nodes: cooperative localization in wireless sensor networks. IEEE Signal Processing Magazine 22(4), 54–69 (2005)
8. Peng, R., Sichitiu, M.: Angle of arrival localization for wireless sensor networks. In: Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), vol. 1, pp. 374–382 (2006)

9. Saad, C., Benslimane, A., Konig, J.C.: At-angle: A distributed method for localization using angles in sensor networks. In: IEEE Symposium on Computers and Communications (ISCC) (2008)
10. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. Wireless Networks 8(2/3), 153–167 (2002)
11. Wu, J., Li, H.: On calculating connected dominating set for efficient routing in ad hoc wireless networks. In: DIALM 2099: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7–14. ACM, New York (1999)

Planning and Deploying Long Distance Wireless Sensor Networks: The Integration of Simulation and Experimentation

Marco Zennaro[1], Antoine Bagula[2],
David Gascon[3], and Alberto Bielsa Noveleta[3]

[1] Royal Institute of Technology - KTH, Stockholm, Sweden
mzennaro@ictp.it
[2] University of Cape Town, Cape Town, South Africa
bagula@cs.uct.ac.za
[3] Libelium, Zaragoza, Spain
d.gascon@libelium.com, a.bielsa@libelium.com

Abstract. Wireless sensor networks allow unprecedented abilities to observe and understand large-scale, real-world phenomena at a fine spatial-temporal resolution. Their application in Developing Countries is even more interesting: they can help solve problems that affect communities. One of the limitations of current wireless sensors is the communication range, with most devices having 100 meters as maximum range. In contrast, many applications require long-range wireless sensor network where nodes are separated by large distances, giving the advantage of being able to monitor a large geographic area. In this paper we will present the results of an integrated approach combining a planning step using simulations and an experimental step carried out using off-the-shelf equipment over distances ranging from 600m to 12km. The results reveal that the simulation results agree with experimentation and show that long distance wireless sensor networks (LDWSN) are possible and that the quality of these links is high. Finally, we discuss the relative efficiency of our solution in terms of range compared to other wireless sensor networks.

Keywords: Waspmote, Long Distance WSN, WSN, Frequency, ICT4D.

1 Introduction to LDWSN

Wireless Sensor Networks (WSNs) are a branch of ICT technologies which have been widely deployed in industrialized regions in many applications to achieve environment observation, healthcare and medical monitoring, home security, machine failure diagnosis, chemical/biological detection and plant monitoring. WSNs are deployed in large numbers of tiny sensor nodes, each node being regarded as a low power and cheap computer that can perform sensing, computation and communication. The sensor nodes communicate wirelessly and are

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 191–204, 2010.
© Springer-Verlag Berlin Heidelberg 2010

deployed in three forms : (1) Sensor node used to sense the environment, (2) Relay node used as relay for the information sensed by other nodes and (3) Sink node acting as base station with higher energy to transmit the sensed information to a local or remote processing place.

Traditional sensor technology allows the deployment of wireless sensor networks in a 1-to-m fashion where all the nodes, excepted the sink node, sense their environment and send the collected information to the base station for further processing. As currently deployed, WSNs are based on a multi-hop model allowing these networks to 1) span distances much larger than the transmission range of a single node 2) adapt to network changes, for example, by routing around a failed node using a different path in order to improve performance and 3) use less transmitter power as a result of the shorter distance transmission mode enabled by the potential to achieve local communication between neigbhor nodes.

In many practical applications that require sensor monitoring over long distances such as farming or water quality monitoring in developing regions where environmental conditions such as temperature, soil moisture and other levels of water troughs need to be measured at widely separated locations, the short wireless range provided by WSNs may be become a limiting factor in terms of both cost since multi-hop routing over long distances may require many sensors and coverage as the short range sensors can cover only a few hundred of meters. However, while being resolved for Wi-FI technology [1], the problem of range limitation has been only poorly addressed by the research community. The work presented in [2] proposes a sensor network in Australia where the range of a mote has been extended to 300 m, a distance that does not meet long distance application requirements. Motivated by the fact that the directional antenna is an established technology that has been proven effective in improving the radio link quality, the work presented in [3] proposes the integration of radio communication technology to not only compensate for the higher path loss intrinsic of shorter wavelengths but also to ensure higher link quality and to implement a form of antenna diversity. A switched beam directional antenna operating in the 2.4GHz ISM band (e.g. using the IEEE 802.15.4 standard) with dimensions, cost and complexity constraints comparable to those of commercially available sensor nodes is presented in [3]. Used outdoors, the antenna extends the communication range from 140m to more than 350m, while indoors it suppresses the interference due to multipath fading by reducing the signal variability of more than 70%. The antenna also reveals interference suppression from IEEE 802.11g systems and can be used as a form of angular diversity useful to cope with the variability of the radio signal. Similarly, the work presented in [4] considers the use of switched beam directional antennas in wireless sensor networks. Using comparison with an existing solution based on S-MAC, the paper shows that the introduction of directional antennas reduces interference, transmission delay and flooding and consequently improves throughput and energy consumption. As presented by [5], a long-range ad-hoc wireless sensor network is proposed where a radio propagation model is used to enhance the range of wireless nodes. Using this model, distances of up to 10 kilometers are reached using non-directional antennae by having the radio transceiver of the Berkeley Mote replaced with a lower frequency, higher power unit operating in the 40.66-41.00 MHz frequency band with a maximum power of 1 W EIRP.

Simulations have been used in many research works to shorten development time by having all of the variables of a real system under the control of the designer, allowing better testing and debugging for example. However, it often happen that by making simplicative assumptions on the system requirements, the designers unintentionally introduce biases into the model wich affect the validity of the simulation such as leading to unrealistic behaviours or behaviours that do not map to real world behaviour. This paper revisits the problem of long distance wireless sensor network deployment in developing regions by (1) assessing the relevance of using simulation in planning long distance links and (2) proposing a long distance wireless sensor network (LDWSN) deployment as case study. The main contributions of our paper are twofold. First, using the radio mobile simulation software, we evaluate the accuracy of using a simulation package that builds around real maps to preplan long distance wireless sensor links. Secondly, we present a case study of a long distance wireless sensor network deployment using the Waspmote [6] technology with experiments conducted in harsh conditions.

The remainder of this paper is organized as follows. Section 2 describes the Radio Mobile simultaion software and present the simulation results obtained when planning long distance WSN (LDWSN) links. Section 3 describes the experiments conducted in harsh conditions in the Los Monegros Desert near Huesca in Spain and compare the experimental results with the simulative results. Section 4 discusses the relevance of LDWSN in developing countries and compare some of the features of the Waspmote to other WSN technologies in terms of long distance deployment. Our conclusions are presented in section 5.

2 Simulation of the Links

To check if radio links were feasible, we decided to use Radio Mobile [7], a free tool for the design and simulation of wireless systems. It predicts the performance of a radio link by using information about the equipment and a digital map of the area. Radio Mobile uses a digital terrain elevation model for the calculation of coverage, indicating received signal strength at various points along the path. It automatically builds a profile between two points in the digital map showing the coverage area and first Fresnel zone. During the simulation, it checks for line of sight and calculates the Path Loss, including losses due to obstacles. The software calculates the coverage area from the base station in a point-to-multipoint system. It works for systems having frequencies from 100 kHz to 200 GHz. It is based on the ITS (Longley-Rice) propagation model. Digital elevation maps (DEM) are available for free from several sources, and are available for most of the world. DEMs do not show coastlines or other readily identifiable landmarks, but they can easily be combined with other kinds of data (such as aerial photos or topographical charts) in several layers to obtain a more useful and readily recognizable representation. The digital elevation maps can be merged with scanned maps, satellite photos and Internet map services (such as Google Maps) to produce accurate prediction plots.

2.1 Candidate Locations

To test the feasibility of long wireless sensor links, it is necessary to find a location with an unobstructed line-of-sight between two sites. As the distance between sites increases, higher elevation is required at both ends.

For our experiments we selected 10 sites in the Los Monegros Desert near Huesca, Spain. Los Monegros is located within the provinces of Zaragoza and Huesca. The area is prone to chronic droughts, and much of the area is semi-desert. The climate is semiarid, with scare rainfall and high temperatures in the fall. Its maximum elevation is 822 meters, which can be found on the mountain called Oscuro. The lack of human activity ensured an interference-free environment. We did not carry out a site survey when selecting the candidate locations. The localization of the testbed is depicted by Figure 1. We selected ten spots in the area, which allowed us to establish six links. We considered both links with line of sight (LOS) and those with non line of sight (NLOS) as sensor networks are meant to be deployed in different environments as such trees, buildings, forests, etc. Table 1 shows the positions of the sites, the names of the 6 links and their types.

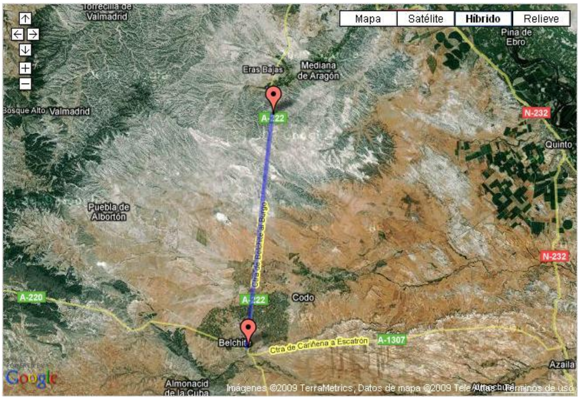


Fig. 1. Experimental setup in Los Monegros

2.2 Simulation Data

In addition to the locations, more data is required to run a simulation in Radio Mobile. The characteristics of the equipment, type of antennas and elevation above the ground need to be given as inputs to the software. For the experiments, we used Waspmote devices produced by Libelium, equipped with seven different 802.15.4/ZigBee transceivers. Waspmotes are built around XBee transceivers

Table 1. Position of candidate sites

Position 1	Position 2	Distance	Link Number	Link Type
41.377708N 0.732896W	41.380916N 0.732873W	356m	Link 1	LOS
41.375178N 0.733515W	41.380916N 0.732873W	639m	Link 2	LOS
41.324061N 0.740585W	41.380916N 0.732873W	6363m	Link 3	LOS
41.316091N 0.742146W	41.424445N 0.725913W	12136m	Link 4	LOS
41.390453N 0.731088W	41.401531N 0.729388W	1238m	Link 5	NLOS
41.394053N 0.731088W	41.424445N 0.725913W	3810m	Link 6	NLOS

which provide flexibility in terms of multiplicity of operating power, protocols, and operating frequencies as depicted by the XBee features in Table 2. Other Waspmote characteristics include (1) minimum power consumption of the order of 0.7 μA in the Hibernate mode (2) flexible architecture allowing extra sensors to be easily installed in a modular way, and (3) the provision of GPS, GPRS and SD card on board. Furthermore, Waspmotes are powered with a lithium battery which can be recharged through a specially dedicated socket for the solar panel; this option is specially interesting for deployments in Developing Countries where power supply is not stable.

