Xiang-Yang Li
Symeon Papavassiliou
Stefan Ruehrup (Eds.)

LNCS 7363

Ad-hoc, Mobile, and Wireless Networks

11th International Conference, ADHOC-NOW 2012
Belgrade, Serbia, July 2012
Proceedings

Springer

Lecture Notes in Computer Science 7363

Xiang-Yang Li Symeon Papavassiliou
Stefan Ruehrup (Eds.)

Ad-hoc, Mobile, and Wireless Networks

11th International Conference, ADHOC-NOW 2012
Belgrade, Serbia, July 9-11, 2012
Proceedings

Springer

Volume Editors

Xiang-Yang Li
Illinois Institute of Technology
Department of Computer Science
10, West 31st Street
Chicago, IL 60616, USA
E-mail: xli@cs.iit.edu

Symeon Papavassiliou
National Technical University of Athens
School of Electrical and Computer Engineering
Iroon Polytechniou 9
Athens 15780, Greece
E-mail: papavass@mail.ntua.gr

Stefan Ruehrup
FTW - Telecommunications Research Center Vienna
Donau-City-Strasse 1
1220 Vienna, Austria
E-mail: stefan.ruehrup@ftw.at

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-31637-1 e-ISBN 978-3-642-31638-8
DOI 10.1007/978-3-642-31638-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012940929

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

LNCS Sublibrary: SL 5 – Computer Communication Networks
and Telecommunications

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW) has become a well-known venue for research dedicated to wireless sensor networks and mobile computing. Its first event took place in Toronto, Canada, in 2002. ADHOC-NOW was then hosted further times in Canada as well as in France, Mexico, Spain, and Germany. In 2012 it was the first time that ADHOC-NOW took place in Serbia. The 11th ADHOC-NOW was held, during July 9–11, in Belgrade, the capital and largest city of Serbia, located at the confluence of the Sava and the Danube river.

The 11th ADHOC-NOW attracted 76 submissions of which 36 papers were accepted for presentation after rigorous reviews by external reviewers, Technical Program Committee members and discussions among Technical Program Chairs. Most papers received at least three reviews. The accepted papers cover a wide spectrum of traditional networking topics ranging from routing to the application layer, to localization in various networking environments such as wireless sensor and ad-hoc networks, and give insights into a variety of application areas. ADHOC-NOW addresses both experimental and theoretical research and this was reflected in the 2012 program. Overall, the variety of topics made up an interesting and versatile program, which led to a lively exchange of ideas and fruitful discussions.

Many people were involved in the production of these proceedings. First of all, we would like to thank the members of the Technical Program Committee and the external reviewers for their help in providing detailed expert reviews of papers, especially under tight time constraints. We are also grateful to Springer's team for their great assistance during the review and proceedings preparation phase. Last, but not least, we wish to thank all the people of the Organizing Committee who helped in preparing and organizing the event and putting together an excellent program.

The conference proceedings will allow all attendees to obtain detailed information and share this information with other colleagues, for all the papers accepted. ADHOC-NOW 2012 provided a forum for high-quality discussions on the various aspects and application of the emerging field of ad hoc networks all over the world. The large diversity of the highly qualified participants and contributors, who come from a broad range of countries, universities and companies, contributed to its success.

April 2012

Xiang-Yang Li
Symeon Papavassiliou
Stefan Ruehrup

Organization

Program Committee

General Chair

Ivan Stojmenovic	University of Ottawa, Canada

Program Chairs

Xiang Yang Li	Illinois Institute of Technology, Chicago, USA
Symeon Papavassiliou	National Technical University of Athens, Greece

Submission and Proceedings Chair

Stefan Ruehrup	Telecommunications Research Centre Vienna (FTW), Austria

Publicity Chairs

Sandra Sendra	Universidad Politecnica de Valencia, Spain
Hannes Frey	University of Paderborn, Germany
Xu Li	INRIA, Lille, France

Web Chair and Local Arrangements

Milos Stojmenovic	Singidunum University, Belgrade, Serbia

Technical Program Committee

Flavio Assis	UFBA - Federal University of Bahia, Brazil
Michel Barbeau	Carleton University, Canada
Zinaida Benenson	FAU, Germany
Matthias R. Brust	Louisiana Tech University, USA
Juan-Carlos Cano	University Politecnica de Valencia, Spain
Jean Carle	LIFL, France
Chun Tung Chou	University of New South Wales, Australia
Jacek Cichon	Wroclaw University of Technology, Poland
Hongwei Du	Harbin Institute of Technology, Shenzhen Graduate School, China

Rasit Eskicioglu	University of Manitoba, Canada
Rafael Falcon	University of Ottawa, Canada
Jie Gao	Stony Brook University, USA
Yuan He	Tsinghua University, China
Stella Kafetzoglou	National Technical University of Athens, Greece
Vasileios Karyotis	National Technical University of Athens, Greece
Abdelmajid Khelil	TU Darmstadt, Germany
Ralf Klasing	CNRS, France
Jerzy Konorski	Gdansk University of Technology, Poland
Mo Li	Nanyang Technological University (NTU), Singapore
Tianji Li	National University of Ireland Maynooth, Ireland
Xiaoyan Li	Lafayette College, USA
Weifa Liang	Australian National University, Australia
Benyuan Liu	University of Massachusetts Lowell, USA
Hai Liu	Hong Kong Baptist University, Hong Kong SAR
Rongxing Lu	University of Waterloo, Canada
Pietro Manzoni	University Politecnica de Valencia, Spain
Xufei Mao	Tsinghua University, China
Nikola Milosavljevic	Max-Planck-Institut für Informatik, Saarbrücken, Germany
Marc Mosko	Palo Alto Research Center, USA
Marina Papatriantafilou	Chalmers University, Sweden
Dennis Pfisterer	University of Lübeck, Germany
S.S. Ravi	University at Albany – SUNY, United States
Francisco Ros	University of Murcia, Spain
Pedro M. Ruiz	University of Murcia, Spain
Sushmita Ruj	University of Ottawa, Canada
Juan A. Sanchez	University of Murcia, Spain
Violet Syrotiuk	Arizona State University, United States
Shaojie Tang	Illinois Institute of Technology, USA
Eirini Eleni Tsiropoulou	National Technical University of Athens, Greece
Xinbing Wang	ShangHai JiaoTong University, China
Jozef Wozniak	Gdansk University of Technology, Poland
Yulei Wu	Chinese Academy of Sciences, China
Qin Xin	University of the Faroe Islands, Faroe Islands

External Reviewers

Nicolas Bonichon	University of Bordeaux – LaBRI, France
Xiaomin Chen	NUI Maynooth, Ireland
Sebastian Ebers	University of Lübeck, Germany
Juan J. Galvez	University of Murcia, Spain
Florian Huc	EPFL, Switzerland
Aubin Jarry	University of Geneva, Switzerland
Ryszard Katulski	Gdansk University of Technology, Poland
Marek Klonowski	Wroclaw University of Technology, Poland
Florian Massel	University of Lübeck, Germany
Dominik Pajak	INRIA, France
Peter Rothenpieler	University of Lübeck, Germany

Table of Contents

Theory and Localization

Opportunistic Communication, DTN, and Mobility

Sensor Networks

Platforms and Experimentation

Service Discovery, Content Delivery and Control

Routing and Message Dissemination

Applications and Performance Analysis

On Message Complexity of Extrema Propagation Techniques*

Jacek Cichoń, Jakub Lemiesz, and Marcin Zawada

Institute of Mathematics and Computer Science
Wrocław University of Technology
Poland

Abstract. In this paper we discuss the message complexity of some variants of the Extrema Propagation techniques in wireless networks. We show that the average message complexity, counted as the number of messages sent by each given node, is $\mathrm{O}(\log n)$, where n denotes the size of the network.

We indicate the connection between our problem and the well known and deeply studied problem of the number of records in a random permutation. We generalize this problem onto an arbitrary simple and locally finite graphs, prove some basic theorems and find message complexity for some classical graphs such us lines, circles, grids and trees.

1 Introduction

We analyze a synchronous model of communication. At each round each node (1) receives messages from its neighbors; (2) makes some calculations and finally (3) sends, if necessary, some messages to its neighbors.

Our main goal is to investigate the message complexity of algorithms based on Extrema Propagation Techniques discussed and analyzed in [1], [2]. This technique can be treated as a framework for the construction of efficient algorithms in a distributed environment. For example, in [3] this technique was adopted to an algorithm for approximate estimation of a size of the network. The last algorithm was later improved in [4], where the balls and urns model used in [3] was replaced with independent Bernoulli trials in order to obtain a provable precision of proposed algorithm. In this paper we will show that the message complexity of one node for algorithms based on the Extrema Propagation Technique is logarithmic in the network size.

In Section 2 we consider a distributed algorithm which computes minimum from random numbers generated by nodes. In Section 3 we extend our discussion to a distributed algorithm which determines kth order statistics from numbers generated randomly by nodes, which was used in [3] for the estimation of the cardinality of a wireless network. Theorem 1 and Proposition 1 are known. The remaining results are presumably original.

We assume that each node in considered networks can calculate a random real number uniformly in the interval $[0, 1]$ and that this generators are independent.

* Supported by grant N N206 369739 of the Polish National Science Center.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 1–13, 2012.

1.1 Notation and Basic Definitions

We model a network as a simple directed graph $\mathcal{G}$ =(V, E), i.e. V is a nonempty set and $E \subseteq V \times V \setminus \{(v, v) : v \in V\}$. By $d(x, y)$ we denote the length of a shortest directed path from x to y. If there is no such path, then we put $d(x, y) = \infty$. Let $x \in V$ and $r \geq 0$. We put $\mathrm{B}(x, r) = \{y \in V : d(x, y) < r\}$, $\mathrm{D}(x, r) = \{y \in V : d(x, y) \leq r\}$ and $\mathrm{S}(x, r) = \{y : d(x, y) = r\}$. Observe that $\mathrm{D}(x, 0) = \mathrm{S}(x, 0) = \{x\}$. The diameter of a graph $\mathcal{G}$ is the number $\Delta = \sup\{d(x, y) : x, y \in V\}$.

In this paper we shall consider only locally finite graphs, i.e. we shall assume that for all $x \in V$ and $r \geq 0$ we have $|\mathrm{D}(x, r)| < \infty$.

Let us recall that the nth harmonic number is defined by $\mathrm{H}_n = \sum_{k=1}^{n} \frac{1}{k}$ and that $\mathrm{H}_n = \ln(n) + \mathrm{O}(1)$. We will also use the standard extension of the function H_n to the complex plane defined, for example, by the formula $\mathrm{H}_z = \sum_{j \geq 1} z/(j(z + j))$. The Euler Beta function is defined by the formula $\mathrm{B}(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1} dt$ for $\Re(a) > 0$ and $\Re(b) > 0$. We will use the following identity $\mathrm{B}(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$. By $(x)^{\underline{k}}$ we denote the factorial power of x, i.e. $(x)^{\underline{k}} = \prod_{j=0}^{k-1}(x - j)$. By $|A|$ we denote the cardinality of the set A.

2 Propagation of Minimal Number

We start our investigations from the following algorithm (see [1], [2]) of propagation of minimal value of randomly generated real numbers (the pseudo-code of this algorithm is shown at Listing 1):

1. Initially each node $x \in V$ selects independently at random a real ξ_x from the interval [0,1] according to uniform distribution and sends it to all $y \in V$ such that $\{x, y\} \in E$.
2. At each round each node listens to information sent by nodes $S \subseteq \{y : (x, y) \in E\}$ and if $S \neq \emptyset$ and $\xi_x > \min\{\xi_y : y \in S\}$ then
   (a) it puts $\xi_x = \min\{\xi_y : y \in S\}$
   (b) it sends ξ_x to all $y \in V$ such that $\{x, y\} \in E$.

Let $\mathcal{G} = (V, E)$ be the communication graph of considered network, i.e. $\{x, y\} \in E$ if the node x can directly communicate with the node y. Let us assume for the moment that the graph $\mathcal{G}$ is strongly connected. Let Δ denotes the diameter of the graph $\mathcal{G}$. It is easy to see that after Δ rounds for all nodes $x \in V$ we have $\xi_x = \min\{\xi_y : y \in V\}$. Therefore, this algorithm may be used, for example, for leader election in connected networks.

The first goal of our paper is to investigate the message complexity of this algorithm. Let us fix a graph $\mathcal{G} = (V, E)$ and $x \in V$. We say that the node x transmits at the round r if the part (1) or the part ($2b$) of the considered algorithm is executed during the rth round. Let $M_{x,r}$ denote the event "node x transmits at the rth round". Notice that $\Pr[M_{x,0}] = 1$ (each node transmits at initialization step) and that for $r > 0$ we have

$$\Pr[M_{x,r}] = \Pr[\min\{\xi_b : b \in \mathrm{S}(x, r)\} < \min\{\xi_b : b \in \mathrm{B}(x, r)\}] \ .$$

Algorithm 1.

```
Initialization:
$\xi := Random(0,1)$
broadcast $\langle \xi \rangle$ to neighbors
At each round:
gather $\{\eta_i\}_{i \in S}$ from all neighbors
$x = \min\{\eta_i : i \in S\}$
if $x < \xi$ then
    $\xi := x$
    broadcast $\langle \xi \rangle$ to neighbors
end if
```

Theorem 1. *Let $\mathcal{G} = (V, E)$ be a simple directed graph and let $x \in V$. Suppose that $\mathrm{S}(x,r) \neq \emptyset$. Then the events $M_{x,1}, \ldots M_{x,r}$ are independent and*

$$\Pr[M_{x,r}] = \frac{|\mathrm{S}(x,r)|}{|\mathrm{D}(x,r)|} \ .$$

This theorem can be deduced from [5]. We give here a short and self contained proof of it.

Proof. Let $(\xi_v)_{v \in V}$ be a family of independent uniformly distributed random variables in the interval $(0,1)$. Suppose that the theorem is true for a number r and that $\mathrm{S}(x,r+1) \neq \emptyset$. Notice that the event $M_{x,r+1}$ holds if and only if $\min_{v \in \mathrm{S}(x,r+1)} \xi_v < \min_{v \in \mathrm{D}(x,r)} \xi_v$.

Let $a = |\mathrm{S}(x,r+1)|$, $b = |\mathrm{B}(x,r+1)|$, let $X = \min_{v \in \mathrm{S}(x,r+1)} \xi_v$ and $Y = \min_{v \in D(x,r)} \xi_v$. Then $\Pr[X > t] = (1-t)^a$ for $t \in (0,1)$, therefore the function $\phi_X(t) = a(1-t)^{a-1}$ is the density function of the random variable X. Hence

$$\Pr[M_{x,r+1}] = \int_0^1 \Pr[X < Y | X = t] \phi_X(t) dt = \int_0^1 (1-t)^b a (1-t)^{a-1} dt = \frac{a}{a+b} \ ,$$

therefore $\Pr[M_{x,r+1}] = \frac{a}{a+b} = |\mathrm{S}(x,r+1)/|\mathrm{D}(x,r+1)|$.

Let C denote a conjunction of events of a form $(\pm M_{x,1}) \wedge \ldots \wedge (\pm M_{x,r})$, where $+M_{x,i}$ denotes $M_{x,i}$ and $-M_{x,i}$ denotes $\neg M_{x,i}$. Observe that

$$(C \wedge M_{x,r+1} \wedge (X = t)) \leftrightarrow (C \wedge (Y > t) \wedge (X = t))$$

and that $\Pr[C \wedge (Y > t)] = \Pr[C] \cdot (1-t)^b$. Therefore

$$\Pr[C \wedge M_{x,r+1}] = \int_0^1 \Pr[C \wedge M_{x,r+1} | X = t] a (1-t)^{a-1} dt =$$

$$\int_0^1 \Pr[C] (1-t)^b a (1-t)^{a-1} dt = \Pr[C] \frac{a}{a+b} \ ,$$

hence the event $M_{x,r+1}$ is independent from events $\{M_{x,1}, \ldots, M_{x,r}\}$. □

Let $\mathrm{MC}_{x;\mathcal{G}}$ denote the number of times when the part (1) or $(2b)$ of the considered algorithm is executed. Notice that the energy consumption of sending message is much higher than the cost of listening. Hence this number may be treated as the message complexity of the considered algorithm for the node x. Observe that

$$\mathrm{MC}_{x;\mathcal{G}} = \sum_{r\geq 0} \mathbf{1}_{M_{x,r}} \ .$$

Therefore the random variable $\mathrm{MC}_{x;\mathcal{G}}$ can be expressed as a sum $\sum_{r\geq 0}\xi_r$ of independent Bernoulli random variables with mean $\frac{|S(x,r)|}{|D(x,r)|}$. Hence

$$\mathrm{E}\left(\mathrm{MC}_{x;\mathcal{G}}\right) = \sum_{r=0}^{\infty} \frac{|S(x,r)|}{|D(x,r)|} \tag{1}$$

and

$$\mathrm{var}\left(\mathrm{MC}_{x;\mathcal{G}}\right) = \sum_{r=1}^{\infty} \frac{|S(x,r)|}{|D(x,r)|}\left(1 - \frac{|S(x,r)|}{|D(x,r)|}\right) \ . \tag{2}$$

Example. Let us consider the line graph $\mathcal{L}_n$, i.e. let $n > 0$, $V = \{1, 2, \ldots, n\}$ and $E = \{(a, b) \in V \times V : 1 \leq a, b \leq n, |a - b| = 1\}$. Note that $|\mathrm{D}(1, r)| = r + 1$, $|\mathrm{S}(1, r)| = 1$ for $r < n$. Therefore the random variable $L_n = \mathrm{MC}_{1;\mathcal{L}_n}$ has the same distribution as a sum $\sum_{k=1}^{n} X_k$ of independent random variables, where X_k is a Bernoulli random variable such that $\mathrm{E}\left(X_k\right) = \frac{1}{k}$. Hence the random variable L_n has the same distribution as a well studied number of records in random permutation (see [6]). Thus $\mathrm{E}\left(L_n\right) = \mathrm{H}_n$ and (see e.g. [7]) the normalized random variable $(L_n - \mathrm{H}_n)/\sqrt{\mathrm{H}_n}$ converges in distribution to the standard normal distribution.

2.1 Arbitrary Finite Graphs

It is clear that $1 \leq \mathrm{MC}_{x;\mathcal{G}} \leq 1 + \max\{r : \mathrm{S}(x, r) \neq \emptyset\}$. Here we give other bounds:

Theorem 2. *For any finite graph $\mathcal{G} = (V, E)$ and any vertex $x \in V$ we have*

$$2 - \frac{1}{N} \leq \mathrm{E}\left(\mathrm{MC}_{x;\mathcal{G}}\right) \leq \mathrm{H}_N \ ,$$

where $N = |\mathrm{B}(x, \infty)|$.

Notice that $\mathrm{B}(x, \infty)$ is the set of all nodes from which the node x can obtain any message hence it is the connected component of the graph $\mathcal{G}$ to which the node x belongs.

Proof. Let $Z = \mathrm{B}(x, \infty)\setminus\{x\}$. Then $\Pr[\min_{z\in Z}\xi_z < \xi_x] = \frac{N-1}{N}$ and any message ξ_a such that $\xi_a = \min_{z\in Z}\xi_z$ will be eventually transmitted to the node x. This proves the first inequality.

We prove the second inequality using a series of simple transformations of the original graph (V, E). Observe that if $|\mathrm{S}(x,r)| \leq 1$ for all $r \geq 1$ then (V, E) is a line graph with the vertex x at its end. Suppose hence that $|\mathrm{S}(x,r)| \geq 2$ for some $r \geq 1$. Let $a \in \mathrm{S}(x,r)$. We perform the following transformations:

1. remove all edges adjacent to a,
2. remove all edges joining $\mathrm{S}(x,r)$ with $\mathrm{S}(x,r+1)$,
3. add new edges $\{\{a,x\} : x \in \mathrm{S}(x,r)) \cup \mathrm{S}(x,r+1)\}$

Let $\mathcal{H} = (V, E')$ be the resulting graph. Due to inequality $(b \geq 1, c \geq 2)$

$$\frac{c}{b+c} < \frac{c-1}{b+c-1} + \frac{1}{b+c}$$

we have $\mathrm{E}\left(\mathrm{MC}_{x;\mathcal{G}}\right) < \mathrm{E}\left(\mathrm{MC}_{x;\mathcal{H}}\right)$. After a finite number of such transformations we obtain the line graph with n vertices with the vertex x at its end which was discussed at the end of the previous section. □

Let us note that the lower limit from Theorem 2 is reached in the complete graph, and that the upper limit is achieved by the boundary vertex in the line graph.

2.2 Infinite Graphs

Let us consider an arbitrary locally finite graph $\mathcal{G} = (V, E)$ and $x \in V$. Let $L_{x,n} = \sum_{r=0}^{n} X_r$, where X_n are independent Bernoulli random variables such that $\mathrm{E}\left(X_r\right) = p_r$ where $p_r = |\mathrm{S}(x,r)|/|\mathrm{D}(x,r)|$. Then $\mathrm{E}\left(L_{x,n}\right) = 1 + p_1 + \ldots + p_n$. Since $\mathrm{var}\left(X_r\right) < 1$ for each r we may apply the Strong Law of Large Numbers (see e.g. [8], Thm. 22.4) to the sequence X_r and deduce that

$$\Pr[\lim_{n\to\infty} \frac{1}{n}(L_{x,n} - \mathrm{E}\left(L_{x,n}\right)) = 0] = 1 \ .$$

Example. Suppose that $|\mathrm{D}(x,r)| = 2^{r^2}$. Then $|\mathrm{S}(x,r)| = 2^{r^2} - 2^{(r-1)^2}$ for $r > 0$, so $\mathrm{E}\left(L_{x,n}\right) = n + \frac{1}{3} + \frac{2}{3}\frac{1}{4^n}$. Hence

$$\Pr[\lim_{n\to\infty} \frac{L_{x,n}}{n} = 1] = 1 \ .$$

The following result is a reformulation of a result formulated in Exercise 20.12 from [8]:

Proposition 1. *If* $\mathrm{S}(x,r) \neq \emptyset$ *for each* r, *then* $\Pr[\lim_{n\to\infty} L_{x,n} = \infty] = 1$.

Proof. Let (V, E) be a fixed infinite, locally finite graph. Let $(\xi_v)_{v \in V}$ be a family of independent random variables uniformly distributed in $[0, 1]$. Notice that

$$\lim_{n\to\infty} L_{x,n} < \infty \equiv (\exists r)\left(\min_{v\in\mathrm{D}(x,r)} \xi_v < \min_{v\in V\setminus\mathrm{B}(x,r)} \xi_v\right) \ .$$

For each fixed r the event $\min_{v\in\mathrm{D}(x,r)} \xi_v < \min_{v\in V\setminus\mathrm{B}(x,r)} \xi_v$ has probability null, since the set $\mathrm{D}(x,r)$ is finite and the set $V \setminus \mathrm{B}(x,r)$ is infinite. □

The trivial inequality $L_{x,n} \leq 1 + n$ can be improved:

Proposition 2. $\mathrm{E}\left(L_{x,n}\right) \leq 1 + n\left(1 - \sqrt[n]{\frac{1}{|\mathrm{D}(x,n)|}}\right)$

Proof. Let $p_r = |\mathrm{S}(x,r)|/|\mathrm{D}(x,r)|$ and $q_r = 1 - p_r$. Then

$$q_r = |\mathrm{D}(x,r-1)|/|\mathrm{D}(x,r)|$$

for $r > 0$. Therefore $q_1 \cdot \ldots \cdot q_n = 1/|\mathrm{D}(x,r)|$. From the inequality of arithmetic and geometric means we get $q_1 + \ldots + q_n \geq n/\sqrt[n]{|\mathrm{D}(x,n)|}$. Hence, $\mathrm{E}\left(L_{x,n}\right) = 1 + \sum_{r=1}^{n}(1-q_r) \leq 1 + n - n/\sqrt[n]{|\mathrm{D}(x,n)|}$. □

Example. Let us consider an infinite complete binary tree $\mathcal{T} = (V,E)$ and let $x \in V$ be its root. Note that $|\mathrm{D}(x,r)| = 2^{r+1} - 1$. Then, from Proposition 2 by the simple calculation we get $\mathrm{E}\left(L_{x,n}\right) \leq f(n)$, where

$$f(n) = 1 + n\left(1 - \frac{1}{2\sqrt[n]{2}}\right) = \frac{n}{2} + 1 + \frac{\ln 2}{2} + \mathrm{O}\left(\frac{1}{n}\right) .$$

Note that $1 + \frac{\ln 2}{2} \approx 1.3466$. One can verify that this upper bound is sharp. Namely, by some technical manipulation in this case we are able to show that $\mathrm{E}\left(L_{x,n}\right) = \frac{n}{2} + \alpha + \mathrm{O}\left(2^{-n}\right)$, where $\alpha \approx 1.3033$.

There are a lot of examples of infinite graphs where $|\mathrm{S}(x,r)|/|\mathrm{D}(x,r)| = \Theta(\frac{1}{r})$ when r runs to infinity - natural examples of such graphs are the grid-like graphs of arbitrary dimension.

Theorem 3. *Suppose that* $|\mathrm{S}(x,r)|/|\mathrm{D}(x,r)| = \Theta(\frac{1}{r})$. *Then* $\mathrm{E}\left(L_{x,n}\right) = \Theta(\ln n)$ *and the random variable* $(L_{x,n} - \mathrm{E}\left(L_{x,n}\right))/\sqrt{\mathrm{var}\left(L_{x,n}\right)}$ *converges in distribution to the standard normal variable.*

Proof. Let $x_r = |\mathrm{S}(x,r)|/|\mathrm{D}(x,r)|$. From the assumption $x_r = \Theta(\frac{1}{r})$ we deduce that $\sum_{r=0}^{n} x_r = \Theta(\ln n)$ and $\sum_{r=0}^{n} x_r^2 = \mathrm{O}\left(1\right)$. This implies that $\mathrm{E}\left(L_n\right) = \Theta(\ln n)$ and $\mathrm{var}\left(L_n\right) = \sum_r x_r(1-x_r) = \Theta(\ln n)$. Thus the Lindeberg condition (see e.g. [8]) is satisfied, so we may apply Central Limit Theorem. □

2.3 Examples

In this section we shall study message complexity of Algorithm 1 on some classical graphs. Equation 1 gives a possibility to estimate the expected value and variance of the random variable $\mathrm{MC}_{x;\mathcal{G}}$ for many graphs with any required precision. For example, if $\mathcal{C}_n$ denotes the complete graph with n vertices then for any $x \in \mathcal{C}_n$ we have $\mathrm{E}\left(\mathrm{MC}_{x;\mathcal{C}_n}\right) = 2 - \frac{1}{n}$ and $\mathrm{var}\left(\mathrm{MC}_{x;\mathcal{C}_n}\right) = \frac{n-1}{n^2}$.

Line Graph. Let us consider once again the line graph $\mathcal{L}_n$, i.e. let $V = \{1, 2, \ldots, n\}$ and $E = \{(a,b) \in V \times V : 1 \leq a,b \leq n, |a-b| = 1\}$. For an arbitrary number $1 \leq a \leq n/2$ we have

$$\mathrm{E}\left(\mathrm{MC}_{a;\mathcal{L}_n}\right) = 1 + \sum_{k=1}^{a-1}\frac{2}{2k+1} + \sum_{k=2a}^{n}\frac{1}{k} .$$

Hence

$$\mathrm{E}\left(\mathrm{MC}_{\lfloor \frac{n}{2} \rfloor;\mathcal{L}_n}\right) = \mathrm{H}_n - \ln\frac{e}{2} + \mathrm{O}\left(\frac{1}{n}\right) \approx \mathrm{H}_n - 0.306853 + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

At Fig. 1 we plot the diagram of the function $f(a) = \mathrm{E}\left(\mathrm{MC}_{a;\mathcal{L}_{100}}\right)$ for $a = 1, \ldots, 100$. We observe that the maximum value of f is achieved at ends of the graph $\mathcal{L}_n$.

Circle. Let $\mathcal{C}_n$ denote the circle graph with n vertices. If $n = 2k+1$ then for each $x \in \mathcal{C}_n$ we have

$$\mathrm{E}\left(\mathrm{MC}_{x;\mathcal{C}_n}\right) = 1 + \sum_{a=1}^{k} \frac{2}{2a+1} = \mathrm{H}_{k+\frac{1}{2}} + \log\frac{4}{e} = \mathrm{H}_n - \ln\frac{e}{2} + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

For $n = 2k$ we obtain a similar formula

$$\mathrm{E}\left(\mathrm{MC}_{x;\mathcal{C}_n}\right) = \mathrm{H}_{k-\frac{1}{2}} + \log\frac{4}{e} + \frac{1}{n} = \mathrm{H}_n - \ln\frac{e}{2} + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

Grid. Let $\mathcal{G}_n$ denote the grid graph with vertices $V = \{1, \ldots, n\} \times \{1, \ldots, n\}$ and edges $E = \{\{(x,y),(x',y')\} : |x-x'|+|y-y'| = 1\}$. Theorem 2 implies that $\mathrm{E}\left(\mathrm{MC}_{(a,b);\mathcal{G}_n}\right) \leq \mathrm{H}_{n^2}$ for each vertex $(a,b) \in V$.

Proposition 3. *Let $n = 2k-1$ and $N = n^2$. Then*

(a) $\mathrm{E}\left(\mathrm{MC}_{(1,1);\mathcal{G}_n}\right) = \mathrm{H}_N - \delta_1 + \mathrm{O}\left(\frac{1}{\sqrt{N}}\right)$ *where* $\delta_1 \approx 0.729637$,

(b) $\mathrm{E}\left(\mathrm{MC}_{(k,k);\mathcal{G}_n}\right) = \mathrm{H}_N - \delta_2 + \mathrm{O}\left(\frac{1}{\sqrt{N}}\right)$ *where* $\delta_2 \approx 1.415467$.

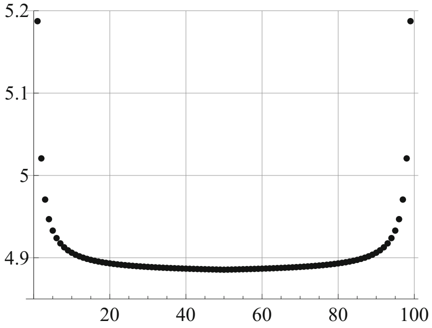


Fig. 1. Plot of $\mathrm{E}\left(\mathrm{MC}_{a;\mathcal{L}_{100}}\right)$ for a= 1,. . . ,100

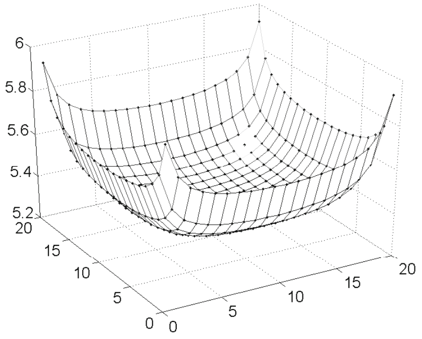


Fig. 2. Plot of $\mathrm{E}\left(\mathrm{MC}_{(a,b);\mathcal{G}_{20}}\right)$ for $a, b \in \{1, \ldots, 20\}$

Proof. (a) Let us consider the vertex $v = (1,1)$ and let us define $\mathcal{S}_{r_1}^{r_2}(x) = \sum_{r=r_1}^{r_2} \frac{|S(x,r)|}{|D(x,r)|}$. Then $\mathrm{E}\left(\mathrm{MC}_{v;\mathcal{G}_n}\right) = \mathcal{S}_0^{n-1}(v) + \mathcal{S}_n^{2n-2}(v)$ where (see Fig. 3)

$$\mathcal{S}_0^{n-1}(v) = \sum_{r=0}^{n-1} \frac{r+1}{\frac{1}{2}(r+1)(r+2)} = 2(\mathrm{H}_{n+1} - 1)$$

and

$$\mathcal{S}_n^{2n-2}(v) = \sum_{r=1}^{n-1} \frac{n-r}{\frac{1}{2}n(n+1) + r\left(n - \frac{r+1}{2}\right)} = \ln 2 + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

Hence,

$$\mathrm{E}\left(\mathrm{MC}_{v;\mathcal{G}_n}\right) = 2\mathrm{H}_{n+1} - 2 + \ln 2 + \mathrm{O}\left(\frac{1}{n}\right) = \mathrm{H}_{n+1} + (\gamma - 2 + \ln 2) + \mathrm{O}\left(\frac{1}{\sqrt{N}}\right) \ .$$

(b) Let us now consider the vertex $v = (k,k)$. In a similar way we split the required sum into two parts $\mathrm{E}\left(\mathrm{MC}_{v;\mathcal{G}_n}\right) = 1 + \mathcal{S}_1^{k-1}(v) + \mathcal{S}_k^{n-1}(v)$ and check that

$$\mathcal{S}_1^{k-1}(v) = \sum_{r=1}^{k-1} \frac{4r}{1 + 2r(r+1)} = \mathrm{H}_N + c + \mathrm{O}\left(\frac{1}{\sqrt{N}}\right) \ ,$$

where $c = -3.108614341\ldots$ and

$$\mathcal{S}_k^{n-1}(v) = \sum_{r=1}^{k-1} \frac{4(k-r)}{(1 - 2k + 2k^2) + (-2 + 4k)r - 2r^2} = \ln 2 + \mathrm{O}\left(\frac{1}{\sqrt{N}}\right) \ .$$

Finally we have $1 - 3.108614341 + \ln 2 = -1.415467160\ldots$.

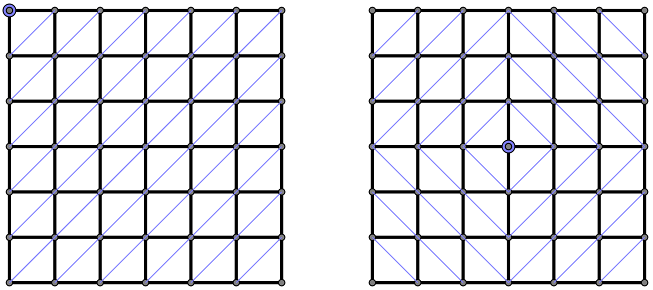


Fig. 3. Division of the graph $\mathcal{G}_7$ into layers depending on the distance from the vertex $(1,1)$ and from the vertex $(4,4)$

At Fig. 2 we plot the diagram of the function $f(a,b) = \mathrm{E}\left(\mathrm{MC}_{(a,b);\mathcal{G}_{20}}\right)$ for all $a, b = 1, \ldots, 20$. We may observe that the maximum value of f is achieved at "corners" of the graph $\mathcal{G}_{20}$.

Algorithm 2.

```
Initialization:
$X := \underbrace{(1,\ldots,1)}_{k}$
$X[1] := Random(0,1)$;
broadcast $X$ to neighbors
At each round:
$Y := X$;
for every obtained array $Z$ from neighbors do
    append $Z$ to $X$;
    sort $X$;
    $X := X[1 \ldots k]$;
end for
if $X \neq Y$ then
    broadcast $X$ to neighbors;
end if
```

3 Propagation of Order Statistics

In [3] and [4] a protocol for wireless networks which propagate kth order statistics of real numbers randomly generated by nodes was used for estimation of the size of a network. Here is the description of the transmission part of this algorithm:

1. Initially each node $v \in V$ sets $X_v[1..k] = (1, 1, \ldots, 1)$, selects a random number $\xi_v \in (0, 1)$, puts $X_v[1] = \xi_v$ and sends X_v to all its neighbors.
2. At the beginning of each round the node v makes a copy $Y = X_v$; next with each obtained array Z from a neighbor the node v makes the following operation: $X_v = \mathrm{sort}(X_v \oplus Z)[1, \ldots, k]$ (where $\oplus$ denotes the concatenation of arrays); finally, at the end of the round, if $X_v \neq Y$ then node v sends the array X_v to all its neighbors.

The pseudo-code of this algorithm is shown at Listing 2. Let us note that the case $k = 1$ was considered in previous section.

Lemma 1. *Let $A, B \subseteq V$, $|A| = a$, $|B| = b$, $A \cap B = \emptyset$. Suppose that $1 \leq k \leq a$. Let $(\xi_v)_{v \in A \cup B}$ be a family of independent random variables uniformly distributed in $(0, 1)$. Let $\xi_{1:a} \leq \ldots \leq \xi_{a:a}$ be the order statistics generated by $(\xi_v)_{v \in A}$. Then*

$$\Pr[\min_{v \in B} \xi_v < \xi_{k:a}] = 1 - \frac{\binom{a}{k}}{\binom{a+b}{k}} .$$

Proof. Let us recall (see e.g. [9]) that the density of the kth order statistic derived from a sequence $(\xi_1, \ldots, \xi_a)$ of independent random variables uniformly distributed in $(0,1)$ is given by the formula $f_{k:a}(t) = \mathrm{B}(k, a-k+1)^{-1} t^{k-1}(1-t)^{a-k}$. Let $\eta = \min_{v \in B} \xi_v$. Notice that $\Pr[\eta < t] = 1-(1-t)^b$ and that η is independent from $\xi_{k:a}$. Therefore

$$\Pr[\eta < \xi_{k:a}] = \int_0^1 (1-(1-t)^b) \frac{1}{\mathrm{B}(k, a-k+1)} t^{k-1}(1-t)^{a-k} dt =$$

$$1 - \frac{1}{\mathrm{B}(k, a-k+1)} \int_0^1 t^{k-1}(1-t)^{a+b-k} dt = 1 - \frac{\mathrm{B}(k, a+b-k+1)}{\mathrm{B}(k, a-k+1)} \ .$$

□

Let $\mathrm{MC}_x^{(k)}$ denote the number of rounds in which the node $x \in V$ sends a message. Observe that if $|\mathrm{D}(x,\infty)| \leq k$ then

$$\mathrm{MC}_x^{(k)} = 1 + \max\{r : \mathrm{S}(x,r) \neq \emptyset\} \ .$$

Theorem 4. *Suppose that $|\mathrm{D}(x,\infty)| > k$. Let $s = \min\{r : |\mathrm{D}(x,r)| \geq k\}$ and $m = \max\{r : \mathrm{S}(x,r) \neq \emptyset\}$. Then*

$$\mathrm{MC}_x^{(k)} = \sum_{r=0}^{m} \xi_r \ ,$$

where $(\xi_r)_{r=0,\ldots,m}$ is a sequence of independent Bernoulli trials such that $\mathrm{E}(\xi_r) = 1$ for $r \leq s$ and

$$\mathrm{E}(\xi_r) = 1 - \frac{\binom{|\mathrm{D}(x,r-1)|}{k}}{\binom{|\mathrm{D}(x,r)|}{k}}$$

for $r > s$.

Proof. Notice that while $|\mathrm{D}(x,r)| \leq k$ then from each sphere $\mathrm{S}(x,j)$ where $j \leq r$ some new information about the kth statistic will be obtained with probability 1. If $r > s$ then the node x has gathered at least k different values from nodes from ball $\mathrm{B}(x,r)$. Hence, its register changes its contents if $\min_{v \in \mathrm{S}(x,r)} \xi_v < X_v[k]$. So we may apply Lemma 1 and deduce that this happens with probability

$$1 - \frac{\binom{|\mathrm{B}(x,r)|}{k}}{\binom{|\mathrm{B}(x,r)|+|\mathrm{S}(x,r)|}{k}} = 1 - \frac{\binom{|\mathrm{D}(x,r-1)|}{k}}{\binom{|\mathrm{D}(x,r)|}{k}} \ .$$

The proof of independence of constructed random variables follows the same lines as in the proof of Theorem 1 of the corresponding fact. □

Example. Let us again consider the vertex $x = 1$ of the line graph $\mathcal{L}_n = \{1, \ldots, n\}$. Recall that $|\mathrm{D}(1,r)| = r+1$. Let us suppose that $k < n$. Therefore we have $\min\{r : |\mathrm{D}(1,r)| \geq k\} = k-1$ and from Theorem 4 we get

$$\mathrm{E}\left(\mathrm{MC}_1^{(k)}\right) = k + \sum_{r=k}^{n-1} \left(1 - \frac{\binom{r}{k}}{\binom{r+1}{k}}\right) = k + \sum_{r=k}^{n-1} \frac{k}{r+1} = k\left(\mathrm{H}_n - \mathrm{H}_k + 1\right) \ .$$

The next result is a generalization of Theorem 2 onto the case of kth order statistics.