Table 2. Characteristics of XBee Transceivers

Model	Protocol	Frequency	TX power	Sensitivity	Label
XBee-802.15.4	802.15.4	2.4 GHz	1 mW	-92 dB	Dev1
XBee-802.15.4-Pro	802.15.4	2.4 GHz	63 mW	-100 dB	Dev2
XBee-ZB	ZigBee-Pro	2.4 GHz	2 mW	-96 dB	Dev3
XBee-ZB-Pro	Zigbee-Pro	2.4 GHz	50 mW	-102 dB	Dev4
XBee-868	RF	868 MHz	315 mW	-112 dB	Dev5
XBee-900	RF	900 MHz	50 mW	-100 dB	Dev6
XBee-XSC	RF	900 MHz	100 mW	-106 dB	Dev7

Table 2 describes the characteristics of the XBee transceivers. As described by the table, these transceivers will be referred to in the rest of the paper as Dev1,. . .,Dev7. The XBee transceivers are equipped with SMA antenna connectors so an external antenna can be used. For the simulation we considered omnidirectional antennas, with a gain of 2dBi and 5dBi in 2.4GHz and in 868/900MHz. Antennas with such gain can be commonly found on the market and do not require special alignments. The links used vertically polarized antennas.

The height from ground is assumed to be 2m, as this is the maximum height of a tripod.

2.3 Fresnel Zone and Link Budget Calculation

When simulating a wireless link, one has to check two important parameters to determine if the link is possible or not: Fresnel zone and link margin. The Fresnel zone is an ellipsoid area around the direct line between two communicating devices. It is widely known that the radius of the fresnel zone at its widest point is expressed by

$$r = 17.32\sqrt{zd/4f} \tag{1}$$

where z is the zone number with the value $z = 1$ referring to the first Fresnel Zone, f is the frequency used (expressed in MHz) and d is the exact distance (in meters) between the receiver and transmitter. If this area were partially blocked by an obstruction, e.g. a tree or a building, the signal arriving at the far end would be diminished. When building wireless links, we therefore need to be sure that these zones be kept free of obstructions. Of course, nothing is ever perfect, so usually in wireless networking we check that about 60 percent of the radius of the first Fresnel zone should be kept free.

For example, let's calculate the size of the first Fresnel zone in the middle of our longest, 12km link, transmitting at 2.4 GHz:

$$r = 17.32\sqrt{12000/4 * 2400} = 19.36m \tag{2}$$

We need to have at least one elevated point to be able to have a 12km link using the 2.4 GHz frequency. Radio Mobile takes care of calculating the Fresnel zone, once the positions and the equipment characteristics have been entered in the software.

In order to have a communication between two wireless devices, the radios require a certain minimum signal to be collected by the antennas and presented to their input socket. Determining if the link is feasible is a process called link budget calculation. Whether or not signals can be passed between the radios depends on the quality of the equipment being used and on the diminishment of the signal due to distance, called path loss.

As suggested by [9], to perform the link budget calculation, one must know the characteristics of the equipment being used and evaluate the path loss. Adding up all the gains and subtracting all the losses gives:

$$\begin{aligned}
TransmitPower \\
+TransmittingAntennaGain \\
+ReceivingAntennaGain \\
= \\
TotalGain \\
-FreeSpaceLoss \\
= \\
ExpectedReceivedSignalLevel \\
-ReceiverSensitivity \\
-AntennaCableLoss \\
= \\
LinkMargin
\end{aligned}$$

The Transmitting Antenna Gain, Receiving Antenna Gain and Receiver Sensitivity are dependent on the hardware used. The Free Space Loss depends on the frequency used and on the distance. The longer the distance, the higher the Free Space Loss.

On a given path, the variation in path loss over a period of time can be large, so a certain margin (difference between the signal level and the minimum received signal level) should be considered.This margin is the amount of signal above the sensitivity of radio that should be received in order to ensure a stable, high quality radio link during bad weather and other atmospheric disturbances. A margin of 10 to 15 dB is fine.

Radio Mobile is able to calculate the link margin for a specific link, given the positions and the characteristics of the equipment used.

2.4 Simulation Results

The equipment we wanted to use for the experiment consisted of seven different XBee cards, each one with two possible antennas. We thus had fourteen different

Table 3. Link Margin and Fresnel Zone Clearance at 2dBi

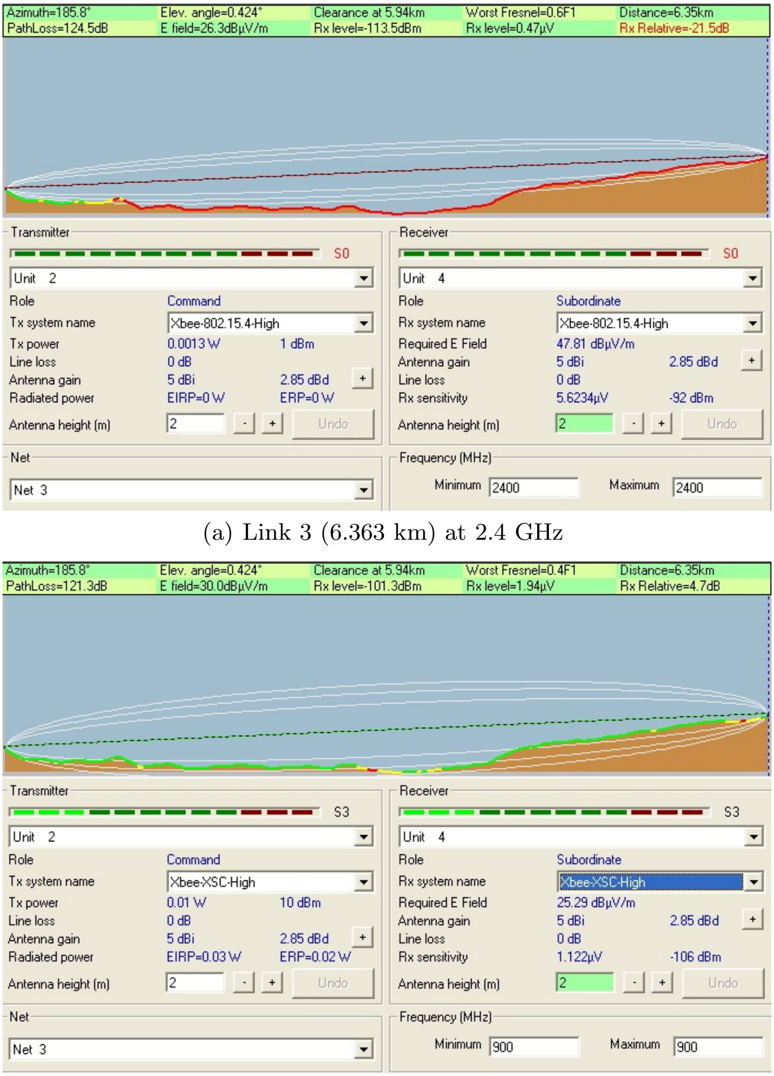
Xbee card at 2dBi	Antenna	Link 1	Link 2	Link 3
XBee-802.15.4	2dBi	2.6,1.0F1	-4.7,0.7F1	-28.5,0.6F1
XBee-802.15.4-Pro	2dBi	18.6,1.0F1	11.3,0.7F1	-12.5,0.6F1
XBee-ZB	2dBi	9.6,1.0F1	2.3,0.7F1	-21.5,0.6F1
XBee-ZB-Pro	2dBi	19.6,1.0F1	12.3,0.7F1	-11.5,0.6F1
XBee-868	2dBi	37.9,0.6F1	31.0,0.4F1	9.8,0.4F1
XBee-900	2dBi	17.6,0.6F1	10.8,0.4F1	-10.3,0.4F1
XBee-XSC	2dBi	26.6,0.6F1	19.8,0.4F1	-1.3,0.4F1

Xbee card at 2dBi	Antenna	Link 4	Link 5	Link 6
XBee-802.15.4	2dBi	-28.3,0.8F1	-24.3,0.1F1	-46.8,-0.5F1
XBee-802.15.4-Pro	2dBi	-12.3,0.8F1	-8.3,0.1F1	-30.8,-0.5F1
XBee-ZB	2dBi	-21.3,0.8F1	-17.3,0.1F1	-39.8,-0.5F1
XBee-ZB-Pro	2dBi	-11.3,0.8F1	-7.3,0.1F1	-29.8,-0.5F1
XBee-868	2dBi	8.5,0.5F1	15.4.3,0.1F1	-1.7,-0.3F1
XBee-900	2dBi	-11.6,0.5F1	-4.7,0.1F1	-22.1,-0.3F1
XBee-XSC	2dBi	-2.6,0.5F1	4.3,0.1F1	-13.0,-0.3F1

Table 4. Link Margin and Fresnel Zone Clearance at 5dBi

Xbee card at 5dBi	Antenna	Link 1	Link 2	Link 3
XBee-802.15.4	5dBi	9.5,1.0F1	2.3,0.7F1	-21.5,0.6F1
XBee-802.15.4-Pro	5dBi	24.6,1.0F1	17.3,0.7F1	-6.5,0.6F1
XBee-ZB	5dBi	15.6,1.0F1	8.3,0.7F1	-15.5,0.6F1
XBee-ZB-Pro	5dBi	25.6,1.0F1	18.3,0.7F1	-5.5,0.6F1
XBee-868	5dBi	43.9,0.6F1	37.0,0.4F1	15.8,0.4F1
XBee-900	5dBi	23.6,0.6F1	16.8,0.4F1	-4.3,0.4F1
XBee-XSC	5dBi	32.6,0.6F1	25.8,0.4F1	4.7,0.4F1