Theorem 5. *Let $k \geq 2$. Suppose that $N = |\mathrm{D}(x,\infty)| > k$. Then*

$$2 \leq \mathrm{E}\left(\mathrm{MC}_x^{(k)}\right) \leq k\left(\mathrm{H}_N - \mathrm{H}_k + 1\right) \ .$$

Proof. The proof follows the same lines as the proof of Theorem 2: we transform original graph as long as we get a line graph and use the inequality

$$1 - \frac{\binom{a}{k}}{\binom{a+b}{k}} < 1 - \frac{\binom{a}{k}}{\binom{a+b-1}{k}} + 1 - \frac{\binom{a+b-1}{k}}{\binom{a+b}{k}}$$

which holds for $k \geq 2$. □

Let us now consider an infinite graph. Suppose that s and the sequence $(\xi_r)_{r=0,1,\ldots}$ are defined similarly as in Theorem 4 and let us denote $L^k_{x,n} = \sum_{r=0}^{n} \xi_r$. Theorems 4 and 6 presented below can be proved in the analogous way as the corresponding theorems in Section 2.2.

Proposition 4. $\mathrm{E}\left(L^k_{x,n}\right) \leq 1 + s + (n-s)\left(\ 1 - \sqrt[n-s]{\frac{|\mathrm{D}(x,s)|^{\underline{k}}}{|\mathrm{D}(x,n)|^{\underline{k}}}}\ \right)$

Proof. Note that

$$\mathrm{E}\left(L^k_{x,n}\right) = 1 + s + (n-s) - \sum_{r=s+1}^{n} (1 - \xi_r) \ .$$

From the inequality of arithmetic and geometric means we get

$$\sum_{r=s+1}^{n} (1 - \xi_r) \geq (n-s) \sqrt[n-s]{\frac{|\mathrm{D}(x,s)|^{\underline{k}}}{|\mathrm{D}(x,n)|^{\underline{k}}}} \ .$$

Theorem 6. *Suppose that $|\mathrm{S}(x,r)|/|\mathrm{D}(x,r)| = \Theta(\frac{1}{r})$. Then $\mathrm{E}\left(L^k_{x,n}\right) = \Theta(\ln n)$ and the random variable $(L^k_{x,n} - \mathrm{E}\left(L^k_{x,n}\right))/\sqrt{\mathrm{var}\left(L^k_{x,n}\right)}$ converges in distribution to the standard normal variable.*

Proof. Note that from the fact that $|\mathrm{S}(x,r)|/|\mathrm{D}(x,r)| = \Theta(1/r)$ we can easily deduce that $1 - \frac{|\mathrm{D}(x,r-1)|^k}{|\mathrm{D}(x,r)|^k} = \Theta(1/r)$. One can also check that above relation holds if we replace power k by the falling factorial $\underline{k}$.

3.1 Examples

Circle Let us consider the circle graph $\mathcal{C}_N$ where $N = 2n+1$. Let x be any vertex from this graph. Then $|\mathrm{D}(x,r)| = 2r+1$ for $r \leq n$, so

$$\mathrm{E}\left(\mathrm{MC}_x^{(k)}\right) = m + \sum_{r=m+1}^{n} \left(1 - \frac{\binom{2r-1}{k}}{\binom{2r+1}{k}}\right) \ ,$$

where $m = \lceil \frac{k-1}{2} \rceil$. After some simplification we get

$$\mathrm{E}\left(\mathrm{MC}_x^{(k)}\right) = m + \sum_{r=m+1}^{n} \left(\frac{k}{2r} + \frac{k}{2r+1} - \frac{k^2}{2r(2r+1)} \right) =$$

$$m + k(\mathrm{H}_N - \mathrm{H}_{2m+1}) - k^2 \sum_{r=m+1}^{n} \frac{1}{2r(2r+1)} =$$

$$m + k(\mathrm{H}_N - \mathrm{H}_{2m+1}) + k^2(\mathrm{H}_m - \mathrm{H}_{m+\frac{1}{2}}) + \mathrm{O}\left(\frac{1}{N}\right) \ .$$

Therefore, $\mathrm{E}\left(\mathrm{MC}_x^{(k)}\right) \approx \frac{k}{2} + k(\mathrm{H}_N - \mathrm{H}_k) - k$, so $\mathrm{E}\left(\mathrm{MC}_x^{(k)}\right) \approx k(\mathrm{H}_N - \mathrm{H}_k - \frac{1}{2})$.

Grid. Let us consider the set $V_n = \{(x,y) \in \mathbf{N} \times \mathbf{N} : |x| + |y| \leq n\}$, $E_n = \{((x,y),(x'y')) : |x-x'| + |y-y'| = 1\}$ and the vertex $v = (0,0)$. Let $\mathcal{G}_n = (V_n, E_n)$. Then, for all $r \leq n$ we have $|\mathrm{D}(v,r)| = 1 + 2r + 2r^2$. Applying Theorem 4 to the graph $\mathcal{G}_n$ we get

$$\mathrm{E}\left(\mathrm{MC}_v^{(2)}\right) = 2 + \sum_{r=2}^{n} \left(1 - \frac{(1-2r+2r^2)^{\underline{2}}}{(1+2r+2r^2)^{\underline{2}}} \right) = 2 + \sum_{r=2}^{n} \frac{2+8r^2}{1+3r+4r^2+2r^3} \ .$$

After some transformations we obtain

$$\mathrm{E}\left(\mathrm{MC}_v^{(2)}\right) = 4 \cdot \mathrm{H}_n - 5.62667\ldots + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

Notice that the average message complexity in this case very close to the upper bound

$$2(\mathrm{H}_{2+2n+2n^2} - \mathrm{H}_2 + 1) = 4 \cdot \mathrm{H}_n - 0.768137\ldots + \mathrm{O}\left(\frac{1}{n}\right) \ .$$

given by Theorem 5. In a similar way we can show that for an arbitrary k

$$\mathrm{E}\left(\mathrm{MC}_v^{(k)}\right) = (2k) \cdot \mathrm{H}_n + \mathrm{O}\left(1\right) \ ,$$

in the graph $\mathcal{G}_n$ when $n \to \infty$.

Tree

Let $\mathcal{T}_n$ be a complete binary tree of depth n rooted at node v. Let us recall that $|\mathrm{D}(v,r)| = 2^{r+1} - 1$ and let us set $s = \min\{r : 2^{r+1} - 1 \geq k\}$. Then, from Theorem 4 we have

$$\mathrm{E}\left(\mathrm{MC}_v^{(k)}\right) = 1 + s + \sum_{r=s+1}^{n} \left(1 - \frac{(2^r-1)^{\underline{k}}}{(2^{r+1}-1)^{\underline{k}}} \right) = 1 + n - \sum_{r=s+1}^{n} \left(2^{-k} + \mathrm{O}\left(2^{-r}\right) \right) \ .$$

Hence, we obtain

$$\mathrm{E}\left(\mathrm{MC}_v^{(k)}\right) = (1 - 2^{-k})n + \mathrm{O}\left(1\right) = \alpha_k \mathrm{H}_{(2^{n+1}-1)} + \mathrm{O}\left(1\right) \ ,$$

where $\alpha_k \leq 1.4427$. Observe that in the upper bound given by Theorem 5 the corresponding constant is equal to k.

4 Summary

We analyzed a message complexity of two algorithms based on the Extrema Propagation Techniques - a simple algorithm and an algorithm gathering kth order statistics. We showed that the average message complexity for each node in both algorithms is of order $\mathrm{O}(\log n)$, where n denotes the size of the network.

Note that while considering the records of i.i.d. continuous random variables only the relative order of their outcomes matters (see e.g. [10]). Hence, it is straightforward to observe that the presented results hold for any random variables with a common continuous distribution function. Thus, they can be widely applied. For instance, in [2] Shah et al. consider a general framework for a distributed computing of separable functions, which is based on finding the minimum of exponential random variables.

References

1. Baquero, C., Almeida, P.S., Menezes, R.: Fast estimation of aggregates in unstructured networks. In: Proceedings of the 2009 Fifth International Conference on Autonomic and Autonomous Systems, pp. 88–93 (2009), http://gsd.di.uminho.pt/members/cbm/ps/IEEEfastFinalICAS2009.pdf
2. Mosk-Aoyama, D., Shah, D.: Computing separable functions via gossip. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, pp. 113–122 (2006)
3. Cichoń, J., Lemiesz, J., Zawada, M.: On Cardinality Estimation Protocols for Wireless Sensor Networks. In: Frey, H., Li, X., Ruehrup, S. (eds.) ADHOC-NOW 2011. LNCS, vol. 6811, pp. 322–331. Springer, Heidelberg (2011)
4. Cichoń, J., Lemiesz, J., Szpankowski, W., Zawada, M.: Two-phase cardinality estimation protocols for sensor networks with provable precision. In: IEEE WCNC 2012 Conference Proceeding, IEEE Xplore (2012)
5. Yang, M.C.K.: On the distribution of the inter-record times in an increasing population. J. Appl. Probab., 148–154 (1975)
6. Rényi, A.: Théorie des 'el'ements saillants d'une suite d'observations. Ann. Fac. Sci. Univ. Clermont-Ferrand, 7–13 (1962)
7. Steele, J.M.: The bohnenblust—spitzer algorithm and its applications. J. Comput. Appl. Math. 142, 235–249 (2002), http://portal.acm.org/citation.cfm?id=586795.586814
8. Billingsley, P.: Probability and Measure, 3rd edn. Wiley-Interscience (1995)
9. Arnold, B., Balakrishnan, N., Nagaraja, H.: A First Course in Order Statistics. John Wiley & Sons, New York (1992)
10. Devroye, L.: Applications of the theory of records in the study of random trees. Acta Informatica 26, 123–130 (1988)

Improved Approximation Bounds for Maximum Lifetime Problems in Wireless Ad-Hoc Network

Sang Hyuk Lee and Tomasz Radzik

Department of Informatics,
King's College London,
London, WC2R 2LS, UK
{sang_hyuk.lee,tomasz.radzik}@kcl.ac.uk

Abstract. A wireless ad-hoc network consists of a number of wireless devices (nodes), that communicate with each other within the network using their built-in radio transceivers. The nodes are in general battery-powered, thus their lifetime is limited. Therefore, algorithms for maximizing the network lifetime are of great interest. In this paper we consider the *Rooted Maximum Network Lifetime* (RMNL) problems: given a network N and a node r, the objective is to find a maximum-size collection of routing trees rooted at the node r for a specified communication pattern. The number of such trees represents the total number of communication rounds executed before the first node in the network dies due to battery depletion. We consider two communication patterns, broadcast and convergecast.

We follow the approach used by Nutov and Segal in [15], who developed polynomial time approximation algorithms with constant approximation ratios for the broadcast and convergecast RMNL problems. Our analysis of their algorithms leads to better approximation ratios than the ratios derived in [15]. In particular, we show a 1/7 approximation ratio for the multiple topology convergecast RMNL problem, improving the previous ratio of 1/31.

Keywords: Network Lifetime, Broadcast, Convergecast, Approximation algorithm, Wireless ad-hoc network.

1 Introduction

In wireless ad-hoc networks, *broadcast* and *convergecast* are the fundamental communication patterns. The goal of broadcast is to transmit a message from a source node to all other nodes in the network, whereas convergecast, the goal is for a destination node to collect messages from all other nodes in the network. Many network operations (services) such as information dissemination and data collection are based on these two communication patterns. Since the nodes of wireless ad-hoc networks are normally battery powered, without an easy way of recharging or replacing batteries, an important design objective for communication algorithm is to maximize the energy efficiency.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 14–27, 2012.

Tree-based broadcast (convergecast) schemes are often employed to improve the energy efficiency by reducing duplicate transmissions, which in turn, may also reduce the interference and collisions. In this manner, for broadcast, a directed spanning tree rooted at the source node, in which all edges are directed away from the source node is constructed. Similarly, for convergecast, a directed spanning tree rooted at the destination node with edges directed towards the destination node is constructed. Such trees are also referred to as *broadcast trees* and *convergecast trees*, respectively. During the actual broadcast (convergecast) round, a message is propagated from the source (leaves) along the pre-computed directed tree.

In this paper, we address the *Rooted Maximum Network Lifetime* (RMNL) problems [4, 15] for these two specific communication patterns, broadcast and convergecast. We consider the case of *unidirectional model*, in which a transmission can only be received by at most one node. The input of a RMNL problem is a network N, a node r, information about the initial battery capacity of the nodes and about the energy cost of individual node-to-node transmissions. We measure the *network lifetime* as the number of broadcast (convergecast) rounds until the first node in the network dies due to battery depletion. Thus, we achieve our goal of maximizing network lifetime by providing a maximum-size collection of broadcast (convergecast) trees, where each tree represents one broadcast (convergecast) round.

The RMNL problems can be divided into two categories: *discrete* and *fractional* [4]. In the fractional variant, a data packet is allowed to be divided into smaller packets, which can be transmitted separately. Whereas, in the discrete variant [17], each packet has to be sent in one transmission. The discrete model seems to reflect better the existing network protocols. The RMNL problems can be further divided into two variants: *single topology* and *multiple topology* [16]. In the single topology variant, the same broadcast (convergecast) tree is used for all broadcast (convergecast) rounds. On the other hand, in the multiple topology variant, the trees used in different rounds, do not have to be identical. Additionally, for convergecast, we assume full aggregation of packets.

In this paper, we address the discrete variant of the RMNL problems for broadcast and convergecast, and we consider both single and multiple topology. That is, we consider the following four problems:

- Single Topology Maximum Network Lifetime Broadcast (STB),
- Single Topology Maximum Network Lifetime Convergecast (STC),
- Multiple Topology Maximum Network Lifetime Broadcast (MTB),
- Multiple Topology Maximum Network Lifetime Convergecast (MTC).

1.1 Previous Work

A significant amount of research has been carried out in the area of maximizing the network lifetime under various communication algorithms. In particular, broadcast and convergecast (data gathering) have received a lot of attention. Those studies considered mainly the *omnidirectional* communication model

[9–11, 16, 17, 20], where every node has a 360 degree coverage. Kang and Poovendran [8, 9] investigate the fractional variants of the maximum network lifetime problem for broadcast communication, proposing a polynomial time algorithm for the STB problem and some heuristics for the MTB problem. Orda and Yassour [16] improve the time complexity of the STB problem, prove that the MTB problem is NP-hard, and propose some additional MTB heuristics. Segal [18] further improves the running time of the STB problem. Additional results related to the the maximum network lifetime problem under broadcast communication can be found in [2, 12, 17]. Kalpakis *et al.* [7] consider the fractional variants of the MTC problem with full aggregation, giving a polynomial time algorithm, but the polynomial bound is of high-degree. For the same problem, Standford and Tongngam [19] give $(1-\epsilon)$-approximation algorithm with a considerably faster running time. Wu *et al.* [20] consider the convergecast problem with full aggregation and propose an online approximation algorithm, which is based on Fürer and Raghavachari approach [5] for the minimum-degree Steiner tree problem. Lin *et al.* [11] extend [20] to a more general model in which transmission power levels of nodes are adjustable. Liang and Liu [10] propose some heuristics for the online maximum network problem for convergecast.

Orda and Yassour [16] were the first to consider the complexity of the RMNL problem in *unidirectional* communication model. Under this model, they show that the fractional variant of the STB problem is NP-hard, and propose a polynomial time algorithm for the fractional variant of the MTB problem. It is not difficult to show that the discrete variant of the STC problem can be solved in polynomial time. Segal [18] shows that we can actually get a linear time algorithm for this problem. Elkin *et al.* [4] show that the discrete variant of the STB and MTB problems are NP-hard. In the same paper, they provide an $\Omega(1/\log n)$-approximation algorithm for the STB problem assuming that k_{opt} (the maximal number of rounds) is appropriately large. Nutov and Segal [15] improve the approximation ratios for the same problems. They provide a constant ratio approximation algorithm for the STB and MTB problems, as well as for the MTC problem, if k_{opt} is appropriately large. They further show that the MTC problem admits a 1/31-approximation polynomial time algorithm. The previous results for the discrete variant of the RMNL problems under the unidirectional model are summarized in Table 1.(a). The values in the table are the lower bounds on the computed number of rounds.

1.2 Our Contribution

We study the discrete variant of the *Rooted Maximum Network Lifetime* Problems for two basic communication patterns, broadcast and convergecast under the unidirectional model. We consider these problems in single and multiple topology variants. We improve the approximation ratios shown in [15] to the values given in the theorem and corollary below (See also Table 1.(b)).

Theorem 1. *For each of the problems, STB MTB and MTC there exists a polynomial time algorithm, which computes a solution for k broadcast (convergecast) rounds, such that* $k \geq \lfloor k_{opt}/\beta \rfloor$*, where*

Table 1. Our contribution and previous results for discrete variant RMNL problems

(a). Previous results for the RMNL problems

	Convergecast	Broadcast
Single Topology	k_{opt} [18]	$\lfloor k_{opt}/25 \rfloor$ [15]
Multiple Topology	$\max\{\lfloor k_{opt}/16 \rfloor, 1\}$ [15] 1/31 [15]	$\lfloor k_{opt}/36 \rfloor$ [15]

(b). Our contribution

	Convergecast	Broadcast
Single Topology	-	$\lfloor k_{opt}/5 \rfloor$
Multiple Topology	$\max\{\lfloor k_{opt}/4 \rfloor, 1\}$ 1/7	$\lfloor k_{opt}/6 \rfloor$

- $\beta = 5$ *for the STB problem;*
- $\beta = 6$ *for the MTB problem;*
- $\beta = 4$ *for the MTC problem.*

The Single Topology Maximum Network Lifetime Convergecast (STC) problem can be solved in polynomial time [18]. Therefore, for the Multiple Topology Maximum Network Lifetime Convergecast (MTC) problem, we can determine in polynomial time whether $k_{opt} \geq 1$. This yields the following corollary.

Corollary 1. *The Multiple Topology Rooted Maximum Lifetime Convergecast (MTC) problem admits a* $1/7$*-approximation polynomial time algorithm.*

Proof. Let k_{opt}^{STC} and k_{opt}^{MTC} denote the optimal number of rounds for the STC and MTC problems, respectively. We run a polynomial time algorithm for STC problem to obtain k_{opt}^{STC}. If $k_{opt}^{STC} = 0$, then $k_{opt}^{MTC} = 0$. If $k_{opt}^{STC} \geq 1$, then we run the polynomial time algorithm of Theorem 1 to get $k \geq \lfloor k_{opt}^{MTC}/4 \rfloor$ convergecast trees. Our solution for the MTC problem is now $k_{sol} = \max\{k_{opt}^{STC}, k\} \geq 1$. Because $k_{sol} \geq 1$, we have,

$$k_{sol} \geq \left\lfloor \frac{k_{opt}^{MTC}}{4} \right\rfloor \geq \frac{k_{opt}^{MTC}}{4} - \frac{3}{4} \geq \frac{k_{opt}^{MTC}}{4} - \frac{3 \cdot k_{sol}}{4}.$$

This implies that $k_{sol} \geq k_{opt}^{MTC}/7$. □

2 Notation and Preliminaries

2.1 Graph Preliminaries

For a directed graph $G = (V, E)$ and a node $v \in V$, let $\delta_G^{out}(v) = \delta_E^{out}(v)$ be the set of out-going edges from v in G, and let $\delta_G^{in}(v) = \delta_E^{in}(v)$ be the set of in-coming edges to v. For $F \subseteq E, \delta_F^{out}(v)$ is the set of out-going edges from all nodes v in F. We define $\delta_F^{in}(v)$ analogously.

A directed graph G is said to be *k-edge-outconnected from a node r* if there are k-edge-disjoint paths from the node r to any other node. A directed graph G is said to be *k-edge-inconnected to a node r* if there are k-edge-disjoint paths from every node to the node r.

An *out-arborescence* (broadcast tree) is a directed spanning tree that has a unique path from a root r to every node. An *in-arborescence* (convergecast tree) is a directed spanning tree that has a path from every node to the root r. An *arborescence* refers to either out-arborescence or in-arborescence, depending on the context.

It is well known that there are k-edge disjoint paths from a node r to all other nodes in G if and only if $\delta_G^{in}(S) \geq k$, for every subset $S \neq \emptyset \subseteq V \backslash \{r\}$.

2.2 Model

We consider a wireless ad-hoc network N consisting of n stationary nodes. Each node v is equipped with a unidirectional antenna, which only permits a single node to receive a transmitted message. Each node v has a finite amount of battery capacity. We proceed with a formal definition. Let $\mathbb{R}_+$ denote the set of non-negative numbers.

Definition 1. *A static wireless ad-hoc network $N = (V, E, w, B)$ is modeled as a weighted, directed graph (V, E), where V is a set of nodes with $|V| = n$, $E \subseteq V \times V$ is a set of directed edges, $w : E \to \mathbb{R}_+$ is an edge-weight function representing energy cost of transmissions, and $B : V \to \mathbb{R}_+$ is a battery capacity function.*

In the network N, a directed edge (u, v) exists if node u is able to directly transmit a message to node v, i.e. node v is located within the transmission range d of node u. An edge-weight $w(u, v)$ of the directed edge (u, v) denotes the amount of energy consumed to transmit one message from node u to node v. For example, we could consider $w(u, v) = d(u, v)^\alpha$, where α is a path attenuation factor, usually taken to be between 2 and 4. However, in our model, we do not assume any particular relation between the edge-weight and the distance between the nodes in the physical space. The edge weights are simply part of the input to the problems, which we consider. The battery capacity $B(v)$ denotes the current battery power of node v. To support the heterogeneity of nodes in the network, we allow the initial battery capacities to be different.

In our model, we take into account only the energy consumption of transmissions, assuming that in wireless networks the radio frequency transmission dominates the energy usage. In particular, we do not consider energy consumption for receiving and processing data. We assume that every node shares the same frequency band and the MAC layer is based on "collision-free" TDMA, so that transmissions do not interfere with each other.

2.3 Problem Definition

The *Rooted Maximum Network Lifetime* (**RMNL**) problem for two communication patterns, *broadcast* and *convergecast* can be defined as follows. The input to the RMNL problem is a network $N = (V, E, w, B)$, and a node $r \in V$. For the broadcast RMNL problem, we assume that every node can be reached from the node r while for the convergecast RMNL, we assume that the node r can be reached from every node. The output is a collection of out-arborescences $\mathcal{T}_{out} = \{T_1, ..., T_k\}$ rooted at r in the case of broadcast, or in-arborescences $\mathcal{T}_{in} = \{T_1, ..., T_k\}$ rooted at node r in the case of convergecast. In both cases, the following energy constraints have to be satisfied:

$$\sum_{i=1}^{k} w(\delta_{T_i}^{out}(v)) = \sum_{i=1}^{k} \sum_{e \in \delta_{T_i}^{out}(v)} w(e) \le B(v), \text{ for all } v \in V. \tag{1}$$

The right-hand side of (1) is the total energy used by node v over k transmission rounds, when the ith transmission round is done according to tree T_i. The objective of the problem is to maximize k. Since k represents the number of broadcast (convergecast) rounds, which can be executed within the specified battery capacities, larger k means longer network lifetime.

As mentioned earlier, the RMNL problems can be classified into two different variants, the single topology and the multiple topology. In the single topology variant, the same single out-arborescence (in-arborescence) is required for all k rounds, i.e. $T_1 = T_i$ for all $i \le k$. On the other hand, in the multiple topology variant, the out-arborescences (in-arborescences) do not have to be identical. Thus, in this paper, we consider the following *four* RMNL problems. The inputs for each of these problems is a network $N = (V, E, w, B)$, and a node $r \in V$.

Single Topology Maximum Network Lifetime Broadcast (STB) *problem*
compute maximum k and an out-arborescences T rooted at r that satisfy the energy constraints:

$$k \cdot \sum_{e \in \delta_{T}^{out}(v)} w(e) \le B(v), \text{ for all } v \in V. \tag{2}$$

Multiple Topology Maximum Network Lifetime Broadcast (MTB) *problem*
compute a maximum-size collection of out-arborescences $\mathcal{T}_{out} = \{T_1, ..., T_k\}$ rooted at r that satisfies the energy constraints (1).

Single Topology Maximum Network Lifetime Convergecast (STC) *problem*
compute a maximum k and an in-arborescence T rooted at r that satisfy the energy constraints (2).

Multiple Topology Maximum Network Lifetime Convergecast (MTC) *problem*
compute a maximum-size collection of in-arborescences $\mathcal{T}_{in} = \{T_1, ..., T_k\}$ rooted at r that satisfies the energy constraints (1).

Note that (1) becomes (2) when all k trees are the same.

3 Weighted Degree Constrained k-Connected Subgraph

We consider the RMNL approximation algirthms proposed by Nutov and Segal's [15], which find a good k using Nutov's bicriteria algorithm and binary search. We give a different analysis of the overall binary search, which leads to better approximation ratios than the ratios obtained in [15]. Nutov's algorithm [13] solves a generalized *Directed Weighted Degree Constrained Network Design* (DWDCN) problem, where the objective is to compute a minimum cost directed subgraph that satisfies specified connectivity requirements and weighted degree constraints. The DWDCN problem is actually more general than we need, so, as in [15], we consider only two special variants of this problem. One variant computes a k-edge-connected (outconnected or inconnected) subgraph, which is used to solve the multiple topology RMNL problems. The other variant computes a single out- or in-arborescence, which is used to solve the single topology RMNL problems. In both cases, the output satisfies the specified weighted degree constraints. These two special cases of DWDCN are referred to as *Weighted-Degree Constrained k-Outconnected (k-Incconected) Subgraph* and *Weighted-Degree Constrained Out-Arborescence (In-Arborescence)*, respectively, and can be defined formally as follows.

Weighted-Degree Constrained k-Outconnected (k-Inconnected) Subgraph, WDCKOS (WDCKIS)
Input: A directed weighted graph $G = (V, E, w, b)$, where V is the set of nodes, E is the set of edges, w is an edge-weight function $w : E \to \mathbb{R}_+$, and b is a degree bound function $b : V \to \mathbb{R}_+$, a root $r \in V$, and a positive integer k.
Output: A k-edge-outconnected from r (k-edge-inconnected to r) spanning subgraph H of G that satisfies the weighted degree constraints:

$$w(\delta_H^{out}(v)) \leq b(v), \text{ for all } v \in V. \tag{3}$$

Weighted-Degree Constrained Out-Arborescence (In-Arborescence), WDCOA (WDCIA)
Input: A directed weighted graph $G = (V, E, w, b)$, where V is the set of nodes, E is the set of edges, w is an edge-weight function $w : E \to \mathbb{R}_+$, and b is a degree bound function $b : V \to \mathbb{R}_+$, and a root $r \in V$.
Output: An out-arborescence (in-arborescence) H of G that satisfies the weighted degree constraints (3).

We note that WDCOA and WDCIA are actually special cases of the WDCKOS and WDCKIS problems, respectively, when $k = 1$. The WDCIA problem can be solved in linear time in the following ways. For each node $v \in V$, remove from the graph all edges outgoing from v such that $w(e) > b(v)$. Then check whether r is reachable from every node in the remaining graph.

Using Edmonds' theorem stated below, the outputs of the WDCKOS and WDCKIS problems provide solutions for the multiple topology variants of the RMNL problems (that is, for the problems, MTB and MTC respectively).

Theorem 2. *[3] Let $G = (V, E)$ be a directed graph with a specified root $r \in V$. The graph G contains k edge-disjoint spanning out-arborescences (in-arborescences) rooted at r if and only if G is k-edge-outconnected from r (k-edge-inconnected to r). Moreover, there is a polynomial time algorithm that computes such k disjoint arborescences, if they exist.*

Edmonds' theorem says that a directed graph G contains k edge-disjoint out-arborescences rooted at r if and only if G is k-edge-outconnected from r. Therefore, if we find a k-edge-outconnected spanning subgraph H of a graph G that satisfies weighted degree constraints, then Edmonds' theorem implies that we can retrieve in polynomial time k edge-disjoint spanning out-arborescences from H (so, also from G). The fastest known algorithm for computing k edge-disjoint spanning out-arborescences from a k-edge-outconnected graph runs in $O(|E|k \log |V| + |V|k^4 \log^2 |V|)$ time [1]. The same applies to k-edge-inconnected graphs and in-arborescences.

As mentioned in Section 2.1, a graph H contains k edge-disjoint paths from r to all other nodes if and only if $\delta_H^{in}(S) \geq k$ for every subset $\emptyset \neq S \subseteq V \backslash \{r\}$. Therefore, the WDCKOS problem of finding k-edge-outconnected spanning subgraph H that satisfies the weighted degree constraints (3) can be formulated as the following integer program, $P_{IP}^{OS}(k, b)$:

$$\begin{aligned} x(\delta_E^{in}(S)) &\geq k, && \text{for all } \emptyset \neq S \subseteq V\backslash\{r\}, && \text{cut constraints } (C); \\ \sum_{e \in \delta_E^{out}(v)} x(e)w(e) &\leq b(v), && \text{for all } v \in V, && \text{weighted degree constraints } (W); \\ x(e) &\in \{0, 1\}, && \text{for all } e \in E, && \text{integer constraints } (B). \end{aligned}$$

In the above formulation, for a subset of edge $F \subseteq E$, $x(F) = \sum_{e \in F} x(e)$.

The WDCKIS problem of finding k-edge-inconnected spanning subgraph and the WDCOA problem of finding an out-arborescence, which satisfy the weight degree constraints (3), can be similarly formulated as integer programs. For the former problem, the cut constraints (C) is replaced with

$$x(\delta_E^{out}(S)) \geq k, \ \text{ for all } \emptyset \neq S \subseteq V\backslash\{r\},$$

and for the latter problem with

$$x(\delta_E^{in}(S)) \geq 1, \ \text{ for all } \emptyset \neq S \subseteq V\backslash\{r\},$$

giving integer programs, which we refer to as $P_{IP}^{IS}(k, b)$ and $P_{IP}^{OA}(b)$, respectively.

It should be clear that k_{opt}^{STB} (the k_{opt} for the STB problem) is the largest k such that $P_{IP}^{OA}(B/k)$ is feasible. For the multiple topology problems, since the broadcast/convergecast trees do not have to be edge disjoint, we consider a multigraph G_k instead of graph G. The multigraph G_k is obtained from graph G by replacing each edge with its k copies, and the integer programs $P_{IP}^{OS}(k, B)$ and $P_{IP}^{IS}(k, B)$ are constructed for this multigraph G_k. Now, k_{opt}^{MTB} and k_{opt}^{MTC}

can be viewed as the largest integers k such that $P_{IP}^{OS}(k,B)$ and $P_{IP}^{IS}(k,B)$ are feasible, respectively. Consider the MTB problem (analogous remarks apply to the MTC problem). By Edmonds' theorem, we know that the solution for the integer program $P_{IP}^{OS}(k,B)$, which represents a k-edge-outconnected spanning subgraph of G_k, contains k edge-disjoint out-arborescences that satisfy the energy constraints (1). This implies that there are k out-arborescences (not necessarily edge-disjoint) in the input graph G, which satisfy the energy constraints. Conversely, if k is feasible for the MTB problem, then $P_{IP}^{OS}(k,B)$ is also feasible.

Nutov's algorithm [13] for the WDCKOS, WDCKIS, and WDCOA problems runs in polynomial time, and provides a solution that violates the weighted degree constraints (3) by at most a factor of β, where β is as given in Theorem 3 below. The algorithm initially tries to find a basic solution x for the LP-relaxation of its corresponding integer program. For the WDCKOS problem the following LP-relaxation $P_{LP}^{OS}(k,b)$ of $P_{IP}^{OS}(k,b)$ is considered (for the WDCKIS and WDCOA problems analogous LP-relaxations $P_{LP}^{IS}(k,b)$ and $P_{LP}^{OA}(b)$ are taken):

$$\begin{array}{rll} x(\delta_E^{in}(S)) \geq k, & \text{for all } \emptyset \neq S \subseteq V \backslash \{r\}, & \text{cut constraints } (C_1); \\ \sum_{e \in \delta_E^{out}(v)} x(e)w(e) \leq b(v), & \text{for all } v \in V, & \text{weighted degree constraints } (W_1); \\ 0 \leq x(e) \leq 1, & \text{for all, } e \in E, & \text{bounds } (B_1). \end{array}$$

If there is no feasible solution for the LP polytope considered, i.e, the polytope is empty, the algorithm terminates and outputs "UNFEASIBLE", meaning that there is no k-edge-outconnected spanning subgraph H of graph G that satisfies the weighted degree constraints (3) (or no k-edge-inconnected spanning subgraph, or no out-arborescence, in case of WDCKIS or WDCOA problems). Otherwise, the algorithm iteratively transforms the feasible fractional solution into an integral solution. The final integral solution satisfies the cut constraints (C) and violates the weighted degree constraints (3) by at most a factor of β. Consequently, Nutov's algorithm yields the following theorem.

Theorem 3. *[13] There exists a polynomial time algorithm which for an input instance of the* WDCKOS, WDCKIS, and WDCOA *problems computes one of the following two outcomes.*

1. *Correctly determines that the polytopes $P_{LP}^{OS}(k,b)$ or $P_{LP}^{IS}(k,b)$ for the case of the WDCKOS or WDCKIS problems, or $P_{LP}^{OA}(b)$ for the case of WDCOA problem, is empty,*
2. *If the polytope is not empty (the polytope $P_{LP}^{OS}(k,b)$, $P_{LP}^{IS}(k,b)$, or $P_{LP}^{OA}(b)$), then the algorithm finds a k-edge-outconnected spanning subgraph H (WDCKOS problem), or k-edge-inconnected spanning subgraph H (WDCKIS problem), or an out-arborescence H (WDCOA problem) that violates the weighted degree constraints (3) by at most a factor of β, that is,*

$$\sum_{e \in \delta_H^{out}(v)} w(e) \leq \beta \cdot b(v) \text{ for all } v \in V, \tag{4}$$

where:

- $\beta = 6$, *for the WDCKOS problem.*
- $\beta = 4$, *for the WDCKIS problem.*
- $\beta = 5$, *for the WDCOA problem.*

4 Algorithms for Rooted Maximum Network Lifetime

We now show how the weighted degree constraint problems, WDCKOS, WDCKIS, and WDCOA can be used to solve the rooted maximum network lifetime (RMNL) problems. For the sake of simplicity, we focus only on broadcast, that is on the MTB and STB problems. (A similar approach applies also to the convergecast problem, MTC). We start with the single topology problem (STB), and then discuss the multiple topology problem (MTB).

Let k^+ be the largest integer k such that the polytope $P^{OA}_{LP}(B/(\beta \cdot k))$ is not empty, where B is the battery capacity function in the input for STB and β is the constant given in the statement of Theorem 1 ($k^+ = 0$, if $P^{OA}_{LP}(B/\beta)$ is empty). We find k^+ by binary search, checking in each iteration whether a polytope $P^{OA}_{LP}(B/(\beta \cdot k))$ is empty using a polynomial time LP algorithm based on the ellipsoid method. If $k^+ = 0$, then we return 0. Otherwise, if $k^+ \geq 1$, we apply Nutov's algorithm of Theorem 3 for the WDCOA problem with $b = B/(\beta \cdot k^+)$. Since the polytope $P^{OA}_{LP}(B/(\beta \cdot k^+))$ is not empty, Theorem 3 implies that Nutov's algorithm returns a single out-arborescence. This out-arborescence and the number k^+ (the number of rounds) are the output for the STB problem. Observe that this out-arborescence is feasible for the STB problem, because for $b = B/(\beta \cdot k^+)$, the conditions (4) are equivalent to the energy constraints (2):

$$k^+ \cdot \sum_{e \in \delta^{out}_H(v)} w(e) \leq B(v), \quad \text{for all } \; v \in V.$$

We will show in Section 5 that always $k^+ \geq \lfloor k_{opt}/\beta \rfloor$ (in both cases when $k^+ = 0$ and when $k^+ \geq 1$).

Now consider the MTB problem, that is the multiple topology variant of the broadcast RMNL problem. Recall that the integer program $P^{OS}_{IP}(k, B)$ and the polytope $P^{OS}_{LP}(k, B/\beta)$ are constructed for the multigraph G_k. Let k^* be the largest integer k such that the polytope $P^{OS}_{LP}(k, B/\beta)$ is not empty. Similarly to the STB problem, we find k^* by binary search, checking in each iteration whether a polytope $P^{OS}_{LP}(k, B/\beta)$ is empty. If $k^* = 0$, we return an empty collection of trees. Otherwise, if $k^* \geq 1$, we apply Nutov's algorithm for the WDCKOS problem to multigraph G_{k^*} with $b = B/\beta$. Since the polytope $P^{OS}_{LP}(k^*, B/\beta)$ is not empty by the definition of k^*, Theorem 3 implies that Nutov's algorithm returns a k^*-edge-outconnected spanning subgraph H of graph G_{k^*}. Observe that this k^*-edge-outconnected spanning subgraph satisfies the energy constraints (1), because for $b = B/\beta$, the conditions (4) are equivalent to:

$$\sum_{e \in \delta^{out}_H(v)} w(e) \leq B(v), \quad \text{for all } \; v \in V.$$

From Edmonds' theorem (Theorem 2), we know that graph H contains k^* edge-disjoint out-arborescences, and we can retrieve such arborescences in time polynomial in the size of H. This gives us k^* out-arborescences (not necessarily edge-disjoint) in graph G, which satisfy the energy constraints (1). This collection of k^* out-arborescences in G is the output for the MTB problem. In Section 5 we will show that $k^* \geq \lfloor k_{opt}/\beta \rfloor$ (in both cases, when $k^* = 0$ and when $k^* \geq 1$).

It was shown in [15] that for all RMNL problems,

$$k_{opt} \leq k_{max} \overset{\mathrm{dfn}}{=} \sum_{v \in V} \frac{B(v)}{\min\{w(e) : e \in \delta_E^{out}(v), w(e) > 0\}}. \tag{5}$$

Therefore, the binary search for k^* and k^+ can be done over the range $[0, k_{max}]$, that is in $O(\log k_{max})$ iterations.

For the multiple topology problems, we have explained the algorithm in terms of the multigraph G_{k^*}. In this setting, the MTB and MTC algorithm runs in pseudo-polynomial time because the value of k^*, and consequently the size of graph G_{k^*}, can be exponential in the size of the input graph G. However, the algorithm can be modified to run in polynomial time by using the capacitated version of the definition of k-edge-connectivity and the capacitated versions of the WDCKOS and WDCKIS problems. For a directed graph $G = (V, E)$ with positive edge capacities and a distinguished root node r, we say that G is k-edge-outconnected from r if for each node $v \in V \backslash \{r\}$, there is a flow of value k from r to v. The capacitated version of WDCKOS asks for a subgraph of a capacitated graph G, which is k-edge-outconnected from r. In the capacitated case, the output of Nutov's algorithm is a directed capacitated spanning subgraph in which for every node v, there is an integral flow of value k from the root node r to v (if the corresponding LP polytope is not empty). Now, however, we cannot apply the "uncapacitated" algorithm of Edmonds' theorem for computing k out-arborescences discussed in Section 3. Instead, we can use the polynomial time algorithm for packing capacitated arborescences given in [6].