Xbee card at 5dBi	Antenna	Link 4	Link 5	Link 6
XBee-802.15.4	5dBi	-21.3,0.8F1	-17.3,0.1F1	-39.8,-0.5F1
XBee-802.15.4-Pro	5dBi	-6.3,0.8F1	-2.3,0.1F1	-24.8,-0.5F1
XBee-ZB	5dBi	-15.3,0.8F1	-11.3,0.1F1	-33.8,-0.5F1
XBee-ZB-Pro	5dBi	-5.3,0.8F1	-1.3,0.1F1	-23.8,-0.5F1
XBee-868	5dBi	14.5,0.5F1	21.4,0.1F1	4.3,-0.3F1
XBee-900	5dBi	-5.6,0.5F1	1.3,0.1F1	-16.1,-0.3F1
XBee-XSC	5dBi	3.4,0.5F1	10.3,0.1F1	-7.0,-0.3F1



Azimuth=185.8°	Elev. angle=0.424°	Clearance at 5.94km	Worst Fresnel=0.6F1	Distance=6.35km
PathLoss=124.5dB	E field=26.3dBµV/m	Rx level=-113.5dBm	Rx level=0.47µV	Rx Relative=-21.5dB

Transmitter S0

Unit 2
Role: Command
Tx system name: Xbee-802.15.4-High
Tx power: 0.0013 W 1 dBm
Line loss: 0 dB
Antenna gain: 5 dBi 2.85 dBd
Radiated power: EIRP=0 W ERP=0 W
Antenna height (m): 2

Receiver S0

Unit 4
Role: Subordinate
Rx system name: Xbee-802.15.4-High
Required E Field: 47.81 dBµV/m
Antenna gain: 5 dBi 2.85 dBd
Line loss: 0 dB
Rx sensitivity: 5.6234µV -92 dBm
Antenna height (m): 2

Net: Net 3

Frequency (MHz) Minimum: 2400 Maximum: 2400

(a) Link 3 (6.363 km) at 2.4 GHz

Azimuth=185.8°	Elev. angle=0.424°	Clearance at 5.94km	Worst Fresnel=0.4F1	Distance=6.35km
PathLoss=121.3dB	E field=30.0dBµV/m	Rx level=-101.3dBm	Rx level=1.94µV	Rx Relative=4.7dB

Transmitter S3

Unit 2
Role: Command
Tx system name: Xbee-XSC-High
Tx power: 0.01 W 10 dBm
Line loss: 0 dB
Antenna gain: 5 dBi 2.85 dBd
Radiated power: EIRP=0.03 W ERP=0.02 W
Antenna height (m): 2

Receiver S3

Unit 4
Role: Subordinate
Rx system name: Xbee-XSC-High
Required E Field: 25.29 dBµV/m
Antenna gain: 5 dBi 2.85 dBd
Line loss: 0 dB
Rx sensitivity: 1.122µV -106 dBm
Antenna height (m): 2

Net: Net 3

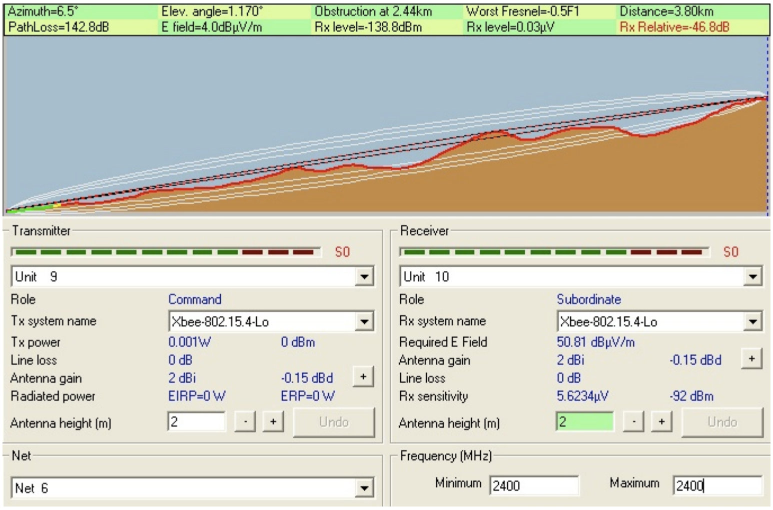
Frequency (MHz) Minimum: 900 Maximum: 900

(b) Link 3 (6.363 km) at 900 MHz

Fig. 2. Link 3 (6.363 km)

hardware solutions for each of the six links. It is not possible to report a graph of all the results, but Table 3 and 4 summarize the simulation results.

As can be seen from the simulation results, links behave differently according to the frequency used and to the output power. Longer links are only possible using lower frequencies (868 and 900 MHz), while 2.4 GHz is only usable for shorter links.



Azimuth=6.5° Elev. angle=1.170° Obstruction at 2.44km Worst Fresnel=-0.5F1 Distance=3.80km
PathLoss=142.8dB E field=4.0dBµV/m Rx level=-138.8dBm Rx level=0.03µV Rx Relative=-46.8dB

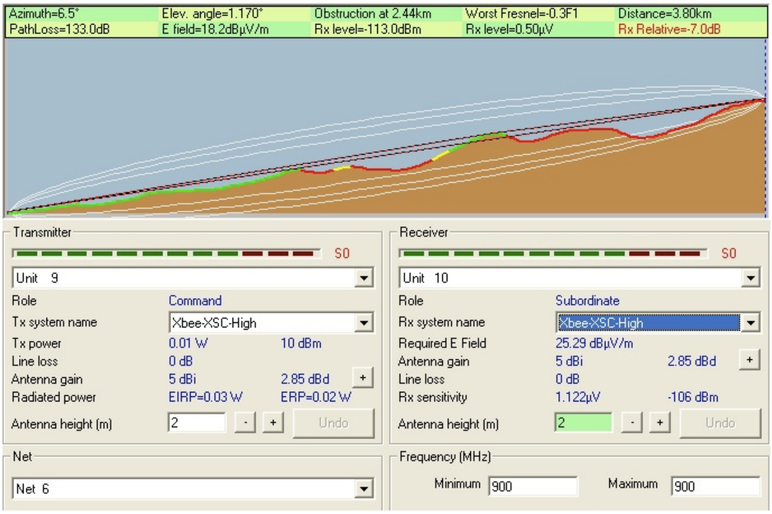
Transmitter S0
Unit 9
Role Command
Tx system name Xbee-802.15.4-Lo
Tx power 0.001W 0 dBm
Line loss 0 dB
Antenna gain 2 dBi -0.15 dBd
Radiated power EIRP=0 W ERP=0 W
Antenna height (m) 2

Receiver S0
Unit 10
Role Subordinate
Rx system name Xbee-802.15.4-Lo
Required E Field 50.81 dBµV/m
Antenna gain 2 dBi -0.15 dBd
Line loss 0 dB
Rx sensitivity 5.6234µV -92 dBm
Antenna height (m) 2

Net
Net 6

Frequency (MHz)
Minimum 2400 Maximum 2400

(a) Link 6 (3.810 km) at 2.4 GHz



Azimuth=6.5° Elev. angle=1.170° Obstruction at 2.44km Worst Fresnel=-0.3F1 Distance=3.80km
PathLoss=133.0dB E field=18.2dBµV/m Rx level=-113.0dBm Rx level=0.50µV Rx Relative=-7.0dB

Transmitter S0
Unit 9
Role Command
Tx system name Xbee-XSC-High
Tx power 0.01 W 10 dBm
Line loss 0 dB
Antenna gain 5 dBi 2.85 dBd
Radiated power EIRP=0.03 W ERP=0.02 W
Antenna height (m) 2

Receiver S0
Unit 10
Role Subordinate
Rx system name Xbee-XSC-High
Required E Field 25.29 dBµV/m
Antenna gain 5 dBi 2.85 dBd
Line loss 0 dB
Rx sensitivity 1.122µV -106 dBm
Antenna height (m) 2

Net
Net 6

Frequency (MHz)
Minimum 900 Maximum 900

(b) Link 6 (3.810 km) at 900 MHz

Fig. 3. Link 6 (3.810 km)

Figure 2 (a) and (b) show the Fresnel zone and link margin as presented by Radio Mobile for link number 3 at 2.4 GHz and 900 Mhz. The Fresnel zone is much larger at 900 MHz and the link margin is bigger.

Figure 3 (a) and (b) show the Fresnel zone and link margin as presented by Radio Mobile for link number 6 at 2.4 GHz and 900 Mhz. The two figures reveal a performance pattern similar to link 3 where the Fresnel zone is much larger and the link margin is bigger at 900 MHz.

3 Experiments

In October 2009 we performed the experiments in the Los Monegros Desert near Huesca, Spain, over a period of 3 days. We wanted to check if the experimental results were consistent with the simulation ones, and wanted to measure power consumption in a real-world environment. To test the link quality, we sent 100 packets of 90 Bytes each and counted how many packets were received to measure throughput. We also measured the RSSI level.

3.1 Experimental Results

Table 5 show the results of our tests. To check if the simulations give similar results compared to the experiments, we graphed the simulated link margin and

Table 5. Experimental performance

XBee features	Feature	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
	Protocol	802.15.4	802.15.4	Zigbee-Pro	ZigBee-Pro	RF	RF	RF
	Frequency (Hz)	2.4G	2.4G	2.4G	2.4G	868M	900M	900M
	TX power (mW)	1	63	2	50	315	50	100
	Sensivity(-dBm)	92	100	96	102	112	100	106
Throughput	Distance	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
2dBi	356m (LOS)	85%	100%	100%	100%	100%	100%	100%
	639m (LOS)	0%	100%	0%	100%	100%	100%	100%
	6363m (LOS)	0%	18%	0%	25%	100%	0%	80%
	12136m (LOS)	0%	0%	0%	0%	100%	0%	0%
	1239m (NLOS)	0%	0%	0%	0%	100%	0%	100%
	3810m (NLOS)	0%	0%	0%	0%	0%	0%	0%
5dBi	356m (LOS)	100%	100%	100%	100%	100%	100%	100%
	639m (LOS)	19%	100%	100%	100%	100%	100%	100%
	6363m (LOS)	0%	100%	0%	100%	100%	0%	100%
	12136m (LOS)	0%	0%	0%	0%	100%	0%	100%
	1239m (NLOS)	0%	0%	0%	0%	100%	0%	100%
	3810m (NLOS)	0%	0%	0%	0%	50%	0%	10%
RSSI(dBm)	Distance	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
2dBi	356m (LOS)	-94	-72	-84	-70	-70	-70	-70
	639m (LOS)		-91		-78	-70	-70	-70
	6363m (LOS)					-97		-94
	12136m (LOS)					-100		
	1239m (NLOS)							
	3810m (NLOS)					-77		
5dBi	356m (LOS)	-87	-70	-72	-70	-70	-70	-70
	639m (LOS)	-94	-70	-90	-70	-70	-70	-70
	6363m (LOS)					-80		-101
	12136m (LOS)					-97		-83
	1239m (NLOS)		-97		-83			-93
	3810m (NLOS)					-78		