5 Analysis of the Algorithms

In this section, we prove Theorem 1, that is, we prove the lower bound of $\lfloor k_{opt}/\beta \rfloor$ on the number of trees returned by the algorithm of Section 4 (we already discussed in Section 4 that the algorithm returns feasible solutions for the MTB, MTC and STB problems). From the two multiple topology problems MTB and MTC we consider only the MTB problem. The analysis for MTC is analogous. We will consider the STB problem at the end of this section. We refer to the integer program P_{IP}^{OS} and the polytope P_{LP}^{OS} of the WDCKOS problem introduced in Section 3, dropping the superscript "OS" for simplicity of notation. (Recall again that these polytopes are defined for the multigraph G_k.)

Lemma 1. *If the polytope $P_{LP}(k, B)$ is not empty, then the polytope $P_{LP}(\lfloor k/\beta \rfloor, B/\beta)$ is also not empty, for any $\beta \geq 1$.*

Proof. Let $\langle \overline{x}(e)\rangle_{e\in E_k}$ be a feasible solution for the polytope $P_{LP}(k,B)$, where E_k is the set of edges in the multigraph G_k. Let F stand for the set $E_{\lfloor k/\beta \rfloor}$ of edges in the multigraph $G_{\lfloor k/\beta \rfloor}$. We show a feasible solution $\langle \overline{\overline{x}}\rangle_{e\in F}$ for the polytope $P_{LP}(\lfloor k/\beta \rfloor, B/\beta)$, which is defined by the following constraints:

$$x(\delta_E^{in}(S)) \geq \lfloor k/\beta \rfloor, \quad \text{for all } \emptyset \neq S \subseteq V\backslash\{r\}, \qquad \text{cut constraints } (C_2);$$

$$\sum_{e\in\delta_F^{out}(v)} x(e)w(e) \leq B(v)/\beta, \quad \text{for all } v\in V, \quad \text{weighted degree constraints } (W_2);$$

$$0 \leq x(e) \leq 1, \qquad \text{for all } e \in F, \qquad \text{bounds } (B_2).$$

For an edge $e \in E$, let $\overline{e}_1, \overline{e}_2, \ldots, \overline{e}_k$ and $\overline{\overline{e}}_1, \overline{\overline{e}}_2, \ldots, \overline{\overline{e}}_{\lfloor k/\beta \rfloor}$ be the copies of e in multigraphs G_k and $G_{\lfloor k/\beta \rfloor}$, respectively. If $\left(\sum_{i=1}^{k} \overline{x}\left(\overline{e}_i\right)\right)/\beta \leq \lfloor k/\beta \rfloor$, then we set $\overline{\overline{x}}(\overline{\overline{e}}_1), \ldots \overline{\overline{x}}\left(\overline{\overline{e}}_{\lfloor k/\beta \rfloor}\right)$ in such a way that $0 \leq \overline{\overline{x}}\left(\overline{\overline{e}}_i\right) \leq 1$ for each $i = 1, \ldots, \lfloor k/\beta \rfloor$ and $\sum_{i=1}^{\lfloor k/\beta \rfloor} \overline{\overline{x}}\left(\overline{\overline{e}}_i\right) = \left(\sum_{i=1}^{k} \overline{x}\left(\overline{e}_i\right)\right)/\beta$. On the other hand, if $\left(\sum_{i=1}^{k} \overline{x}\left(\overline{e}_i\right)\right)/\beta > \lfloor k/\beta \rfloor$, then we set $\overline{\overline{x}}\left(\overline{\overline{e}}_i\right) = 1$, for each $i = 1, \ldots, \lfloor k/\beta \rfloor$.

It is clear that the solution $\langle \overline{\overline{x}}(e)\rangle_{e\in F}$ satisfies the bounds (B_2) and it is not difficult to see that it also satisfies the weighted degree constraints (W_2). We show now that $\langle \overline{\overline{x}}(e)\rangle_{e\in F}$ satisfies also the cut constraints (C_2). Take any $\emptyset \neq S \subseteq V\backslash\{r\}$. If for each edge $e \in \delta_E^{in}(S)$, $\left(\sum_{i=1}^{k} \overline{x}\left(\overline{e}_i\right)\right)/\beta \leq \lfloor k/\beta \rfloor$, then $\overline{\overline{x}}\left(\delta_F^{in}(S)\right) = \left(\overline{x}\left(\delta_{E_k}^{in}(S)\right)\right)/\beta \geq k/\beta \geq \lfloor k/\beta \rfloor$. On the other hand, if there is an edge $e \in \delta_E^{in}(S)$ such that $\left(\sum_{i=1}^{k} \overline{x}\left(\overline{e}_i\right)\right)/\beta > \lfloor k/\beta \rfloor$, then $\overline{\overline{x}}\left(\delta_F^{in}(S)\right) \geq \sum_{i=1}^{\lfloor k/\beta \rfloor} \overline{\overline{x}}(\overline{\overline{e}}_i) = \lfloor k/\beta \rfloor$. □

Lemma 2. *The polytope $P_{LP}\left(\lfloor k_{opt}/\beta \rfloor, B/\beta\right)$ is not empty.*

Proof. The integer program $P_{IP}(k_{opt}, B)$ is feasible. This implies that the polytope $P_{LP}(k_{opt}, B)$ is not empty as the LP-relaxation of $P_{IP}(k_{opt}, B)$. Lemma 1 implies that the polytope $P_{LP}\left(\lfloor k_{opt}/\beta \rfloor, B/\beta\right)$ is also not empty. □

Lemma 3. *The algorithm for the multiple topology maximum network lifetime broadcast problem MTB returns a solution with k broadcast trees such that $k \geq \lfloor k_{opt}/\beta \rfloor$.*

Proof. We show that the solution k^* is at least $\lfloor k_{opt}/\beta \rfloor$. The algorithm for the MTB problem always returns a feasible solution (either $k^* = 0$ or a collection of out-arborescences, which is feasible for k^* rounds). By definition, k^* is the largest integer k such that the polytope $P_{LP}(k, B/\beta)$ is not empty. Therefore, we know that the polytope $P_{LP}(k^*+1, B/\beta)$ is empty. From Lemma 2, we know that the polytope $P_{LP}\left(\lfloor k_{opt}/\beta \rfloor, B/\beta\right)$ is not empty. Thus, $\lfloor k_{opt}/\beta \rfloor < k^* + 1$, so, $\lfloor k_{opt}/\beta \rfloor \leq k^*$. □

Lemma 4. *The algorithm for the single topology maximum network lifetime broadcast problem STB returns a solution with k broadcast trees such that $k \geq \lfloor k_{opt}/\beta \rfloor$.*

Proof. We show that the solution k^+ is at least $\lfloor k_{opt}/\beta \rfloor$. The algorithm for the STB problem always returns a feasible solution (either $k^+ = 0$ or a broadcast tree, which is feasible for k^+ rounds). This implies that if $k_{opt} = 0$, then $k^+ = 0$ as well. Assume now that $k_{opt} \geq 1$. Hence, the integer program $P_{IP}(B/k_{opt})$ is feasible. This implies that the polytope $P_{LP}(B/k_{opt})$ is not empty as it is the LP-relaxation of $P_{IP}(B/k_{opt})$. Recall the definition of k^+, which is the largest integer k such that the polytope $P_{LP}(B/(\beta \cdot k))$ is not empty. Therefore, we know that the polytope $P_{LP}(B/(\beta \cdot (k^+ + 1)))$ is empty. Therefore, $k_{opt} < \beta \cdot (k^+ + 1)$, so $\lfloor k_{opt}/\beta \rfloor \leq k^+$. □

6 Conclusion

We considered the discrete variant of the Rooted Maximum Network Lifetime (RMNL) problems for broadcast and convergecast under the unidirectional model. We considered these problems in both single and multiple topology variants. Our analysis of Nutov and Segal's [15] algorithms improves approximations ratios for these problems. Our analysis can be also applied to the *Maximum Integral Flows with Energy constraints* problem considered in [14]: given a network N and two nodes s and t, the objective is to find a maximum integral st-flow that satisfies the energy constraints of the nodes. Our analysis can improve the approximation ratio of the algorithm given in [14] from 1/16 to 1/4, if k_{opt} is large.

We conclude with suggestions for some possible further research. One direction is to investigate whether the approximation ratios which we have obtained can be further improved. The approach we consider is an iterative process, where in each iteration the LP-relaxation of an appropriate IP problem has to be solved. Therefore, this approach seems to have high computational costs for large networks. Hence, a practical approach for solving the RMNL problems would be worth further investigation. Another direction is to solve these problems by distributed algorithms. In this study we have considered a centralized approach, in which a prior knowledge of the full network topology is assumed. However, due to the decentralized nature of ad hoc networks, a distributed approach would be more desirable. It would be also interesting to develop models, which would more accurately reflect energy consumption in wireless networks and other general properties of such networks.

References

1. Bhalgat, A., Hariharan, R., Kavitha, T., Panigrahi, D.: Fast edge splitting and edmonds' arborescence construction for unweighted graphs. In: SODA, pp. 455–464 (2008)
2. Deng, G., Gupta, S.K.S.: Maximizing broadcast tree lifetime in wireless ad hoc networks. In: GLOBECOM (2006)
3. Edmonds, J.: Edge-disjoint branchings. In: Rustin, B. (ed.) Combinatorial Algorithms, pp. 91–96. Academic Press (1973)

4. Elkin, M., Lando, Y., Nutov, Z., Segal, M., Shpungin, H.: Novel algorithms for the network lifetime problem in wireless settings. Wireless Networks 17(2), 397–410 (2011)
5. Fürer, M., Raghavachari, B.: Approximating the minimum-degree steiner tree to within one of optimal. J. Algorithms 17(3), 409–423 (1994)
6. Gabow, H.N., Manu, K.S.: Packing algorithms for arborescences (and spanning trees) in capacitated graphs. Math. Program. 82, 83–109 (1998)
7. Kalpakis, K., Dasgupta, K., Namjoshi, P.: Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. Computer Networks 42(6), 697–716 (2003)
8. Kang, I., Poovendran, R.: Maximizing static network lifetime of wireless broadcast adhoc networks. In: IEEE International Conference on Communications, ICC 2003, pp. 2256–2261 (2003)
9. Kang, I., Poovendran, R.: Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. MONET 10(6), 879–896 (2005)
10. Liang, W., Liu, Y.: Online data gathering for maximizing network lifetime in sensor networks. IEEE Trans. Mob. Comput. 6(1), 2–11 (2007)
11. Lin, H.C., Li, F.J., Wang, K.Y.: Constructing maximum-lifetime data gathering trees in sensor networks with data aggregation. In: ICC, pp. 1–6 (2010)
12. Maric, I., Yates, R.D.: Cooperative multicast for maximum network lifetime. IEEE Journal on Selected Areas in Communications 23(1), 127–135 (2005)
13. Nutov, Z.: Approximating directed weighted-degree constrained networks. In: APPROX-RANDOM, pp. 219–232 (2008)
14. Nutov, Z.: Approximating maximum integral flows in wireless sensor networks via weighted-degree constrained k-flows. In: DIALM-POMC, pp. 29–34 (2008)
15. Nutov, Z., Segal, M.: Improved Approximation Algorithms for Maximum Lifetime Problems in Wireless Networks. In: Dolev, S. (ed.) ALGOSENSORS 2009. LNCS, vol. 5804, pp. 41–51. Springer, Heidelberg (2009)
16. Orda, A., Yassour, B.A.: Maximum-lifetime routing algorithms for networks with omnidirectional and directional antennas. In: MobiHoc, pp. 426–437 (2005)
17. Park, J., Sahni, S.: Maximum lifetime broadcasting in wireless networks. In: AICCSA, p. 8. IEEE Computer Society (2005)
18. Segal, M.: Fast algorithm for multicast and data gathering in wireless networks. Inf. Process. Lett. 107(1), 29–33 (2008)
19. Stanford, J., Tongngam, S.: Approximation algorithm for maximum lifetime in wireless sensor networks with data aggregation. In: SNPD, pp. 273–277 (2006)
20. Wu, Y., Fahmy, S., Shroff, N.B.: On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm. In: INFOCOM, pp. 356–360 (2008)

Distributed Geometric Distance Estimation in Ad Hoc Networks

Sabrina Merkel, Sanaz Mostaghim, and Hartmut Schmeck

Institute AIFB, Karlsruhe Institute of Technology (KIT)
76128 Karlsruhe, Germany
{sabrina.merkel,sanaz.mostaghim,hartmut.schmeck}@kit.edu
www.aifb.kit.edu

Abstract. Distributed localization algorithms for nodes in ad hoc networks are essential for many applications. A major task when localizing nodes is to accurately estimate distances. So far, distance estimation is often based on counting the minimum number of nodes on the shortest routing path (hop count) and presuming a fixed width for one hop. This is prone to error as the length of one hop can vary significantly. In this paper, a distance estimation method is proposed, which relies on the number of shared communication neighbors and applies geometric properties to the network structure. It is shown that the geometric approach provides reliable estimates for the distance between any two adjacent nodes in a network. Experiments reveal that the estimation has less relative percentage error compared to a hop based algorithm in networks with different node distributions.

Keywords: Ad Hoc networks, localization, distance estimation.

1 Introduction

Mobile ad hoc networks (MANETs), a network of devices with local communication ability and without a fixed topology have been more and more subject to research. In such networks, adding a GPS-receiver to the devices might not always be desirable, for example due to power consumption or cost issues. In addition, GPS does not help in indoor or underwater scenarios. Nevertheless, location-awareness plays an important role such as for the allocation of event reporting in a monitoring sensor network [1–3], location dependent routing [4–8] assistance of group querying [9], pattern formation [10,11] and many more. For that reason, alternative localization techniques were proposed to derive the location of each device in the network (cf. [12–14]).

Many of these algorithms use a small number of so called anchor nodes which are assumed to know their own coordinates either through a GPS-receiver or due to a priori configuration. Examples for such algorithms are given in [15–20]. Many of these algorithms rely on an estimate of the distance between each node and the anchors to calculate the nodes' coordinates. There are several methods for estimating distances in ad hoc networks. The most commonly addressed approach uses the strength of the radio frequency signal [21–24] or the time-of-flight

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 28–41, 2012.

analyzes of the signal [25,26]. Both technique require suitable hardware which might not always be available. To avoid this problem, mathematical approaches have been developed, mostly counting communication hops between the node and an anchor and multiplying this value with an estimate for the width of one hop [21,27–31]. Different from the existing approaches, the main idea of GeoDE is based on estimating the distance between two adjacent nodes taking into account the individual local conditions.

The idea to derive a distance estimate from the number of shared communication partners was first presented in [32], and later on refined in [33, 34]. In [33] the ratio of shared to total communication partners was used for the first time and the mapping between this ratio and the distance of two adjacent nodes was derived through empirical studies. In [34] a first order Taylor series expansion is applied to approximate the mapping function. Here, an alternative approach to the approximation in [34] is proposed. Furthermore, a technique of averaging estimation results between neighbors is introduced and it is shown that this improves the robustness in non-uniformly, distributed networks. Additionally, an algorithm is presented to derive long range distance estimation from the estimates between adjacent nodes which can be used to estimate distances to remote anchor nodes for subsequent localization, for example using multilateration [15]. Experiments are conducted to analyze the performance of GeoDE and the influence of identified error sources. GeoDE is examined in two scenarios (a) computing the distance between any two neighbors in the network and (b) computing the distance between all nodes and one anchor node. The behavior of GeoDE is tested in different network scenarios and for varying signal radius of the devices. The results indicate that the estimation using GeoDE is more accurate than estimates derived by a hop based algorithm.

The applied model of an ad hoc network assumes randomly distributed devices on a two dimensional obstacle free plane. The devices do not have global knowledge of the network topology or their locations. Each device can communicate with adjacent devices, i.e. all devices in its neighborhood. The neighborhood of a device is defined as a physical neighborhood on the plane within a fixed distance r from the device. The radius r is identical and known to all devices and assumed to be much smaller than the dimensions of the plane. All devices are assumed to have the same properties (homogeneous devices), except for anchor devices which posses knowledge of their own positions. Even though mobility is not regarded in this paper, the adjustment of the presented distance estimation algorithm to a mobile network is straightforward.

This paper is structured as follows. In Section 2, the GeoDE algorithm is specified. Section 3 presents the experiments' settings and displays and discusses the results. Section 4 concludes the paper.

2 Distributed Geometric Distance Estimation (GeoDE)

The basic idea of GeoDE is to approximately determine the common surface of two overlapping communication areas by the ratio of shared to total neighbors.

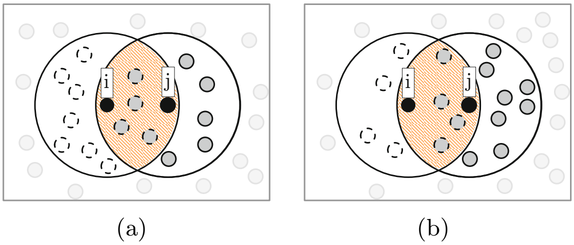


Fig. 1. Two examples for adjacent nodes i, j and their neighborhoods. Nodes with dotted lines belong to N_i, Grey filled nodes to N_j. S_{ij} are nodes in the shaded area.

Knowing the overlapping surface O, the distance between the two communicating nodes can be derived. The distance can then be used as input for the localization algorithms presented in Section 1 to obtain coordinates for each device. In this section, it is shown how to estimate the overlapping surface of the communication area of two adjacent nodes and the necessary steps to derive an estimate for the distance between the two nodes. The requirements for the GeoDE algorithm are that each node knows all its neighbors and can communicate with them. For node i to derive the distance to its neighbor j applying GeoDE, the neighbors of node i have to be distinguished with respect to j as follows:

Definition 1 (Classification of Neighbors). *Let i, j be two adjacent nodes and N_i, N_j the sets of nodes situated in the neighborhood of i and j respectively. The neighbors of i can be categorized with respect to j as:*

$$\textit{shared neighbors: } S_{ij} := (N_i \cap N_j)$$

$$\textit{individual neighbors: } I_{ij} = (N_i \backslash S_{ij})$$

Figure 1 shows two examples for adjacent nodes i and j and the corresponding classification of their neighbors.

The network structure of two adjacent nodes and their communication areas can be mapped to the geometrical shape of two overlapping circles. The problem to determine the distance between the adjacent nodes is hence transfered to computing the distance between the corresponding circles' centers. The ratio of shared S_{ij} to total neighbors N_i of a node i might deliver a good estimate for the ratio of overlapping to total circular surface area. Assuming this correlation holds, the surface of the overlapping area O can be estimated from the perspective of node i as $O \approx \pi r^2 \cdot \frac{|S_{ij}|}{|N_i|}$.