Table 6. Power consumption

State	From OFF to ON	Time	From sleep to ON	Time
TX Unicast without encryption	890.82nAh	79.4ms	849.16nAh	76.4ms
TX Unicast with encryption	904.73nAh	79.36ms	863.07nAh	76.36ms
TX Broadcast without encryption	887.79nAh	78,7ms	846.13nAh	75.7ms
RX Broadcast without encryption	889.45nAh	78,6ms	847.79nAh	75.6ms
RX Unicast without encryption	825.52nAh	74ms	783.86nAh	71ms
RX Unicast with encryption	826.11nAh	73.92ms	784.45nAh	70.92ms
RX Broadcast without encryption	818.55nAh	73.4ms	776.89nAh	70.4ms
RX Broadcast with encryption	818.63nAh	73.4ms	776.97nAh	70.4ms

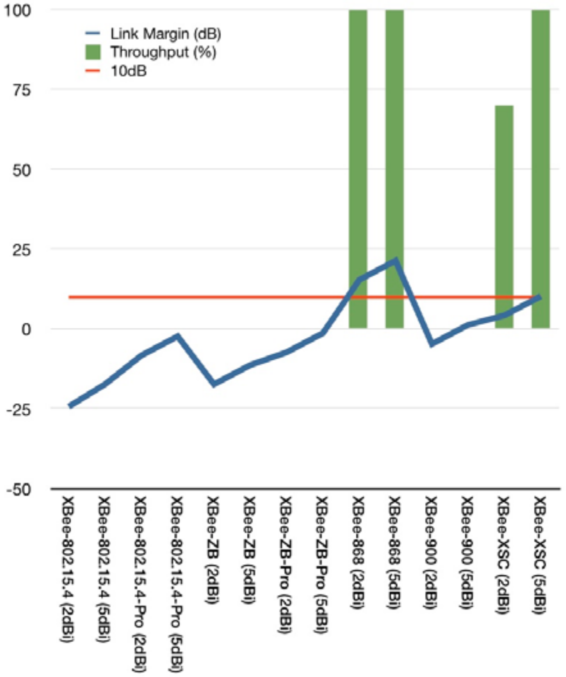


Fig. 4. Comparison between the simulated link margin and the measured throughput. Highlighted is the threshold value of 10 dB.

the measured throughput for link number 3 (1239 m). The results are shown in Figure 4. When the link margin is above 10 dB, then the link is possible and the throughput is high (70% up to 100%). When it is lower than 10 dB, then the link is not possible. From the experimental results, only link that use 868 or 900 MHz were possible at 1239 m. This is in agreement with the simulation results which predicted that longer distance links were feasible in only the lower frequency bands of 868 MHz and 900 MHz.

3.2 Impact of Encryption of Power Consumption

During the experiments we also made some measurements to assess the impact of the encryption implemented by the waspmote platform on power consumption. We used four different type of transmissions:

1. Unicast without encryption
2. Unicast with encryption
3. Broadcast without encryption
4. Broadcast with encryption

Note that in these experiments, we measured the time and energy consumption from the sleep and OFF modes to the ON mode to evaluate what is the best energy saving mode for a possible synchronization algorithm. In case of a unicast transmission the protocols waits for an ACK signal, while in case of broadcast there is no ACK. However, in broadcast mode each packet is always sent three times. As depicted by Table 6, the results reveal that encryption (AES 128b) does not add any consumption due to the fact that it is performed using specific hardware circuits included in the XBee card and not in the software layer.

4 LDWSN in Developing Countries

Long distance wireless networks are a necessity for developing countries. Large scale deployments of long wireless networks has been revealed mostly for the WiFi technology with the Technology and Infrastructure for Emerging Regions (TIER) project at University of California at Berkeley [1] spearheading the first efforts in collaboration with Intel, by utilizing a modified Wi-Fi setup to create long-distance point-to-point links for several of its projects in the developing world. This initiative was followed by several others in the developing regions such as (1) an unamplified Wi-Fi link of 279 km link achieved by Fundacin Escuela Latinoamericana de Redes (Latin American Networking School) [8] (2) a chain multi-hop WiFi based longest network of the world spanning 445 km in the jungle region of Peru, Loreto, implemented by the Rural Telecommunications Research Group of the Pontificia Universidad Catlica del Per (GTR PUCP) [9] and (3) other networks such as the implemented by the APRL unit of the International Centre for Theoretical Physics in Malawi [10].

While most long distance deployments of WiFi have been focussed on fine-tuning the MAC protocol [[11]-[16]], long distance WSN deployment has been demonstrated in the Waspmote family of sensor networks. Waspmote achieves much longer range compared hundred meters range limitation of many of the existing sensor technologies. Waspmote achieves much longer range compared to hundred meters range limitation of many of the existing sensor technologies. The main differences between Waspmote and these technologies in order to get LDWSN are:

- Higher sensibility.
- Higher txpower.
- Waspmote uses an external connector for the antenna (SMA) allowing the connection of antennas with a higher gain and with the right polarity.

It should also be observed that the frequency plays a capital role in long range deployment. While for the 2.4GHz band, the links can be quite similar, WiFi can not compete with the 868 and 900MHz bands used by some of the Waspmote transceivers.

5 Conclusion

Building upon the Radio Mobile and Waspmote family of WSNs, this paper has assessed the relevance of using simulation in wireless sensor network preplanning and presented a long distance WSN deployment scenario in harsh conditions. The preliminary results presented reveal that simulation may be in agreement with the reality obtained through experimentation when planning long distance links. These results also reveal that by offering a diversity of transceivers running in different frequency bands, the Waspmote family of WSNs present a good platform for the deployment of long distance WSNs. Using a testbed in desert conditions, this paper has presented the preliminary steps towards the implementation of WSNs beyond their traditional ranges. A future step consist of building upon our study to compare different radio propagation models to assess how closely fit with real-life deployment they are. Comparing WiFi long distance efforts with LDWSN using the 802.15.4 modulation and/or protocol is another direction for future research work.

References

1. Technology and Infrastructure for Emerging Regions, TIER (2007), http://tier.cs.berkeley.edu/wiki/
2. Braue, D.: Future Tech: Mote Control. In: Australian Personal Computer, pp. 94–96 (2005)
3. Giorgetti, G., et al.: Exploiting Low-Cost Directional Antennas in 2.4 GHz IEEE 802.15.4 Wireless Sensor Networks. In: 2007 European Conference on Wireless Technologies (2007)
4. Dunlop, J., Cortes, J.: Impact of Directional Antennas in Wireless Sensor Networks. In: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, pp. 1–6 (2007)
5. Willis, S.L., Kikkert, C.J.: Radio Propagation Model for Long-Range Ad Hoc Wireless Sensor Network. In: International Conference on Wireless Networks, Communications and Mobile Computing, Maui, Hawaii (2005)
6. http://www.libelium.com (last accessed 2010)
7. http://radiomobile.pe1mew.nl (last accessed 2010)
8. Flickenger, R., et al.: Very Long Distance Wi-Fi Networks. In: Proceedings of NSDR 2008 The Second ACM SIGCOMM Workshop on Networked Systems for Developing Regions, USA (2008)
9. Flickenger, R., et al.: Wireless Networking in the Developing World. Limehouse Book Sprint Team, 1st edn. (2006)
10. Zennaro, M., et al.: On a long wireless link for rural telemedicine in Malawi. In: Proceedings of the 6th International Conference on Open Access, Lilongwe, Malawi (2008)
11. Sen, S., Raman, B.: Long Distance Wireless Mesh Network Planning: Problem Formulation and Solution. In: The 16th Annual Interntional World Wide Web Conference (WWW 2007), Banff, Canada (2007)
12. Bhagwat, P., Raman, B., Sanghi, D.: Turning 802.11 Inside-Out. In: Second Workshop on Hot Topics in Networks (HotNets-II), Cambridge, MA, USA (2003)
13. Das, A., Zhu, T.: A Reservation-Based TDMA MAC Protocol Using Directional Antennas (RTDMA-DA) For Wireless Mesh Networks. In: IEEE Global Telecommunications Conference, GLOBECOM 2007, Washington, DC, USA (2007)

14. Kumar, P.: Design, Implementation, and Evaluation of new MAC Protocols for Long Distance 802.11 Networks. Masters thesis, Indian Institute of Technology, Kanpur (2006)
15. Patra, R., et al.: WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks. In: USENIX NSDI, Cambridge, MA, USA (2007)
16. Raman, B., Chebrolu, K.: Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks. In: 11th Annual International Conference on Mobile Computing and Networking (MO-BICOM), Cologne, Germany (2005)

A Generalized Framework for Integrated Vehicle Traffic and Wireless Network Simulation

Mahmoud Taghizadeh[1], Francois Dion[2], and Subir Biswas[1]

[1] Electrical & Computer Engineering, Michigan State University
[2] University of Michigan Transportation Research Institute

Abstract. This paper presents a generalized framework for integrating a wireless network simulator and a vehicle traffic simulator for rapid prototyping and evaluation of Dedicated Short Range Communication (DSRC) based vehicular communication protocols and their applications in the context of Intelligent Transportation System (ITS). A novel method of inter-simulator time synchronization has been designed to simulate the interactions between vehicles and wireless networks at resolutions of up to few tens of milliseconds. This fine grain time synchronization enables the simulation of time-critical ITS safety applications requiring less than 100 ms timing resolutions. The architecture-driven approach that has been adopted further leads to a generalized integration that is agnostic of the specific internal syntaxes used by the individual communication network and vehicle simulators that are being integrated. As a feasibility demonstration, the framework is used to integrate Paramics, a vehicular traffic simulator, with ns-2, a communication network simulator. The resulting integrated simulator is then utilized to investigate the performance of wireless message propagation in the context of a freeway car collision avoidance application.

Keywords: DSRC, ITS, Car Collision Avoidance, Vehicular Communication, Simulation.

1 Introduction

Intelligent Transportation Systems (ITS) promote applications of information and communication technologies to improve transportation system safety, traffic efficiency, environmental quality, and economic productivity. With the development of cellular and various short range wireless technologies such as WiFi and Bluetooth, there has been increasing interest in the development of applications relying on vehicle-to-vehicle and vehicle-to-infrastructure wireless communications. To cater to these needs, ASTM and IEEE jointly adopted in 2003 the Dedicated Short Range Communication (DSRC) standard (ASTM E2213-03 [1]) to provide vehicles with reliable wireless communication capabilities within a 1000-meter range at typical highway speeds. This standard provides seven channels at the 5.9 GHz licensed band reserved for ITS applications, with different channels designated for different applications, including one dedicated to vehicle-to-vehicle communications. It is currently envisioned that DSRC technology will be the predominant communication medium for safety applications relying on vehicle-to-vehicle and vehicle-to-infrastructure

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 205–219, 2010.
© Springer-Verlag Berlin Heidelberg 2010

communications, such as cooperative car collision avoidance, lane-changing assistance, and emergency road hazard notification systems.

Before applications can be deployed, they must be evaluated extensively to ensure their functionality and performance in a range of likely situations. However, evaluations in real world settings are not always feasible, particularly when tests require costly, and often temporary and non-scalable, equipment to be deployed. Many real world tests are further constrained by the inability to evaluate applications in situations that may put the safety of drivers at risk or the difficulty to create or replicate specific traffic scenarios. An efficient alternative to this problem is to use simulation to conduct preliminary evaluations before attempting any field deployment. However, simulation is only effective if models adequately replicating the real word can be used. To adequately simulate Vehicular Ad-hoc Networks (VANETs), a simulation model needs to address all the critical details affecting both vehicle movements and network communications. This includes road network elements, driver behavior, and wireless communication features. However, popular communication network simulators such as ns-2 [2], Qualnet [3] and GloMoSim [4] are primarily designed to simulate data packet networks and lack the sophisticated mobility patterns required to simulate realistic vehicle mobility in road networks. Similarly, commercial vehicular traffic simulators such as Paramics [5], VISSIM [6], AIMSUN [7] and CORSIM [8] are designed primarily to replicate car-following, lane-changing, gap-acceptance and other driver behaviors. They are therefore generally not able to adequately represent critical wireless network elements, such as physical layer noise, fading, data packet collision, and routing protocols.

This paper presents a general-purpose simulation framework that has been developed to address the respective shortcomings of communication network and vehicle traffic simulators and allow the development of simulation environments adequately supporting the evaluation of vehicle-based wireless applications. These shortcomings are addressed by providing a simulation environment in which a communication network simulator can be integrated with a vehicular traffic simulator to provide a comprehensive end-to-end modeling of wireless-based vehicle applications. A unique feature of this framework is the mechanism that has been implemented to ensure accurate, fine-grained time synchronization between the simulation models, which is critical for adequately simulating time-critical ITS safety applications. Use of this framework is demonstrated through a test integration of Paramics, a commercial vehicular traffic simulator that is extensively used in the transportation engineering community, with ns-2, a well-known open source communication network simulator commonly used in the wireless network research community. The resulting integrated simulator is then utilized for a series of case studies investigating wireless message propagation performance in the context of a freeway car collision avoidance application.

2 Related Work and Uniqueness of the Proposed Model

Various efforts documented in the literature have explored how improved mobility patterns can be implemented within communication network simulators. Two examples are the MOVE [9] model, which uses input from the SUMO [10] open-source vehicle traffic simulator to approximate real-world node mobility, and the VanetMobiSim model [11-12], which models mobility based on inputs from the open-source

Intelligent Driver Model [13]. Both models only use the mobility patterns generated by the vehicle traffic simulator as input to the communication network simulators, which can be ns-2, Qualnet or GloMoSim. They do not therefore support the simulation of vehicles responding to messages they receive over the wireless network. This limitation thus prevents the simulation of closed loop vehicle control frequently associated with ITS applications.

The MoVES simulator [14] took a different approach by attempting to develop a simulation model featuring both vehicle mobility and communication network functions. Notable properties of this simulator are its ability to distribute computational loads across multiple machines by forming multiple vehicle clusters based on geographical partitioning, and the use of custom-built vehicular simulation modules designed to improve computation efficiency. Developing an entirely new simulation model can however be extremely time-consuming, especially when complex algorithms are required to capture the full complexity of real systems. Custom-built models may also carry validity concerns that may limit their acceptability in professional circles.

Instead of developing entirely new models, many researchers have tried to integrate an existing communication network simulator with an existing vehicular traffic simulator. In one effort [15], the CORSIM vehicular traffic simulator was integrated with the QualNet wireless network simulator using a third-party distributed simulation software package to allow both models to synchronize their operations. A common message format was also defined to facilitate the exchange the vehicle status and position information. Other similar researches include efforts to link QualNet with Paramics [16]. ns-2 with SUMO [17], and VISSIM with ns-2 [18].

The work presented in this paper is in line with the above integration efforts, with two important distinctions. First, a novel inter-simulator time synchronization method has been designed to enable interactions between vehicles and wireless networks at resolutions of up to a few tens of milliseconds. This is in contrast to the coarse-grain synchronization of up to a second that are often used in other research (as in [16] and [15]). Coarse-grain synchronization is not tolerable for simulating time-critical DSRC safety applications as it may introduce too many discrepancies between simulated and real systems. By implementing a fine-grain synchronization, the proposed integration mechanism is thus able to simulate time-critical safety applications requiring no more than 100 ms latency, such as cooperative car collision avoidance [19]. A second unique feature of the proposed framework is its architecture-driven approach. This approach leads to a generalized integration that is agnostic of the specific internal syntaxes used by the individual models used and which allows the integration of any combination of vehicle and network communication simulators.

3 Integration Architecture

This section presents the integration architecture that has been developed.

3.1 Layered Model

The three basic elements of an integrated road network/communication network environment are the road network, the vehicles, and the wireless communication devices. The first two elements can commonly be modeled with relatively high details in existing vehicle traffic simulator, while communication network simulators generally provide the best environment for modeling wireless devices. As shown in Fig. 1, a vehicular simulator is thus used to model the road network and traffic control devices, generate vehicles, and simulate driving behavior. The vehicle simulator is also used to model ITS applications interacting with vehicle controls and to generate the messages to be carried by the communication network. All communication functions, including the determination of which vehicles receive the messages being propagated, are left to the communication network simulator.

The layered approach allows specialized simulation functions that have been developed and extensively validated within each type of simulator to be used to model the corresponding elements in the integrated simulation environment. For example, the 802.11 protocols, which are crucial components for wireless communication in VANETs, have already developed and tested in network simulators such as ns-2 and QualNet. Some commercial vehicle simulators similarly offer complex driver behavior models that have benefited from years of incremental development. The layering further reduces the need to develop complex mechanisms for addressing differences in time granularity between the various models used. For example, whole vehicular traffic simulators are often designed to operate with time steps of 100ms or higher, communication network simulators often operate on a millisecond or microsecond time scale. Since each model retains its original simulation processes in the layered approach, differences in time granularity can be addresses through the implementation of much simply inter-layer synchronization interfaces. As shown later, the layered approach finally allows each simulation model to run on different computers with potentially different operating systems. This offers a potential to reduce computational time, as well as better flexibility and portability.

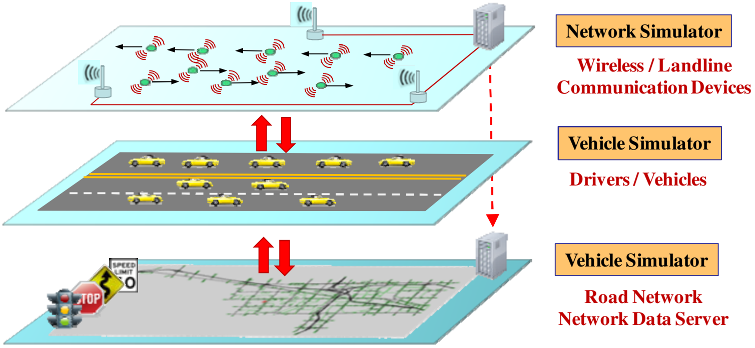


Fig. 1. Layered simulation integration model

3.2 Technical Challenges

The first key challenge in the integration effort is inter-simulator time synchronization. Time synchronization is required to make sure that both the simulation kernels maintain a common notion of time, particularly when different simulators are designed to progress at different paces. For instance, most communication network simulators are *event-driven*. Time progresses from one event to the next, which often results in rates of time progression that do not remain constant from one simulation step to the next. On the other hand, most vehicular traffic simulators are *time-driven*. In this case, time advances at fixed intervals. Because of this difference, time may therefore not advance at the same pace in both types of models. A solution to this problem is to use the system time of one of the simulator as a global reference time. If one of simulator runs faster, a synchronization mechanism would then force it to wait for the other one to reach the same time before allowing it to progress further.

The second challenge is to synchronize the positions of the nodes representing the vehicles in both simulators. To adequately model communication environments, the position of each node representing a vehicle within the communication simulator must match the location of the same vehicle in the vehicle traffic simulator. This problem is solved by synchronizing the position of nodes at the start of each simulation time step. Additional processes must also be considered to handle vehicles entering and leaving the simulation network during simulation run time.

The third challenge is computational efficiency. While each model carries its own computational load, time and node synchronization can significantly increase the computational load associated with a given simulation and result in long execution times. However, while fast simulation speed is desirable, simulation accuracy is deemed to have higher importance, even if it results in slower execution speeds. While efforts have been made to reduce computational loads when possible, the primary focus of the research effort was on functionality. That is why no specific evaluation of computational efficiency is reported in this paper.