The circles' cut surface O has the shape of a concave lens or a mirrored circular segment with surface A (cf. Figure 2), with:

$$A \approx 0.5 \cdot \pi r^2 \cdot \frac{|S_{ij}|}{|N_i|} \tag{1}$$

When two circles of the same surface overlap, the cut's surface O should be inverse proportional to the distance d between the circles' centers. The segment

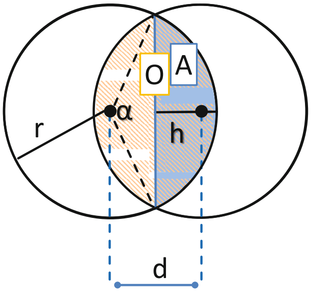


Fig. 2. Geometric characteristics of two overlapping circles

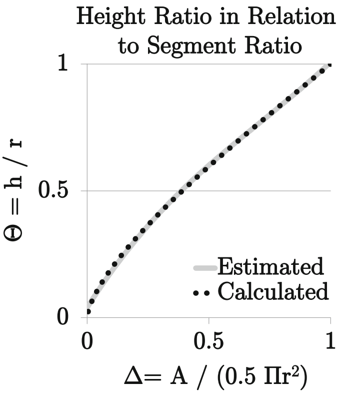


Fig. 3. Relation of θ to Δ and the approximated third-degree polynomial function f

surface A can be calculated from a known radius r and a segment height h using the standard equation (2):

$$A = r^2 \arccos(1 - \frac{h}{r}) - \sqrt{2rh - h^2}(r - h) \tag{2}$$

With known A and r one could try to derive the value of h from equation (2). The segment height h can be mapped to the distance d between the circles' centers with known r. The distance between the center of the circle and the chord is equal to $r - h$. Therefore, the distance between the two centers can be obtained by:

$$d = 2 \cdot (r - h) \tag{3}$$

Resolving Equation (2) to h is not feasible. In [34] the first order Taylor series expansion is used to approximate equation 2 but with the following considerations an alternative solution is possible. As Equation (2) depends on h and r there is no 2-dimensional representation that could be approximated by using regression. Nevertheless, the following considerations help to solve this problem. The height h of a segment can be described as a ratio θ of the circle's radius r and the segment area A is a portion of half the circle's surface:

$$\theta = \frac{h}{r} \tag{4}$$

$$\varDelta = \frac{A}{0.5 \cdot \pi r^2} \tag{5}$$

In the following we show that $\varDelta$ and θ are independent of r with the result that the relationship between $\varDelta$ and θ can be approximated using regression.

The standard equations (7) and (6) describe A and h depending on r and angle α (cf. Figure 2).

$$A = \frac{r^2}{2} \cdot (\alpha - \sin(\alpha)) \qquad (6) \qquad\qquad h = r \cdot (1 - \cos(\frac{\alpha}{2})) \qquad (7)$$

Substitution A and h by rearranging Equations (5) and (4), it becomes apparent that Δ and θ only depend on α, which has a fixed value range, but are independent from r.

The relation of Δ and θ can be approximated using regression. Figure 3 shows data points (Grey line) and the approximated third-degree polynomial function $f : \Delta \rightarrow \theta$ (dotted line) derived through polynomial regression. Apparently, f is an almost perfect approximation of the relationship between Δ and θ.

From the approximated function f, an estimate for the segment height h and, thus, the distance d can be calculated with known Δ:

$$d = 2r(1 - 2 \cdot f(\Delta)) \qquad (8)$$

As stated before, A can be estimated from the relation between shared neighbors S_{ij} to total neighbors N_i which can be computed locally using Equation (1).

Putting it all together, Equation (9) calculates the distance estimate $\hat{d}_{ij}$ for node i to its adjacent neighbor j, given the number of shared neighbors $|S_{ij}|$, total neighbors $|N_i|$ and r.

$$\hat{d}_{ij} = r \cdot (a \cdot (\frac{|S_{ij}|}{|N_i|})^3 + b \cdot (\frac{|S_{ij}|}{|N_i|})^2 + c \cdot (\frac{|S_{ij}|}{|N_i|}) + e))) \qquad (9)$$

Using regression to determine the polynomial f and further computations, the coefficients of the above equation can be estimated as follows:

$$a = 3.90 \qquad b = -4.16 \qquad c = 3.04 \qquad e = 0.04$$

2.1 Evaluation of GeoDE

The accuracy of the proposed GeoDE approach depends on two factors. Firstly, the approximation of A using Equation (1) depends on the distribution of neighbors in the communication area as well as the neighborhood size N_i, secondly, the approximation of function f using polynomial regression is a source of error.

The assumption underlying the GeoDE approach is that the number of nodes within an area of the environment can be mapped to the size of this area. This is a critical assumption when the distribution of nodes is imbalanced. As a result the ratio of shared to individual neighbors might not reflect the relation of overlapping to total circular area anymore. Figure 1(b) illustrates this effect. Also, the neighborhood size N_i determines the possible precision for estimating Δ. There are $|N_i| + 1$ possible estimates for the ratio of segment surface area to total area Δ. The margin between these values is $\frac{1}{|N_i|}$. The resulting possible absolute error for the estimation of Δ lies within the interval $[0, \frac{1}{|N_i|})$. From Equation (8) and (9) the maximum absolute distance estimation error induced

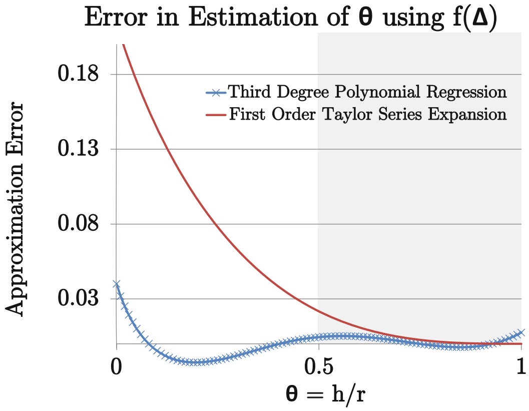


Fig. 4. The approximation error of function f

by a small neighborhood size can be calculated as $\epsilon \in [0, (28a + 12b + 4c)r)$ with $|N_i| = 1$ and $\Delta \to 1$. The impact of the nodes' distribution is assessed in the experiments shown in Section 3.

The other source of error concerns the approximation of function f. Figure 4 shows the deviation between the approximation $f(\Delta)$ and the corresponding calculated values of θ for different values of θ. Also, the approximation error using first order Taylor series expansion as suggested in [34] is printed for comparison. As Figure 4 indicates, the approximation error of function f is at most of 0.04, which leads to a maximum absolute distance estimation error of $0.16r$. The actual error depends on the ratio of height h to radius r and, as the height is coupled with the distance d. It follows that estimating the same distance with different radii r can lead to different estimation errors. Nevertheless, at least for $\theta < 0.9$ the error using the polynomial approximation is smaller than using the first order Taylor series as suggested in [34].

2.2 Distributed GeoDE Algorithm for Ad Hoc Networks

In principle, the distance estimate $\hat{d}_{ij}$ can range between 0 and r as the centers of two overlapping circles have a maximum distance of $2r$. This ignores the fact, that adjacent nodes can have a maximum distance of r to be able to communicate. Therefore, using this concept in a network, $\hat{d}_{ij}$ can be restricted to a maximum value of r. This corresponds to a limited height $h \in [0.5r, r]$ and, therefore, the approximation error of function f is limited to the section highlighted in Grey in Figure 4.

As neighborhoods of i and j, N_i and N_j, commonly differ in size (cf. Figure 1 for an example), node i and node j calculate different estimates for the distance between them. An improved approximation can be obtained when node i and node j exchange their estimates via communication and calculate the average of $\hat{d}_{ij}$ and $\hat{d}_{ji}$.

This leads to the following algorithm computed by node i to estimate its distance to the adjacent node j using the GeoDE approach:

Algorithm 1. CalcDistToNeighbor(i, j)

```
// Computing the distance between $i$ and a neighbor $j$
Input: node $i$ and node $j$
Output: estimated distance $\hat{d}_{ij}$
$N_i$ = set of neighbors nodes
Ask neighbor $j$ to send its set of neighbors $N_j$
Compute the shared neighbors $S_{ij}$
Let $x := \frac{|S_{ij}|}{|N_i|}$
$\hat{d}_{ij} = r \cdot (3.90 \cdot x^3 - 4.16 \cdot x^2 + 3.04 \cdot x + 0.04)$
Limitation: If $(\hat{d}_{ij} > r)$ Then $\hat{d}_{ij} = r$
Averaging: Ask $j$ for $\hat{d}_{ji}$ and if available compute $\hat{d}_{ij} = 0.5 \cdot (\hat{d}_{ij} + \hat{d}_{ji})$
```

To transfer the presented concept to a long range distance estimation between a node i and an anchor node a, all distances along the shortest path between both nodes are aggregated. The assumption is that all nodes in the network estimate their distance to the anchor node a, which is the case for all eligible localization algorithms (cf. Section 1). The distance between a node i and an anchor a can be computed using Algorithm 2.

Algorithm 2. CalcDistToAnchor(i, a)

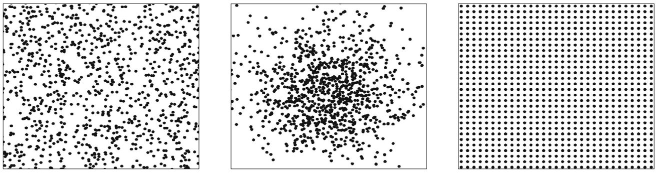
```
// Computing the distance between $i$ and an anchor $a$
Input: node $i$ and node $a$
Output: estimated distance $\hat{d}_{ia}$
$N_i$ = set of neighbors of node $i$
If anchor $(a \in N_i)$ Then $\hat{d}_{ia}$ =CalcDistToNeighbor(i, a)
Else search for neighbor $k$ closest to $a$:
    Ask all neighbors $j \in N_i$ for their estimate $\hat{d}_{ja}$ = CalcDistToAnchor(j, a), j)
    Find neighbor $k$ with minimal estimate: $k = argmin(\hat{d}_{ja}, j)$
    Compute distance to $k$: $\hat{d}_{ik}$ =CalcDistToNeighbor(i, k)
    Aggregate distances: $\hat{d}_{ia} = \hat{d}_{ik} + \hat{d}_{ka}$
End If
```

For comparison, in [15], the distance $\hat{d}_{ia}$ between a node i and the anchor a is estimated as:

$$\hat{d}_{ia} = (\frac{\sum_{j \in N_i} h_{ja} + h_{ia}}{|N_i| + 1} - 0.5) \cdot r \tag{10}$$

h_{ia} denotes the hop count of node i to the anchor a.

Note that in both algorithms each node's calculation depends on other nodes' results. Therefore, the algorithm has to be executed iteratively before a stable



(a) *Scenario 1* (b) *Scenario 2* (c) *Scenario 3*

Fig. 5. Positioning according to a uniform random distribution (a), a Gaussian random distribution (b), and evenly distributed nodes (c)

estimate is achieved. The necessary number of executions is subject to the neighborhood size and the number of nodes that lie on the shortest path between i and a. In mobile networks the algorithm can be executed repeatedly to dynamically compute the distance estimate considering changes in the locations of node i or a respectively.

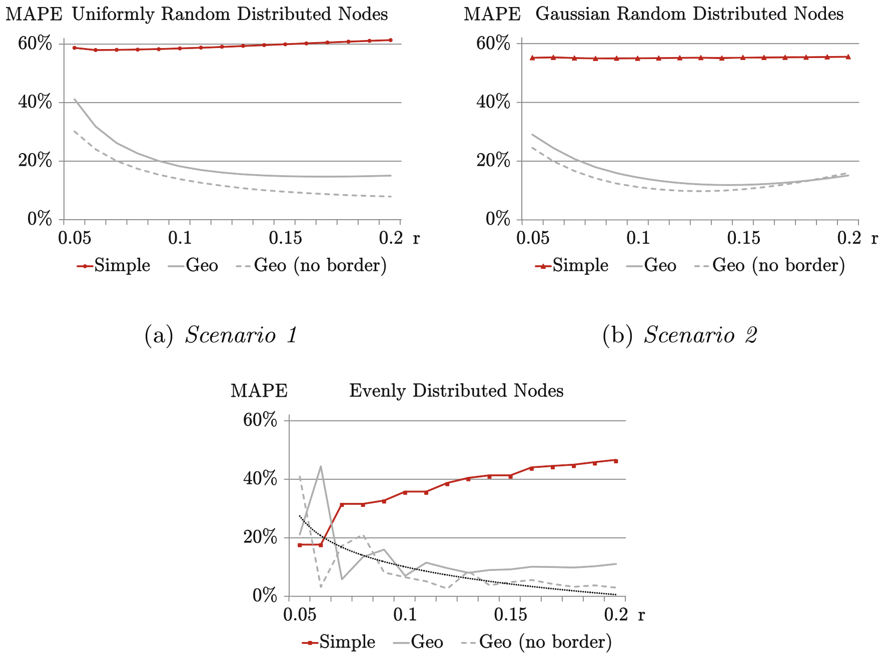
3 Experiments

GeoDE relies on the idea that the ratio of shared to total neighbors can be used as an estimate for the ratio of overlapping to total surface of the communication area. In this section experiments are presented to evaluate whether this assumption holds for a variety of network topologies. The second part of the experiments concerns the usage of the GeoDE approach to estimate distances to anchor nodes. The results are compared with the results of the hop count based approach presented in [29].

For the experiments a 2-dimensional square environment of size 1.0 x 1.0 units containing 1000 nodes is considered. The neighborhood size and the distribution of nodes is expected to influence the quality of GeoDE. Therefore, three different scenarios for the nodes' distribution in the environment are considered. Two randomly distributed networks are investigated using a uniform random distribution in *Scenario 1* and a Gaussian random distribution in *Scenario 2*. In *Scenario 3*, the nodes are evenly positioned in a grid-like shape (cf. Figure 5). These scenarios were selected to investigate the influence of imbalanced distribution of neighbors in the communication area. In addition, different values for the communication radius r were tested to investigate the influence of the neighborhood size which was identified to be a potential source of error (cf. Section 2.1).

3.1 Distance Estimation between Neighbors

In the first set of experiments, every node estimates its distance to all adjacent nodes using the GeoDE approach. For comparison, the average distance between adjacent nodes in the considered scenarios is taken as reference. To evaluate the quality of the estimates, the mean absolute percentage error (MAPE) is



(a) *Scenario 1*

(b) *Scenario 2*

(c) *Scenario 3*

Fig. 6. MAPE using the geometric approach (Geo) compared to the error when using the average distance as an estimate (Simple)

calculated as $MAPE(\hat{d}_{ij}) = \frac{|d_{ij}-\hat{d}_{ij}|}{d_{ij}}$, where d_{ij} denotes the euclidean distance between a node i and its neighbor j and $\hat{d}_{ij}$ denotes the estimate of that distance. The MAPE gives information about the relative deviation of the estimate with respect to the real distance. As nodes near the border of the environment have a cropped communication area, all experiments were repeated using only inner nodes in order to illustrate the influence of border nodes on the network's average estimation error.

The results for *Scenario 1* are shown in Figure 6(a). The GeoDE delivers estimation results ranging between 40% up to approximately 15% (10% for inner nodes) deviation from the real distance which is consistently less error-prone than estimating the distance using the average of the network. The results indicate that the GeoDE approach delivers reliable estimates for distances between adjacent nodes. Furthermore, the quality of the estimation improves with increasing communication radius r. This can be explained by the entailed growth of the number of neighbors.

Figure 6(b) shows the MAPE for distance estimation between any two adjacent nodes in a Gaussian random distributed network. In contrast to what one might intuitively expect, the geometrical estimation performs even better as in uniformly random distributed networks despite the imbalanced distribution of nodes. The reason lies in averaging the estimates of both involved nodes. An unbalanced distribution of nodes leads to an overestimation in one node and an underestimation in the other node which may, under certain circumstances, provide a good estimate on average. Another factor for the less error-prone estimates in the Gaussian distributed network is the larger average neighborhood size due to the concentration of nodes in the center of the environment.

For scenario 2, it is further noticeable, that the percentage error does not decrease continuously with rising radius r, which seemed to be the case for uniformly random distributed networks. Instead, the curve has a convex shape. This is due to the approximation error of f. As stated before, the estimation error induced by approximating the function f depends on θ, i.e. the ratio of height h to radius r. For all experiments θ ranges between (0.61, 0.69), thus the closest zero-error point $\theta*$ lies approximately at $\theta* = 0.745$ (cf. Figure 4). Figure 7 shows the average percentage deviation for all considered node distributions and radii from this zero-error-point. The experiments with Gaussian distributed nodes diverge stronger with increasing radius than the experiments with uniformly random distributed nodes, which explains the convex behavior of the MAPE curve.

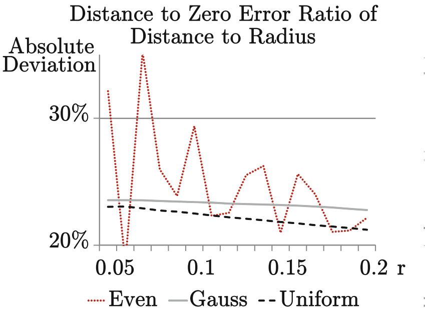


Fig. 7. Percentage deviation between θ and $\theta*$

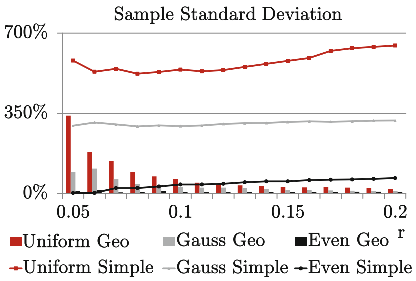


Fig. 8. Sample standard deviation for GeoDE between neighbors

Figure 6(c) shows the results for Scenario 3. Intuitively one would expect a similar MAPE as in uniformly random distributed networks, as the distribution of nodes is very balanced in both scenarios. Nevertheless, this does not appear to be the case at first sight, but when looking at the trendline (black dotted line) the behavior is quite similar. The oscillating error can be explained by the step-like increase of the average distance d due to the symmetric arrangement (cf. Figure 9) in combination with the afore mentioned distance dependent error of the approximated function f.

Figure 8 illustrates the sample standard deviation for the previously presented experiments. It shows that the standard deviation is relatively small compared

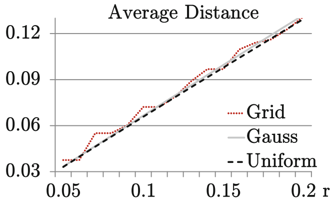
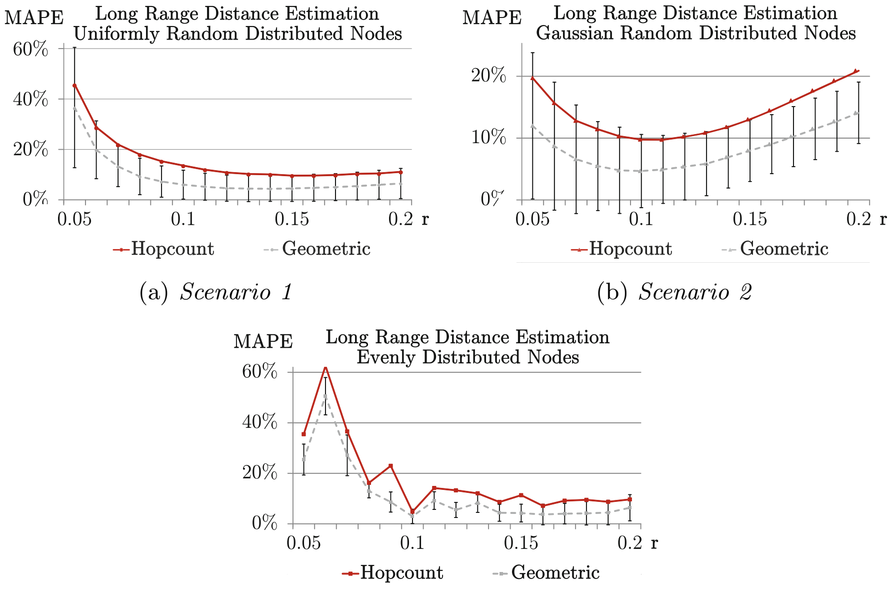


Fig. 9. Average distances between adjacent nodes in networks with different distributions depending on the communication radius r



(a) *Scenario 1*

(b) *Scenario 2*

(c) *Scenario 3*

Fig. 10. MAPE for geometric versus traditional approach on long distance estimation including standard sample deviation

to the estimates using the average distance. This further substantiates the observation that the geometric concept is successfully transferred to the network topology delivering reliable estimates for each regarded distance estimation and not only on average for the whole network.