3.3 Synchronization Model

Fig 2 depicts the proposed inter-simulator synchronization process. The process starts with both simulators being initialized to a common reference time, The vehicle simulator clock is used here as reference since the time-event approach implemented by these simulators allows for a better control over when each model should stop when running in parallel. Following this initialization, the vehicle simulator allowed to move one time step *T* ahead. To ensure time synchronization, the vehicle simulator then sends a message to the network simulator authorizing it to move forward as well

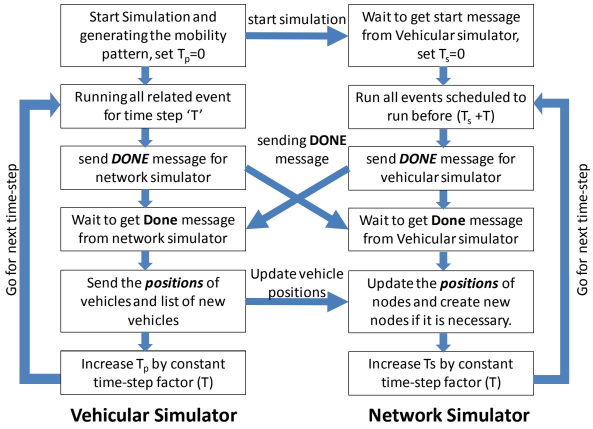

Start Simulation and generating the mobility pattern, set T$_p$=0
start simulation
Wait to get start message from Vehicular simulator, set T$_s$=0
Running all related event for time step 'T'
Run all events scheduled to run before (T$_s$ +T)
sending **DONE** message
send ***DONE*** message for network simulator
send ***DONE*** message for vehicular simulator
Wait to get **Done** message from network simulator
Wait to get **Done** message from Vehicular simulator
Send the ***positions*** of vehicles and list of new vehicles
Update vehicle positions
Update the ***positions*** of nodes and create new nodes if it is necessary.
Increase T$_p$ by constant time-step factor (T)
Increase Ts by constant time-step factor (T)
Go for next time-step
Go for next time-step
Vehicular Simulator
Network Simulator

Fig. 2. Time, node and position synchronization model

by a time *T*. When the vehicle simulator reaches the end of the simulation step, it then sends a DONE message to the network simulator. The network communication simulator does the same when it completes its simulation step. Whichever simulator completes its tasks first then waits for the other one to be done before going any further. When both simulators have completed their step, the node update mechanism described above occurs. This secondary synchronization mechanism starts with the vehicle simulator generating a list detailing the position, speed and heading of all DSRC-equipped vehicles and sending it to the network simulator. Upon reception of this information, the network communication synchronizes location and attributes of all its modeled nodes to that of the vehicle simulator. Whenever information regarding a previously unknown node is received, a new node is then created using the received attributes. A similar mechanism is also used for node deletion. Once this updating process is completed, a new cycle of simulation is then initiated, with both models again allowed to advance *T* seconds.

3.4 Data Communication Model

Fig. 3 shows the functional dependency between the simulators during data communication. A wireless message (e.g., collision warning message) is typically generated by a vehicle within the vehicle simulator, and subsequently passed on to the node within the network simulator that corresponds to the originating vehicle. The message is then routed within the network simulator using a chosen routing/broadcast protocol. Whenever it is determined that a copy of the wireless message has reached a recipient node within the network simulator, the message is then passed back to the corresponding vehicle in the vehicle simulator. This process ensures that vehicle-generated wireless messages are successfully delivered to other vehicles following a realistic routing/broadcast protocol. Upon receiving a message a vehicle would then either discard the received message or react to it (for instance, slow down or change lane). Depending on the application being simulated, the vehicle may also generate a new message intended for other vehicles or simply rebroadcast the received message.

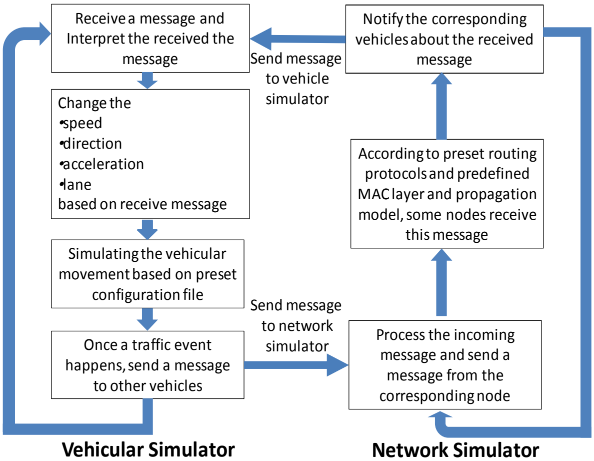

Receive a message and Interpret the received the message
Change the
•speed
•direction
•acceleration
•lane
based on receive message
Simulating the vehicular movement based on preset configuration file
Once a traffic event happens, send a message to other vehicles
Send message to vehicle simulator
Send message to network simulator
Notify the corresponding vehicles about the received message
According to preset routing protocols and predefined MAC layer and propagation model, some nodes receive this message
Process the incoming message and send a message from the corresponding node
Vehicular Simulator
Network Simulator

Fig. 3. Data communication model

4 Implementation

This section presents the details of a specific integration implementation involving Paramics, a commercial vehicular simulator extensively used in the traffic engineering community, and ns-2, a popular open source wireless network simulator. Typical to each type of model, Paramics is *time-driven* and ns-2 is *event-driven*. These simulators further operate on different platforms, with Paramics operating on Windows and ns-2 on Linux.

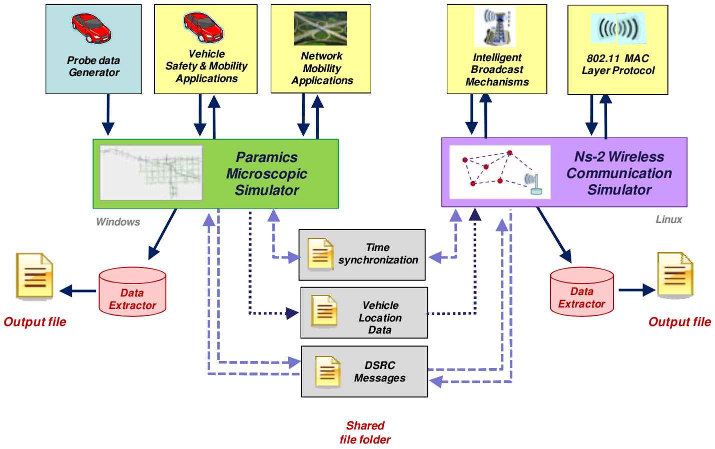

Probe data Generator
Vehicle Safety & Mobility Applications
Network Mobility Applications
Intelligent Broadcast Mechanisms
802.11 MAC Layer Protocol
Paramics Microscopic Simulator
Windows
Ns-2 Wireless Communication Simulator
Linux
Time synchronization
Vehicle Location Data
DSRC Messages
Data Extractor
Output file
Data Extractor
Output file
Shared file folder

Fig. 4. Components of the ns-2-Paramics integrated simulator

Fig. 4 illustrates the functional components of the integration model. As defined in Fig. 1, Paramics is used to model the road network, vehicle generation, vehicle interactions, and ITS application simulation. Similarly, while ns-2 implements the wireless physical, Medium Access Control (MAC), and routing/broadcast mechanisms. Data transfer from one simulator to the next to realize the processes shown in Figs. 2 and 3 is implemented using a shared folder/file mechanism.

4.1 Time Synchronization

Fig. 5 illustrates an example of the mechanism that has been implemented to allow Paramics and ns2 to maintain time synchronization. In the example, the synchronization process starts with both simulators synchronize their clock at execution time 1. As explained in Section III, Paramics internal clock is used as the reference time. When this is completed, a new event, termed the Synchronization event E_t, is introduced within ns-2. This event is scheduled to occur after at time P_t, which corresponds to the duration of one Paramics time step. Paramics then completes the execution of its time step (reaches simulation time P_t) at real time 3 and sends a DONE message to ns-2. In this case, ns-2 only completes its time step (reaches simulation time P_t) at time 4.5. Paramics thus wait for ns-2 to complete its time step before going any further. When ns-2 reaches P_t, it then sends a DONE signal to Paramics and schedules another synchronization event E_t time $2.P_t$. Both simulators then proceed with their computations. In the second step, Paramics reaches $2.P_t$ at time 10, while ns-2 reaches $2.P_t$ at time 6. This causes ns-2 to wait to receive a DONE signal from Paramics, which arrives at time 10, before starting the next time step. After a second synchronization, another synchronization event is schedule for when each simulator reach *3.P_t* and the simulation process repeats.

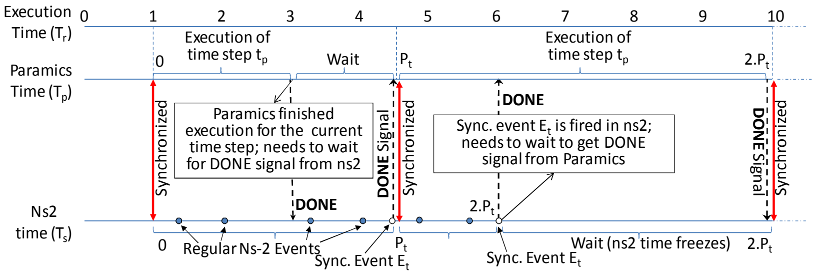


Fig. 5. Time synchronization mechanism in ns-2-Paramics integrated simulator

4.2 Vehicle Position Synchronization

Position synchronization across ns-2 and Paramics is accomplished at each synchronization point (once in each P_t period). At each synchronization point, Paramics sends the following information for each vehicle to ns-2 using the shared folder/file mechanism shown in Fig. 4. Upon receiving the information, ns-2 then updates the position and attributes of the corresponding nodes, to ensure correspondence across both models.

Table 1. Node Synchronization Parameters

Vid	Vehicle ID	SN	Start node of link
X	X Coordinate	EN	End node of link
Y	Y Coordinate	Ln	Lane
S	Speed	B	Bearing
D	Distance from end of the link where vehicle is located		

While the common approach in ns-2 is to create all network nodes at the beginning of a simulation, Paramics normally allows vehicles to be injected and taken out of the system at run time. To address this discrepancy, node handling in ns-2 is modified to support run-time operations. When a vehicle position synchronization message is received by ns-2 with an unknown vehicle id, it creates a new node and maps its attributes to those of the new vehicle from Paramics. When Paramcis removes a vehicle, a message identifying this removal is sent to ns-2 to enable it to remove the corresponding node from its system modeling.