3.2 Distance Estimation to Anchor Nodes

The second set of experiments has the objective to evaluate the GeoDE concept for the estimation of distances to anchor nodes. Therefore, an anchor node is randomly chosen in each experiment iteration and all other nodes estimate their distance to this anchor node according to Algorithm 2 (cf. Section 2.2). For comparison, the hop count based distance estimation described in [29] is used. This method has been successfully used for localization in [15] and does not require more than one anchor node for distance estimation as opposed to the DV-hop propagation model in [27].

Figure 10(a) shows the MAPE for *Scenario 1*, using the uniform random distribution for node positioning. Figure 10(b) for *Scenario 2*, the Gaussian randomly distributed network and Figure 10(c) for *Scenario 3*, with evenly distributed nodes. It can be observed that the GeoDE approach leads to less error-prone estimates than the hop count based estimation for all considered distributions and radii. Furthermore, it should be noted that even the sample standard deviation is much less or equal to the MAPE of hop count based estimates. This confirms that the GeoDE approach is a consistent improvement in distance estimation for all considered ad hoc network scenarios and radii.

4 Conclusion and Future Work

This paper presents a new approach for a distributed distance estimation in an ad hoc network. The method relies on the ratio of shared to total neighbors and applies geometric coherences to the network structure. Three sources for error in the GeoDE approach were identified and, where possible, quantified. Experiments were conducted to investigate the absolute percentage error of the distance estimates in three different network scenarios: uniformly random, Gaussian random, and evenly distributed nodes. The results were compared to a hop count based estimation approach, showing that the GeoDE reliably delivers more precise estimates. This observation was consistent for all investigated communication radii and node distribution scenarios. Furthermore, even the sample standard deviation for GeoDE is close to the average percentage error of the hop count based approach and lies below it for some considered experiment settings. In future work, the GeoDE method is to be investigated for the usage in localization algorithms. We expect to improve the accuracy of the established coordinate system with the GeoDE as a great part of the error in finding coordinates is due to inaccuracy in distance estimation. Besides, the robustness of the algorithm is to be tested under mobile conditions.

References

1. Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., Estrin, D.: Habitat monitoring with sensor networks. Communications of the ACM 47(6), 34–40 (2004)
2. Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J., Welsh, M.: Fidelity and yield in a volcano monitoring sensor network. In: Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI 2006), pp. 381–396. USENIX Association (2006)

3. Cui, J.H., Kong, J., Gerla, M., Zhou, S.: The challenges of building mobile underwater wireless networks for aquatic applications. IEEE Network 20(3), 12–18 (2006)
4. Maihofer, C.: A survey of geocast routing protocols. IEEE Communications Surveys Tutorials 6(2), 32–42 (2004)
5. Li, J., Jannotti, J., De Couto, D.S.J., Karger, D.R., Morris, R.: A scalable location service for geographic ad hoc routing. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 120–130. ACM (2000)
6. Amouris, K.N., Papavassiliou, S., Li, M.: A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In: Proceedings of the 49th IEEE Conference on Vehicular Technology, vol. 2, pp. 1365–1369. IEEE (1999)
7. Navas, J.C., Imielinski, T.: GeoCast - Geographic Addressing and Routing. In: Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking MobiCom 1997, pp. 66–76. ACM (1997)
8. Liao, W., Tseng, Y., Sheu, J.: GRID: a fully location-aware routing protocol for mobile ad hoc networks. Telecommunication Systems 18(1-3), 37–60 (2001)
9. Gehrke, J., Madden, S.: Query processing in sensor networks. IEEE Pervasive Computing 3(1), 46–55 (2004)
10. Coore, D.: Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer. PhD thesis, MIT Department of Electrical Engineering and Computer Science (1999)
11. Nagpal, R.: Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics. PhD thesis, MIT Department of Electrical Engineering and Computer Science (2001)
12. Bachrach, J., Taylor, C.: Localization in sensor networks. In: Handbook of Sensor Networks, pp. 277–310. John Wiley & Sons, Inc. (2005)
13. Allen, M., Baydere, S., Gaura, E., Kucuk, G.: Evaluation of localization algorithms. In: Mao, G., Fidan, B. (eds.) Localization Algorithms and Strategies for Wireless Sensor Networks. IGI Global (2009)
14. Savarese, C., Rabaey, J.M., Langendoen, K.: Robust positioning algorithms for distributed Ad-Hoc wireless sensor networks. In: Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference, pp. 317–327. USENIX Association (2002)
15. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003)
16. Coore, D.: Establishing a coordinate system on an amorphous computer. MIT/LCS/TR MIT/LCS/TR-737 (1998)
17. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low-cost outdoor localization for very small devices. IEEE Personal Communications 7(5), 28–34 (2000)
18. Bulusu, N., Bychkovskiy, V., Estrin, D., Heidemann, J.: Scalable, ad hoc deployable, rf-based localization. In: Proceedings of the Grace Hopper Celebration of Women in Computing. Institute for Women and Technology (2002)
19. Simic, S., Sastry, S.S.: Distributed localization in wireless ad hoc networks. Technical Report UCB/ERL M02/26, EECS Department, University of California, Berkeley (2002)
20. Savvides, A., Han, C.C., Strivastava, M.B.: Dynamic fine-grained localization in Ad-hoc networks of sensors. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), pp. 166–179. ACM (2001)

21. Niculescu, D., Nath, B.: Ad hoc positioning system (APS). In: Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM 2001, vol. 5, pp. 2926–2931. IEEE (2001)
22. Hightower, J., Borriello, G., Want, R.: SpotON: an indoor 3D location sensing technology based on RF signal strength. Techreport 2000-02-02, University of Washington (2000)
23. Rappaport, T.S.: Wireless Communications: Principles and Practice, 2nd edn. Prentice-Hall (2002)
24. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2000), pp. 775–784. IEEE (2000)
25. Werb, J., Lanzl, C.: Designing a positioning system for finding things and people indoors. IEEE Spectrum 35(9), 71–78 (1998)
26. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 32–43. ACM (2000)
27. Niculescu, D., Nath, B.: Dv based positioning in ad hoc networks. Telecommunication Systems 22, 267–280 (2003)
28. Savvides, A., Park, H., Srivastava, M.B.: The n-hop multilateration primitive for node localization problems. Mobile Networks and Applications 8(4), 443–451 (2003)
29. Nagpal, R.: Organizing a global coordinate system from local information on an amorphous computer. MIT A.I. Laboratory (A.I. Memo No. 1666, MIT) (1999)
30. Wong, S.Y., Lim, J.G., Rao, S.V., Seah, W.K.G.: Density-aware hop-count localization (DHL) in wireless sensor networks with variable density. In: Proceedings of 2005 IEEE Wireless Communications and Networking Conference, vol. 3, pp. 1848–1853. IEEE (2005)
31. Liu, Q., Pruteanu, A., Dulman, S.: GDE: a distributed gradient-based algorithm for distance estimation in large-scale networks. In: Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2011, pp. 151–158. ACM Press (2011)
32. Buschmann, C., Hellbrück, H., Fischer, S., Kröller, A., Fekete, S.P.: Radio Propagation-Aware Distance Estimation Based on Neighborhood Comparison. In: Langendoen, K.G., Voigt, T. (eds.) EWSN 2007. LNCS, vol. 4373, pp. 325–340. Springer, Heidelberg (2007)
33. Villafuerte, F.L., Terfloth, K., Schiller, J.: Using network density as a new parameter to estimate distance. In: Proceedings of the 7th International Conference on Networking, ICN 2008, pp. 30–35. IEEE Computer Society (2008)
34. Huang, B., Yu, C., Anderson, B., Mao, G.: Connectivity-Based distance estimation in wireless sensor networks. In: Proceedings of the 2010 Global Telecommunications Conference, pp. 1–5. IEEE Computer Society (2010)

1-D Coordinate Based on Local Information for MAC and Routing Issues in WSNs[⋆]

Alexandre Mouradian and Isabelle Augé-Blum

Université de Lyon, INRIA, INSA Lyon, CITI, F-69621, France
firstname.lastname@insa-lyon.fr

Abstract. New Wireless Sensor Networks (WSNs) applications are emerging with new requirements such as reliability and respect of time constraints. The underlying mechanisms such as MAC and routing must handle such requirements. To meet timing constraint, it is necessary to bound the hop-count between a node and the sink and the time it takes to do a hop. Thus, the end-to-end delay can be bounded and the communications are real-time. Due to the efficiency and scalability of greedy routing in WSNs and the financial cost of GPS chips, Virtual Coordinate Systems (VCSs) for WSNs have been proposed. A category of VCSs is based on the hop-count from the sink, this scheme leads to many nodes having the same coordinate. The main advantage of this system is that the hops number of a packet from a source to the sink is known. Nevertheless, it does not allow to differentiate the nodes with the same hop-count. For reliability purpose we propose to select forwarder nodes depending on how they are connected in the direction of the sink. In order to be able to do so we need a metric that gives information on hop-count, that allows to strongly differentiate nodes and gives information on the connectivity of each node. As this metric is linked to physical organization of the network it can be viewed as a virtual coordinate. In this paper we propose a novel hop-count-based VCS which aims at classifying the nodes having the same hop-count depending on their connectivity and at differentiating nodes in a 2-hop neighborhood. Those properties make the coordinates, which also can be viewed as a local identifier, a very powerful metric which can be used in WSNs mechanisms. We evaluate the performances of our solution theoretically and by simulation.

1 Introduction

In this paper we focus on low convergecast traffic WSNs applications where alarms from source nodes must reach the sink in a bounded time with a given reliability. We can cite for example volcano monitoring [11] and forest fires detection [15] applications.

The WSNs mechanisms such as MAC, routing and data aggregation (before the alarm is forwarded toward the sink) need to have capabilities to handle such

[⋆] This work has been partially founded by French Agence Nationale de la Recherche under contract VERSO 2009-017.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 42–55, 2012.

critical applications. Our approach to the time constraint problem is to bound the hop-count between a node and the sink and the duration of one hop. So the end-to-end delay can be bounded. The bound on the hop duration implies that the MAC mechanism avoids packet collisions. Thus, the nodes have to be strongly differentiated to be able to make a decision on which node accesses the medium at a given time. At routing layer, the length of any path between a source and the sink has to be known and bounded in order to give guaranties on end-to-end delay. In convergecast networks the hop-count-based solutions such as [14] allow this. Nevertheless it does not allow to differentiate the nodes for forwarder selection because many nodes have the same hop-count. If nodes are not differentiated it can lead to packet duplications for example. For reliability purpose, the forwarder selection should be based on node's connectivity and the nodes having more neighbors in proportion with smaller hop-count should be preferred. In data aggregation context, a node that gathers the data is needed. The choice of this node must be deterministic to avoid unbounded delays. A metric or coordinate is needed for these mechanisms to be able to make such decisions.

In this paper we propose a 1-D coordinate which aims at responding to the aforementioned requirements. The key ideas of our proposition are to classify the nodes having the same hop-count and strongly differentiate them in a 2-hop neighborhood. We do not need to differentiate the nodes in the whole network because MAC and routing mechanisms are usually localized at a 2-hop neighborhood level. At MAC level it is due to the hidden terminal problem and at routing layer a node must choose a forwarder belonging to its neighbors. Our proposition uses only local information in order to build the coordinate thus it is scalable.

In Section 2, an overview of the advantages and drawbacks of existing solutions for WSNs is presented. In Section 3, the theoretical reflexion followed to construct the coordinate is explained and possible issues it can induce are discussed. A theoretical analysis of the coordinate is done in section 4. In section 5, a practical solution to construct the coordinates is given. In Section 6, simulations results are presented and the performances of the algorithm and coordinates are discussed. Section 7 concludes on the presented work and lists future works.

2 Related Work

In the last years, many VCSs have been proposed. This can be explained by the fact that greedy routing has been proven to be very efficient in WSNs mainly because of its stateless characteristic. The first propositions [7] [2] were based on geographic coordinates. The issue with these solutions is the high financial cost of a GPS chip which has to be integrated in every node of the WSN. Moreover, the lack of accuracy in the position of the nodes can induce bad performances of greedy routing [12]. These problems led to solutions based on virtual coordinates because the exact location of all the nodes is not necessary. A VCS can be Cartesian [10] or based on anchors [4] [3]. In the first case, the virtual coordinates of the nodes are given in the same space as the real ones. In the last case, the coordinates are given in distance from anchor nodes (thus if there are n anchors the node is placed in a n-dimension space). A special case of the last type is the 1-dimension system based on the hop-count from the sink. It is used in convergecast networks [14].

A solution based on a Cartesian system is proposed in [10]. First, perimeter nodes are identified and are being given coordinates. Then, each node iteratively updates its coordinates with the center of gravity of its neighbors' coordinates. Nodes others than perimeter ones are initially placed at the center of the area and move toward the borders of the network. [13] improves this scheme by constructing the coordinates during the runtime. Moreover, it does not need to detect perimeter nodes and it considers the sink as the center of the coordinates system. With those systems it is difficult to know the routing path length and connectivity information cannot be deduced from the node's coordinate.

Anchors-based VCSs were proposed in [4] and [3]. Anchors nodes broadcast messages which contain a counter incremented at each hop. For example, in a case where there are three anchors, by listening to these messages a node can determine its virtual coordinates $(V1, V2, V3)$ where $V1$ (resp. $V2$ and $V3$) is the minimum number of hops from anchor 1 (resp. 2 and 3) to the given node. As we are interested in convergecast networks, we focus more on 1-D anchor systems with the sink being the anchor [8] [14].

VPCS (Virtual Polar Coordinates System) [8] can be classified in 1-D anchor systems because each node has a coordinate corresponding to its number of hops from the sink. But, another coordinate corresponding to an angle range is added. A tree representing the network connectivity is built with the sink node as its root. Each node has an angle range and divides it between its sons. This scheme has the advantages of giving the information about hop-count and of differentiating the nodes with the angle parameter. Nevertheless, this last parameter is not physically meaningful because two contiguous angles may be attributed to two different nodes which are not neighbors. The solution is centralized thus not scalable (the sink must know the whole topology). Moreover, a change in the topology induces a reconstruction of a part of the tree which can be costly in energy.

In [14], the authors propose GRAB which uses the hop-count as a cost-field. This cost-field can be seen as a 1-dimension VCS, it represents the cost for a node to reach the sink. Each node is assigned its distance to the sink, in number of hops, as a coordinate. Packets are routed using gradient-routing which consists in choosing the link with the highest gradient, the gradient being defined by the difference between the cost-fields of two nodes. As many nodes with the same hop-count can hear the packet, the selection of the forwarder can be based on a random value, and multiple forwarders can be elected, creating multiple paths. The advantages of such a solution are, that the number of hops to reach the sink is known, and multiple path leads to more reliability. Nevertheless GRAB does not give information on the physical organization of nodes having the same hop-count. SGF [6] and LQER [5] propose similar schemes. In SGF only one forwarder is chosen. LQER adds information on the link quality. Both solutions suffer from the same drawbacks of GRAB.

Among the VCSs proposed in the literature, none can give information on the cost in hop numbers from any given node to the sink and strongly differentiate the nodes in a 2-hop neighborhood at the same time, with the differentiation depending on the connectivity of the node. For these reasons we propose a new VCS which provides those properties. It facilitates the development of many new mechanisms for WSNs.

3 Problem Statement and Proposition

3.1 Problem Statement

In order to be as general as possible we assume that the sensor nodes have a limited amount of energy. The radio is half duplex and mono-channel and the nodes have no information on their geographic position. In this context the aim of our solution is to provide a 1-D coordinate that should give information about the physical position in term of hop-count of a node from the sink and classify the nodes having the same hop-counts. The coordinate should also allow to strongly differentiate nodes in a 2-hop neighborhood. Our solution should be scalable and energy-aware in order to be deployable in WSNs.

3.2 Methodology

The production of the coordinates is separated into two steps, the first is data generation from a theoretical model. The second step is the mapping of the theoretical data on the network in order to give coordinates to the nodes. We detail those two steps in the next paragraphs.

Key Ideas. Our system is composed of only one coordinate that is calculated in function of the number of hops to the sink and an offset that is computed from theoretical data. In the theoretical calculations, we suppose that radio links are perfect (Unit Disk Graph model) and that nodes' repartition is dense and homogeneous, these hypothesis are relaxed during the performances evaluation. So in a hop-count-based VCS, nodes having the same hop-count form rings centered on the sink. The aim of this coordinate is to give information on the hops number and to classify the nodes within a given ring and in a 2-hop neighborhood. The key idea is to have a coordinate which strongly differentiates nodes in a 2-hop neighborhood and which has a physical meaning in the ring. Nodes with proportionally more neighbors in the lower ring should be classified before ones having proportionally less neighbors in the lower ring (the lower rings being the ones nearest to the sink). Classification is done in function of the connectivity of the nodes with the different rings.

Theoretical Model. Our reflexion is based on the Unit Disk Graph (UDG) model, this hypothesis is relaxed during the performances evaluation. The coordinate is constructed by using the information on the number of hops from the sink (noted n) and an offset in a ring as depicted on Fig.1, R being the radio range. The formula used to compute the coordinate of the node p is:

$$coord_p = (n-1) * R + offset \text{ with } offset < R$$

The offset is used to classify the nodes within a ring. We assume that each node knows its hop-count and the number of neighbors it has at each ring (in $n-1$, n and $n+1$, n being the ring of the considered node). The algorithm used in order to obtain this information is described further. The node then computes the percentage of neighbors it has at each ring and uses this information to compute the offset. The idea is to find a mapping between these percentages of neighbors at each ring and the offset of the node in the ring. This is achieved by producing theoretical data where the percentages of neighbors are replaced by percentages of areas of the theoretical range of the considered node in each theoretical ring as shown in Fig.1 (percentages of areas A, B and C corresponding respectively to percentages of neighbors in ring $n-1$, n and $n+1$). We insist on the fact that those areas are theoretical because in reality the range of a node may not be a perfect disk and the rings may not be perfect. Nevertheless, this theoretical data can actually be used to compute an offset.

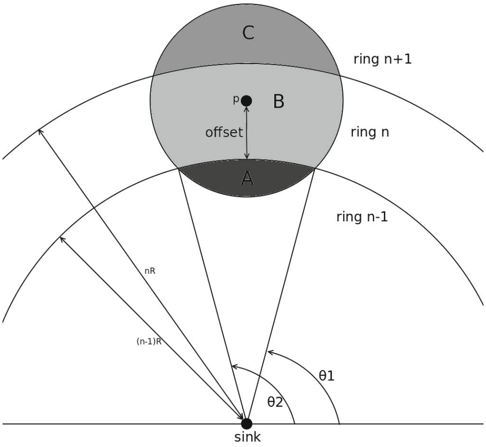


Fig. 1. Theoretical model

Fig.1 shows that the offset parameter is directly linked to the values of A, B and C areas so we can find functions of the type $f(offset) = A$, $g(offset) = B$ and $h(offset) = C$. This is done by calculating one area in function of the position of the node in the ring n. For example, the area A is given by the following integral:

$$A = \int_{\theta_1}^{\theta_2} \int_{r(\theta)}^{(n-1)R} r dr d\theta$$

with

$$\begin{aligned} r(\theta) = {} & [(n-1)R + offset] sin\theta \\ & - \sqrt{R^2 - [(n-1)R + offset]^2 cos\theta^2} \end{aligned}$$

let A_1 be

$$A_1 = \int_{r(\theta)}^{(n-1)R} r dr = \left[\frac{r^2}{2}\right]_{r(\theta)}^{(n-1)R}$$

so

$$A = \int_{\theta_1}^{\theta_2} A_1 d\theta$$

$$\begin{aligned} A = {} & \frac{(n^2-2n)R^2}{2} [\theta]_{\theta_1}^{\theta_2} + \frac{[(n-1)R + offset]^2}{2} \\ & \times [\sin\theta\cos\theta]_{\theta_1}^{\theta_2} + \frac{[(n-1)R + offset]}{2} \\ & \times \Bigg[- \cos\theta \sqrt{R^2 - [(n-1)R + offset]^2 \cos^2\theta} \\ & - \frac{R^2 \tan^{-1}(\frac{[(n-1)R+offset]\cos\theta}{\sqrt{R^2-[(n-1)R+offset]^2\cos^2\theta}})}{(n-1)R + offset} \Bigg]_{\theta_1}^{\theta_2} \end{aligned}$$

$$\text{with} \begin{cases} \theta_1 = \arctan(\frac{(n-1)^2R^2 - R^2 + [((n-1)R+offset)]^2}{2[((n-1)R+offset)]} \\ \times \frac{+1}{\sqrt{(n-1)^2R^2 - [\frac{(n-1)^2R^2-R^2+[((n-1)R+offset)]^2}{2[((n-1)R+offset)]}]^2}}) \\ \theta_2 = \arctan(\frac{(n-1)^2R^2 - R^2 + [((n-1)R+offset)]^2}{2[((n-1)R+offset)]} \\ \times \frac{-1}{\sqrt{(n-1)^2R^2 - [\frac{(n-1)^2R^2-R^2+[((n-1)R+offset)]^2}{2[((n-1)R+offset)]}]^2}}) + \pi \end{cases}$$

On the same principle we can compute C, B is given by $\pi R^2 - A - C$. We see that for a given offset we obtain values of A, B and C so by dividing these areas by the area of the theoretical range we deduce the offset in function of the percentages of areas A, B and C. The principle is then to map the percentages of neighbors on the percentages of areas and thus being able to give an offset to each node.