4.3 Data Communication

Fig. 6 describes the vehicle-to-vehicle wireless message delivery process. When a vehicle in Paramics decides to send a message over the wireless network, the message is first stored in a file containing all the messages that are to be sent to ns-2. ns-2 reads the message file at each synchronization point. Upon receiving a message, ns-2 simulates its wireless delivery using a chosen MAC, routing and/or broadcast protocol. When it is determined that a message has reached its destination, it is then stored in an outgoing message file that Paramics will read at the next synchronization event. In the example of Fig. 6, the unicast message is delivered to a single node at simulation time *3.P_t*/2. The earliest opportunity for ns-2 to deliver the message to the corresponding vehicle in Paramics is the next synchronization point at time *2.P_t*. At this point, vehicle will read the message and potentially respond to it.

Fig. 6 implies that data communication latency may arise from the way time synchronization is implemented. In the example, the maximum potential latency corresponds to the duration of a Paramics time step, or P_t. If Paramics is run with a 100 ms time step, which is often the case, the resulting latency would then typically fall within the recommended tolerable delay for the DSRC-enabled safety applications [19]. Latency effects only occur when a message is passed from Paramics to ns-2, and ns-2 to Paramics. Nothing occurs when a message is continuously handled within ns-2, such as during a wireless broadcast or multi-hop routing.

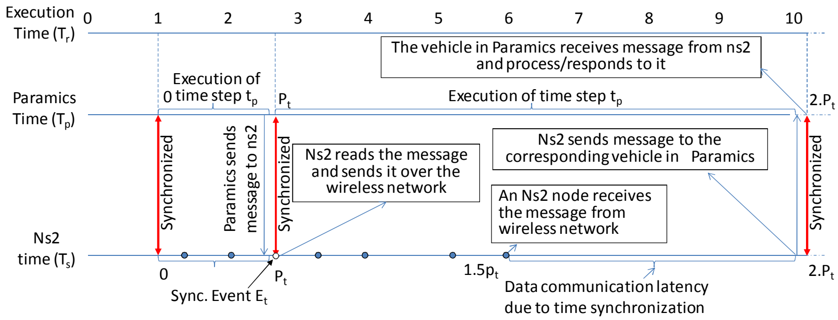

Execution Time (T_r)
0
1
2
3
4
5
6
7
8
9
10
The vehicle in Paramics receives message from ns2 and process/responds to it
Execution of time step t_p
0
P_t
Execution of time step t_p
$2.P_t$
Paramics Time (T_p)
Synchronized
Paramics sends message to ns2
Synchronized
Ns2 reads the message and sends it over the wireless network
Ns2 sends message to the corresponding vehicle in Paramics
An Ns2 node receives the message from wireless network
Synchronized
Ns2 time (T_s)
0
P_t
$1.5p_t$
$2.P_t$
Sync. Event E_t
Data communication latency due to time synchronization

Fig. 6. Message transactions between Paramics and ns-2

5 Application Case Study: Cooperative Car Collision Avoidance

In this section, we analyze a cooperative collision avoidance (CCA) application using the developed ns-2/Paramics integrated simulation system presented in Section 5. On highways, drivers often heavily rely on the activation of tail brake light of the car immediately ahead to determine whether they should brake. This creates potentially dangerous situations when a vehicle is following another one closely, particularly where there is limited ability to see past the vehicle in front. Driver reaction time, which is the duration between when an event is observed and when the driver actually applies the brake, typically ranges from 0.75 seconds to 1.5 seconds [20]. At 70 mph, this means that between 75 and 150 ft is traveled before any reaction occurs. When short inter-vehicle spacing are maintained, there may therefore be little margin of safety to avoid collisions when a sudden braking occurs. In dense traffic, the cumulative reaction times of the successive drivers may further lead to a string of secondary crashes and create multi-car accident chains [19].

Collisions and chain collisions can be potentially avoided, or lessened in severity, by reducing the delay between an emergency event and the moment that approaching vehicles are informed of it. One way to do so is to propagate an incident warning message vehicle-to-vehicle. This could allow bypassing the traditional chain of drivers reacting to the activation of the brake lights of the vehicle immediately ahead, and even allow drivers to respond to an incident before actually seeing it. However, such benefits can only be achieved if the underlying communication system enables each vehicle to receive critical messages in a timely manner.

5.1 Simulation Scenario and Parameters

The case study is implemented by instructing Paramics to simulate an incident suddenly blocking a three-lane freeway. To model this scenario, an existing modeling of a section of I-96 near Novi, Michigan, is used. As shown in Fig. 7, vehicles enter the freeway at the west end of the network and travel up to 7 miles at an average speed of 70 mph before reaching the incident location. This allows typical freeway traffic arrival patterns, such as platooning behind slower moving vehicles, to develop.

The incident is simulated by suddenly adding a stopped vehicle on each lane of the freeway. The appearance of these vehicles then forces other vehicle to react to their presence. Since Paramics does not intrinsically allow collisions to happen, changes were made to the default car-following logic, through the model's Application Programming Interface, to ensure that a realistic response would follow and determine which vehicles would be implicated in a collision. The maximum deceleration rate is first capped at 8 m/s^2. Collisions were then assumed to occur each time Paramics would impose a negative speed to "adjust" the position of a vehicle that has come too close to another one.

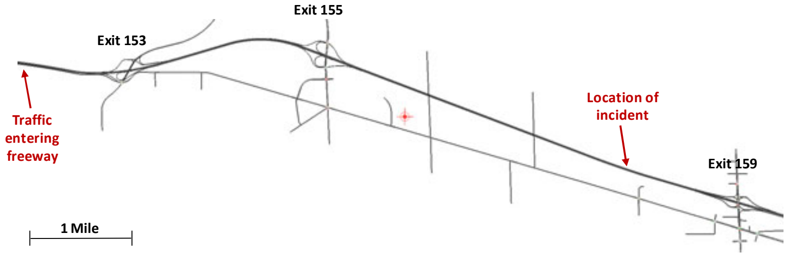


Fig. 7. Simulated freeway (I-96 near Novi, Michigan)

Table 2. Simulation Parameters

Paramics Parameters	ns-2 Parameters
Number of Vehicles: 700 vehicles	MAC Protocol:802.11
Average vehicle Speed: 32 m/s	Radio model: Two ray ground
Recommended Speed: 7,10 and 15 m/s	Packet Error Rate: [0-100]%
Vehicle Length: 4 m	Incident Message Size: 128 bytes
Maximum Emergency Deceleration: 8 m/s^2	Communication Range: 250m
Average Driver Reaction Time: 1 s	Incident Message Period: 100ms
Inter-vehicle Spacing: [0.6 – 1] s	DSRC- enabled Cars:[0,15,25,50,80, 100]%

Table 1 lists the specific simulation parameters that were assigned to Paramics and ns-2 for the modeling of the incident scenario. For simplicity, only passenger cars were simulated. No traffic outside the freeway was also simulated. In the base scenario, drivers are assumed to have an average reaction time of 1 s and to stay on average 1s behind the vehicle ahead. Since Paramics randomly assigns the aggressiveness and awareness of each driver, different drivers have in fact slightly different reaction time, desired speed, and desired vehicle spacing, replicating what typically occurs in real road networks. The overall average of these parameters will however corresponds to the specified average values in the modeling parameters.

5.2 Wireless Message Generation and Propagation

The CCA application is implemented by instructing each vehicle implicated in an incident to broadcast a warning message to the approaching traffic. These messages, generated every 0.1 second, include the location of the vehicle implicated in an incident, and a recommended approach speed. A message may for instance indicate a stopped vehicle on the right lane and instruct approaching vehicles to slow down to 30 mph. Depending on the specific simulation setup, these messages may be propagated only to vehicles within one hop of the vehicles generating it, or up to several kilometers away through multi-hop routing. A one-hop limit is assumed for all the scenarios considered in this study. Upon receiving a warning message, any DSRC equipped vehicle will then gradually slow down to the recommended speed. Non-DSRC vehicles will not respond to the message.

6 Performance Results

The effectiveness of the CCA application is evaluated by compiling the number of cars colliding and the speed of each vehicle before a collision occurs. Any improvement should result in a lower number of cars colliding or, for vehicles unable to avoid a collision, a reduction in the speed at which the collision occurs.

6.1 Number of Cars Colliding

Fig. 8 indicates the effect of the CCA on the number of colliding vehicles in traffic streams featuring a 1.0 s and an aggressive 0.6 s inter-vehicle spacing. The illustrated results are for a single simulation run of each parameter combination considered. In both cases, the scenario without the CCA expectedly produces the largest number of collisions. The number of collisions is also higher for the scenario with the more aggressive 0.6 s inter-vehicle spacing. When the CCA application is activated, the number of collisions significantly drops in both scenarios. In both cases, the recommendation of a different approach speed also results in changes in the number of vehicles colliding.

For the scenario considering a 0.6 s average spacing, the least number of colliding vehicles is obtained with a recommended approach speed of 10 m/s (36 km/h). The scenarios with a 1.0 s average speed, the least number of collisions is obtained for a recommended approach speed of 7 m/s (25 km/h). The fact that the slowest recommended speed does not always result in the least number of collisions is explained by the variability of driver behaviors. While an average spacing of 1.0 or 0.6 is used,

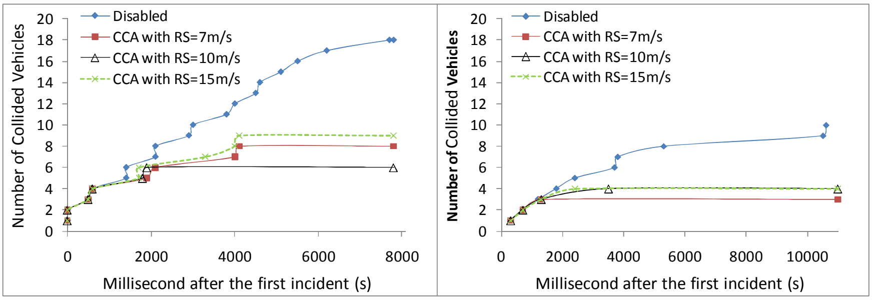


Fig. 8. No. of collided cars with vehicle spacing (a) 0.6s (19.1m) and (b) 1s (32m)

aggressive drivers may travel with shorter spacing while less aggressive drivers will favor a larger spacing. If an aggressive driver follows very closely a vehicle initiating a sudden harsh deceleration, its reaction time may then not allow him to react in time to avoid a collision. Most of the collisions occurring a few seconds after the incident are the result of aggressive drivers following other vehicles too close when a braking occurs. Stochastic variations in vehicle positions and driver aggressiveness effects thus explain why a reduction in the recommended approach speed does not necessarily results in lower incidents.

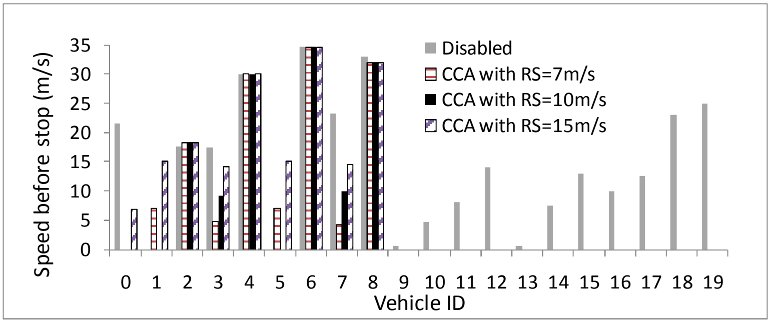


Fig. 9. Speed of colliding cars with and without CCA, 0.6 s vehicle spacing scenario

6.2 Severity of Collisions

Fig. 9 indicates the speeds at which vehicles in scenarios of Fig. 8 considering a 0.6 s average inter-vehicle spacing collided. Without CCA, all vehicles except 1 and 5 collide. The speeds at which collisions occurred varied from close to 0 m/s to full speed (32 m/s). With the CCA, vehicles 9 through 19 are all able to avoid colliding. Some other vehicles are also able to avoid colliding depending on the recommended approach speed. While vehicles 2, 4, 6 and 8 collide in each case, some vehicles are further able to reduce the speed at which they collide, notably vehicles 3 and 7.

Fig. 10 further depicts the speed of colliding cars with an inter-vehicle spacing of 1s. Only vehicles 0, 2, 5 and 9 consistently collide. All other vehicles are able to avoid colliding. For the colliding vehicles, there is either a slight reduction or slight increase in the collision speed. However, changing the recommended approach speed does not significantly impact the collision speeds. The large increase for vehicle 0 is likely due to stochastic changes in the position of the vehicle at the time the incident occurs.

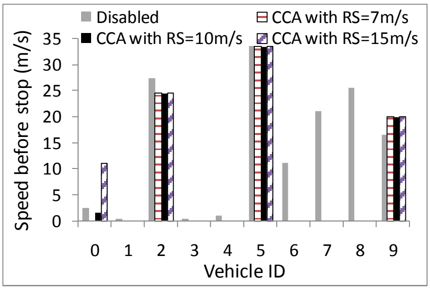


Fig. 10. Speed of colliding cars

6.3 Impact of DSRC Market Penetration Level

While CCA applications are generally expected to provide benefits, it may take several years before all vehicles are equipped with DSRC communication devices. Fig. 11 illustrates the impacts on the number of colliding vehicles of two partial deployments for the scenario of Fig. 8 assuming an average 0.6 s inter-vehicle spacing and incident warning messages recommending a 10 m/s approach speed sent only 1 hop from the generating vehicle. The graph on the left shows the results of 5 simulation runs with different initial number seeds considering an 80% deployment, while the graph on the right considers a 50% deployment.

The simulation results suggest that the performance of the CCA application could be significantly impaired by the presence of non-communicating vehicles within the traffic stream. While the majority of simulations results in a number of collisions that lies between the no-deployment and full deployment scenarios, the stochastic variance of vehicle positioning within the traffic stream is significantly higher with the 50% than the 80% deployment scenario. This variance notably results on one simulation run producing more collisions than the scenario without CCA.

Fig. 12 explains the observed variable. The figure, which shows snapshots from Paramics, illustrates the position of vehicles a few seconds after the incident has occurred in a scenario without CCA and a scenario with a few vehicles with active CCA. In the right frame, the vehicles with active CCA are those with a flag tagged to

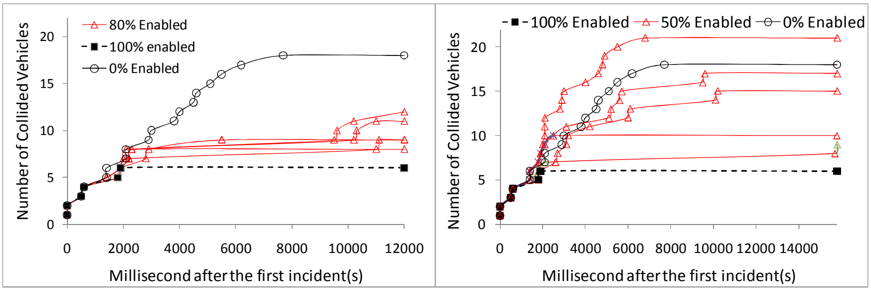


Fig. 11. Impacts of market penetration on vehicle collisions

them. As can be observed, secondary accident chains typically involve a CCA vehicle at their head. In each case, an aggressive driver was following the CCA vehicle very closely. As the CCA vehicles received the warning message and started decelerating, the aggressive driver of the following non-CCA vehicle does not have enough time to respond to the deceleration and thus rear-end the CCA vehicle. This results in the immediate immobilization of both vehicles and triggers a short chain of collisions.

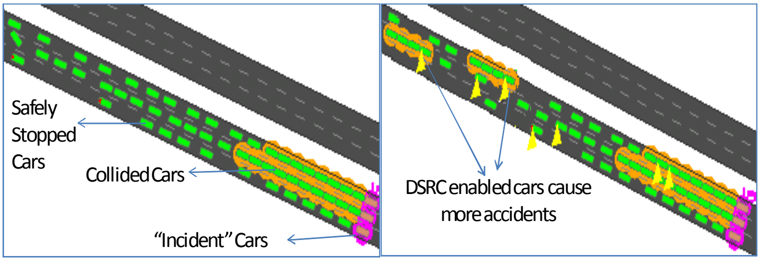


Fig. 12. Snapshot of collided cars; (a) CCA disabled (b) only few cars activate CCA

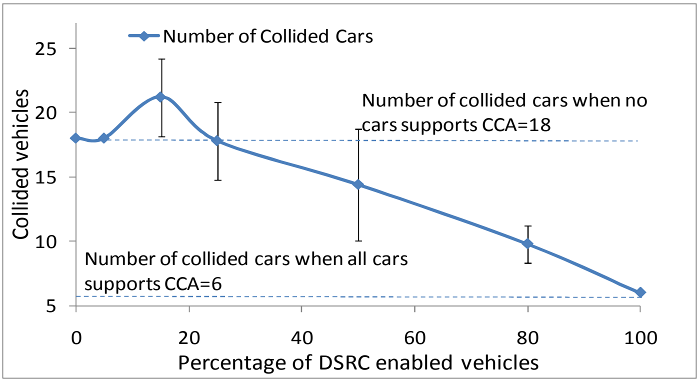


Fig. 13. Number of collided cars with varying market penetration

Figure 13 depicts the impacts on the number of colliding vehicles for wider range of deployment levels. The figure shows both the average number of collision and a 90% confidence range based on the variance of 5 simulation runs. As expected, higher deployment levels generally reduce the number of collisions and the variance of results. For very low deployments (e.g., less than 20%), stochastic effects can create situations with more collisions than a scenario without CCA. The variance observer for the 0%, 5% and 100% deployment cases were negligible.

7 Summary and Conclusions

This paper developed a novel architecture for integrating a wireless network simulator and a vehicle traffic simulator for the purpose of simulating vehicular communication applications. By using a fine-grain inter-simulator time synchronization technique, it has been demonstrated that the proposed framework is able to successfully simulate time-critical ITS safety applications at a 100 ms timing resolution. As a feasibility demonstration, we have integrated the Paramics vehicular traffic simulator with the ns-2 communication network simulator. The resulting integrated simulator was then used to investigate wireless message propagation performance in the context of a freeway car collision avoidance application. Ongoing work on this topic includes a detailed evaluation and optimization of the computational efficiency of the proposed integration technique.

References

1. ASTM E2213-03, Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ASTM International (July 2003)
2. ns-2 Manual, `http://www.isi.edu/nsnam/ns/doc/index.html`
3. QualNet Developer, `http://www.scalable-networks.com`
4. GloMoSim, `http://pcl.cs.ucla.edu/projects/glomosim`
5. Quadstone Paramics Webpage, `http://www.paramics-online.com`
6. PTV America Webpage, `http://www.ptvamerica.com`
7. Aimsun Webpage, `http://www.aimsun.com/site`
8. CORSIM, `http://mctrans.ce.ufl.edu/featured/tsis`
9. Lan, K.C., Chou, C.: Realistic Mobility Models for Vehicular Ad hoc Network (VANET) Simulations. In: Proceedings of 8th International Conference on ITS Telecommunications, pp. 362–366 (October 2008)
10. SUMO: Simulation of Urban MObility, `http://sumo.sourceforge.net`
11. Fiore, M., Harrl, J., Filali, F., Bonnet, C.: Vehicular Mobility Simulation for Vanets. In: Proceedings of the 40th Annual Simulation Symposium (ANSS 2007), pp. 301–309 (2007)
12. VanetMobiSim, `http://vanet.eurecom.fr`
13. Trieber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical Review E 62(2) (August 2000)
14. Bononi, L., Di Felice, M., D'Angelo, G., Bracuto, M., Donatiello, L.: MoVES: A framework for parallel and distributed simulation of wireless vehicular ad hoc networks. Computer Networks 52, 155–179 (2008)
15. Wu, H., Lee, J., Hunter, M., Fujimoto, R.M., Guensler, R.L., Ko, J.: Efficiency of Simulated Vehicle-to-Vehicle Message Propagation in Atlanta, Georgia, I-75 Corridor. Transportation Research Record, No. 1910, 82–89 (2005)
16. Kim, H., Shin, M., Nam, B., Lovell, D.J.: An Integrated Transportation and Communication Simulation Framework for Vehicular Ad Hoc Network Applications. In: 87th Annual Meeting of the Transportation Research Board (January 2008)
17. Traffic and Network Simulation Environment, `http://wiki.epfl.ch/trans/`
18. MSIE, `http://www.cn.uni-duesseldorf.de/projects/MSIE`
19. Xu, Q., Sengupta, R., Jiang, D.: Design and Analysis of Highway Safety Communication protocol in 5.9 GHz Dedicated Short Range Communication Spectrum. Proceedings of IEEE Vehicle Technology Conference 57(4), 2451–2455 (2003)
20. Yang, X., Liu, J., Zhao, F., Vaidya, N.: A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning. In: MobiQuitous 2004 (2004)

Author Index