We can notice that two nodes in the same 2-hop neighborhood having the same percentages of neighbors at each ring are given the same coordinate. From now we refer to this situation as a coordinate collision. The theoretical analysis and the evaluation sections show that, even if this situation can occur, it is actually very rare.

Mapping Issues. At this point we have a function that links the percentages of areas with the offset. The aim is then to give a coordinate to each node. A node knows the percentages of neighbors it has at rings $n-1$, n and $n+1$ (noted $\%(n-1)$, $\%n$ and $\%(n+1)$). So we have to link those percentages of neighbors with the area percentages. This is done with a projection of the neighbors percentages values on the areas percentages.

In reality a node can have percentages of neighbors that do not fit the theoretical values, for instance if a node does not have any neighbors in its own ring (Fig.1 shows that area B is never null with $0 \leq offset < R$). This implies that the space of neighbor percentages values is larger than the area percentages one. Fig.2 represents the plane of neighbors percentages, the plane corresponds to $\%(n-1) + \%n + \%(n+1) = 1$ and the curve which links $\%A$, $\%B$ and $\%C$ is contained in the plane (because $\%A + \%B + \%C = 1$). The projection of the values of the plane on the curve leads to nodes which have different neighbors percentages being given the same areas percentages and thus the same offset as pictured in Fig.2 (with points p and q). This issue is mitigated by the addition of the euclidean distance of the projection (noted d) to the offset so:

$$offset' = offset + d \text{ and } coord = (n-1) * R + offset'$$

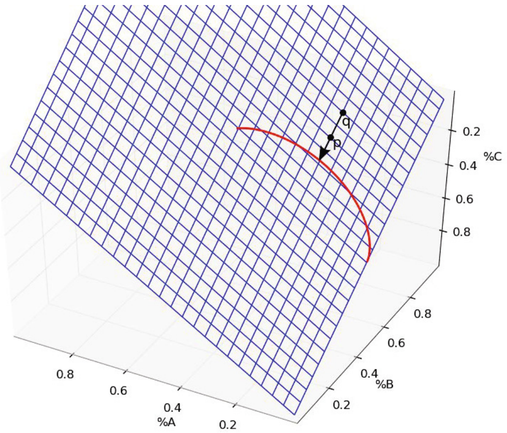


Fig. 2. Curve that links %A, %B and %C values

Table 1. Notations used in Theorem 1 proof

Symbol	Signification
p_1 and p_2	Points in the neighbors percentages space
$offset_1$ and $offset_2$	Respectively the offset of p_1 and p_2 after the projection but before the addition of the projection distance.
d_1 and d_2	Respectively the projection distances of p_1 and p_2
$\Delta offset$	Distance between two consecutive values in the discrete offset space.

Theorem 1. *The addition of the projection distance resolves some collisions without adding more if the offset values space is discrete and theoretical consecutive offset values are separated by at least the maximum projection distance ($\Delta offset \geq d$).*

Proof. We do a proof by contradiction, the notations are detailed in Table 1. Let's suppose the addition of the projection distance creates a collision. It means that two points (p_1 and p_2), which do not get the same $offset$ ($offset_1$ and $offset_2$ with $offset_1 < offset_2$) end with the same $offset'$ because of the addition of the projection distances. It is possible if we have $d_1 = offset_2 + d_2 - offset_1$. We know that $offset_2 - offset_1 \geq \Delta offset$, because $\Delta offset$ is the distance separating two consecutive offset values, so $offset_2 - offset_1 + d_2 > \Delta offset$ and we conclude that $d_1 > \Delta offset$ which is a contradiction.

In practice a WSN's node embeds a table that contains discrete values of the curve $f(\%A, \%B, \%C) = offset$. The point calculated with the percentages of neighbors is projected on the nearest point in the theoretical data table, the corresponding offset is given to the node and the projection distance is then added to the offset. This technique allows to have a relatively low granularity of the theoretical data because the addition of the distance prevents collisions. This property is interesting because the WSNs' nodes have generally a low memory.

In theory, collisions can occur in two cases: either because nodes in the same 2-hop neighborhood have the same percentage of neighbors in each ring or because the projection distance is the same. In practice the calculations of the percentages and the projection distances are done with a finite precision, that induces more collision. We analyze those issues in the next sections.

4 Theoretical Analysis of the Solution

In this section we analyze the theoretical probability of obtaining coordinate collisions. To do so, we characterize the coordinate space in the neighborhood of a node and compute the expected number of pairs of nodes which have the same coordinate and are in the neighborhood of one node. We assume that the nodes are distributed randomly on a plane, that a node in ring n always has at least one neighbor in ring $n-1$ ($\%(n-1) > 0$) and that the coordinate is chosen randomly by using the uniform distribution. This last statement implies that we assume that the positions of the neighbors of the nodes being in the neighborhood of the same node are independent. Intuitively it is not the case because two nodes that are very close in distance tend to have very similar neighborhoods. So this assumption leads to more probabilities that nodes can differentiate themselves. Also, we do not take into account collisions due to the projection in this part, it means that only the nodes having the same proportions of neighbors are in a coordinate collision. These hypothesis imply that we are taking into account less collisions in the theoretical analysis than the ones that can occur in reality. Nevertheless this analysis is useful to evaluate the quality of our proposition.

4.1 Coordinate Space Characterization

First, we characterize the coordinate space. The nodes can differentiate them with their proportions of neighbors in each ring. We consider a node with k neighbors which all have also k neighbors. Each node has a given proportion of neighbors in rings $n-1$, n and $n+1$. The combinations of proportions in each ring represent the coordinate space. The number of accessible proportions depends on the number of neighbors. If a node has 2 neighbors it can have 0% or 50% of them in ring n (100% is impossible because it always has at least one neighbor in ring $n-1$), if it has 3 neighbors it can have 0%, 33% or 66% in ring n. Thus the cardinal of coordinate space (noted N) increases with the number of neighbors. We note that the possibilities of having different proportions in a ring depend on the proportions in the other rings. For example, if a node has 3 neighbors and it has 33% of them in ring $n-1$, it can have 33% in ring n and 33% in $n+1$, or 66% in n and 0% in $n+1$, or 0% in n and 66% in $n+1$ leaving no other possibilities. So we have $\%(n-1)+\%n+\%(n+1) = 1$ which can be written

$$m\frac{1}{k} + o\frac{1}{k} + p\frac{1}{k} = 1$$
$$m + o + p = k$$

with k the number of nodes in a neighborhood, $m \in [1, k]$ and o, $p \in [0, k]$ respectively the numbers of neighbors at $n-1$, n and $n+1$. If m is fixed we have $k - m = o + p$ so

$$\left.\begin{array}{ll} & o = (k-m) \text{ and } p = 0 \\ \text{or} & o = (k-m-1) \text{ and } p = 1 \\ \ldots & \\ \text{or} & o = 0 \text{ and } p = (k-m) \end{array}\right\} k - m + 1 \text{ possibilities}$$

We sum the number of possibilities for each value of m:

$$\sum_{m=1}^{k} k - m + 1 = k + (k-1) + ... + 1 = \frac{k(k+1)}{2}$$

Thus the cardinal of coordinate space (noted N) is

$$N = \frac{k(k+1)}{2}$$

This argument holds if we fix o or p first.

4.2 Expected Number of Coordinate Collisions

In this section we compute the expected number of collisions in function of the number of neighbors of the nodes in the network (noted k). In a neighborhood of k nodes, the probability that a given node i has the same coordinate of a node j is $\frac{1}{N}$ if we assume that we pick the coordinate following a uniform distribution. Let X_{ij} be a random variable such that $X_{ij} = 1$ if there is a collision between i and j and $X_{ij} = 0$ otherwise, thus $X = \sum_{i \neq j} X_{ij}$ is a random variable that represents the number of 2-collisions seen by a node which has k neighbors. We have $E[X] = \sum_{i \neq j} E[X_{ij}]$ with $E[X]$ being the expected number of collisions.

$$E[X] = \binom{k}{2} \frac{1}{N} = \frac{k!}{2!(k-2)!} \frac{2}{k(k+1)} = \frac{k-1}{k+1}$$

because $N = k(k+1)/2$. We also can notice that

$$\lim_{k \to +\infty} \frac{k-1}{k+1} = 1$$

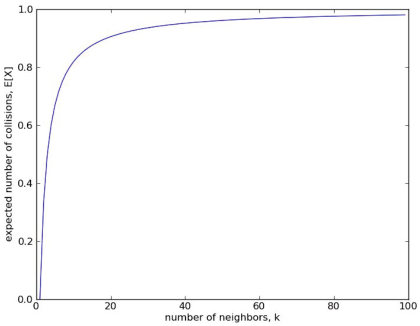


Fig. 3. Expected number of collisions in function of the size of a neighborhood

Fig.3 represents the plot of the expected number of collisions $E[X]$ in function of the number of neighbors k. The curve is always under the value 1 which means that the expected 2-collisions number is bounded by 1. The expected number of coordinate collisions in a 2-hop neighborhood does not depend on the average degree of the network. Nevertheless, in reality there are more collisions on average, as it is described in section 6. This is due to the collisions induced by the projection as mentioned in previous section, the fact that the repartitions of the neighbors of nodes that are neighbors are not independent and also because of the use of finite precision number in the implementation which is described in section 6. The result is still interesting because it shows that the number of collisions should be stable whatever the density of the network is.

5 Practical Construction of the Coordinate

In this section we focus on how the nodes can gather information about their hop-counts and the percentages of neighbors they actually have in the different rings.

5.1 Coordinates Initialization Algorithm

The nodes use a duty-cycle [9] mechanism. They alternately wake up and go into sleep state. This mechanism reduces the amount of energy consumed during the initialization of the coordinates.

During the initialization a node obtains information about in which ring it is and the number of neighbors it has in the different rings. The algorithm describes the exchanges of information among the nodes at MAC level. There are two versions of the algorithm, one synchronous were the nodes know when their neighbors wake up and another asynchronous in which they have no information on wakeup dates. Here we will describe only the asynchronous algorithm since the synchronous is the same without the part which synchronizes the nodes (because, in this case, they are assumed to be synchronized by another mechanism).

The sink begins the algorithm, it starts the initialization process which then progresses from the sink toward the border of the network. The algorithm is described for a node at ring n. The nodes are synchronized with a long preamble [9]. Nodes at $n-1$ ring send a preamble used to synchronize nodes at rings $n-1$ and n. Then there is a slotted contention period where the nodes at ring $n-1$ chose randomly a slot (using a uniform distribution). They send a packet containing their ring number. It allows the nodes at ring n to know in which ring they are (the ring number they receive in the messages plus one) and by counting the number of packets received they deduce the number of neighbors they have at ring $n-1$. The process is repeated with nodes at layer n and $n+1$ thus at the end of three contention periods a node knows its ring number and the number of neighbors it has at ring $n-1$, n and $n+1$.

5.2 Energy Consumption

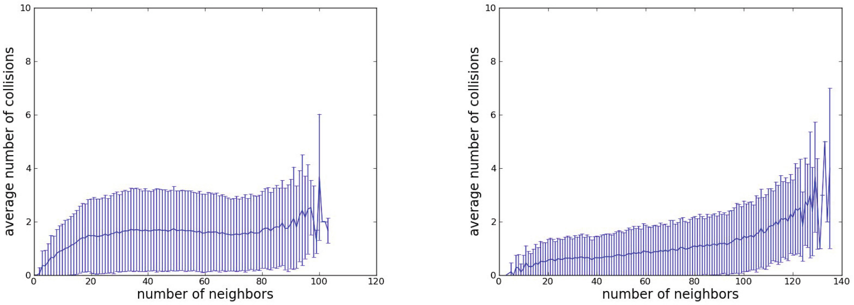
The nodes have to listen to three contentions periods in which they receive packets from their neighbors at the different layers. They send a packet only once. Thus the energy consumed during the initialization is the energy needed to listen during three contention periods and to send one packet plus the energy used for the synchronization. The use of a global synchronization or of a long preamble (respectively synchronous or asynchronous version) depends on the application. If global synchronization is needed by the mechanisms which use the coordinate, it could also be used for the construction of the coordinate.

5.3 Coordinate Update Discussion

WSNs have a dynamic nature: the neighborhood of a node can change due to node deaths or due to changes in the radio environment. In this case the coordinates lose their relevance. The coordinates have thus to be updated with a frequency which depends on the network's dynamic. The update can be out-band i.e., the initialization algorithm is being run periodically or it can be in-band, i.e., the update is integrated to the MAC or routing scheme (these protocols generally need information on the neighborhood so updates can be done on the fly). The in-band solution should be preferred in our opinion, because it induces less overhead and thus less energy consumption.

6 Performances of the Coordinates

Simulations are performed with the discrete events simulator for WSNs, WSNet [1]. We simulate a network of dimensions 50x50 square units with the sink at (25,25) the communication range is 10 (we chose a relatively small simulation area because it limits the simulations duration : we can have a high increase of the network average degree with a relatively low increase of the number of nodes). We simulate with two different propagation models, the free space propagation model which corresponds only to the path loss without shadowing or fading (but there are still packet collisions), this allows to test our algorithm with a perfect channel. The second is the log-normal shadowing model which has been proven [16] to be very suited to model real wireless links in the case of WSNs. We simulate the initialization protocol previously described with 50 to 750 nodes placed randomly. It represents from 3 to 5 hops depending on the topologies. We simulate the asynchronous version of the initialization protocol.



(a) Free space model (b) Log-normal shadowing model

Fig. 4. Average number of collision seen by a node with 95% confidence interval in function of network density

Our goal is to study the number of coordinate collisions induced by the method used to construct the coordinate. Nevertheless, in the simulator the coordinate is represented by floating point numbers with finite precision which can induce collisions. Although collisions which do not come from our construction method appears, they have to be taken into account because real life implementations will also use finite precision numbers to store the coordinate. Here we study the impact of the network density on the number of coordinate collisions seen by a node.

6.1 Perfect Radio Links

Fig.4(a) represents the average number of collisions seen by a node for a given number of neighbors, for a given node we count the number of pairs of its neighbors having the same coordinate (i.e. the number of collisions it sees). This number does not depend on the network density in the case of free space propagation model. It confirms the theoretical results of section 4 with the average number of collisions being higher than the expected number in the analysis. This is due to our hypothesis in section 4 and the previously cited sources of coordinate collisions (projection, finite precision, etc) that we do not take into account in the theoretical analysis. From this observation we can tell that our solution better classifies nodes in dense networks because a node sees less collisions in proportion. The mean coordinate collisions number is near 2 which means that on average a node has 2 pairs of neighbor nodes that have the same coordinate. Thus for a node with 10 neighbors it is 40% of its neighbors and for a node with 100 it is 4%. The curves for highest densities are not very representative because there are few nodes with above 90 neighbors in our simulations, this explains the end of the curve.

6.2 Unreliable Radio Links

Fig.4(b) shows that in the case of log-normal shadowing propagation model the average of collision number seen by a node is slightly less than in the case of free space propagation model with almost the same 95% confidence interval and it does not significantly grows with the network density.

As stated previously those collisions are an issue because we want to use the coordinates to discriminate nodes in a 2-hop neighborhood. On the other hand there are few collisions (we see that at least 95% of the number of collisions for any number of neighbors between 20 and 90 is below 3 with both propagation models). The solution we propose can be used on real radio chips because performances on unreliable radio links are similar to those with perfect channel.

7 Conclusion and Future Works

In this paper, we propose a new VCS which allows to address MAC and routing issues in WSNs. We especially focus on mechanisms which need reliability and the respect of time constraints. Our approach of the problem leads us to produce a solution based on the hop-count, which differentiates the nodes in a 2-hop neighborhood while giving an information on their connectivity with the other hop-count rings. The theoretical background is presented and potential issues are discussed, the main issue being coordinate collisions. The theoretical analysis shows that the expected number of collisions converges to a small value when the network density grows. A coordinate construction algorithm is proposed and the theoretical predictions are verified by simulation. We thus conclude that our solution actually provides a good node differentiation in a 2-hop neighborhood. We also conclude that our solution is better for dense WSNs because there are less coordinates collisions in proportion.

Since our final aim is to provide WSNs MAC and routing mechanisms with reliability and the ability to respect time constraints, in the future we will use the characteristics of the coordinate in such mechanisms.

References

1. http://wsnet.gforge.inria.fr/
2. Bose, P., Morin, P., Stojmenović, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. In: DIALM 1999. ACM, USA (1999)
3. Cao, Q., Abdelzaher, T.: A scalable logical coordinates framework for routing in wireless sensor networks. In: RTSS 2004, Lisbon, Portugal (2004)
4. Caruso, A., Chessa, S., De, S., Urpi, R.: Gps free coordinate assignment and routing in wireless sensor networks. In: IEEE INFOCOM, Miami, USA (2005)
5. Chen, J., Lin, R., Li, Y., Sun, Y.: Lqer: A link quality estimation based routing for wireless sensor networks. Sensors 8(2) (2008)
6. Huang, P., Chen, H., Xing, G., Tan, Y.: Sgf: A state-free gradient-based forwarding protocol for wireless sensor networks. ACM Trans. Sen. Netw. 5 (April 2009)
7. Karp, B., Kung, H.T.: Gpsr: greedy perimeter stateless routing for wireless networks. In: MobiCom 2000, Boston, USA (2000)
8. Newsome, J., Song, D.: Gem: Graph embedding for routing and data-centric storage in sensor networks without geographic information. In: SenSys 2003, Los Angeles, USA (2003)
9. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: SenSys 2004, Baltimore, MD, USA (2004)
10. Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., Stoica, I.: Geographic routing without location information. In: MobiCom 2003, San Diego, CA, USA (2003)
11. Tan, R., Xing, G., Chen, J., Song, W.Z., Huang, R.: Quality-driven volcanic earthquake detection using wireless sensor networks. In: RTSS 2010, San Diego, CA, USA (2010)
12. Watteyne, T., Auge-Blum, I., Dohler, M., Barthel, D.: Geographic forwarding in wireless sensor networks with loose position-awareness. In: PIMRC 2007, Athens, Greece (2007)
13. Watteyne, T., Augé-Blum, I., Dohler, M., Ubéda, S., Barthel, D.: Centroid virtual coordinates - a novel near-shortest path routing paradigm. Comput. Netw. 53 (2009)
14. Ye, F., Zhong, G., Lu, S., Zhang, L.: Gradient broadcast: a robust data delivery protocol for large scale sensor networks. Wirel. Netw. 11, 285–298 (2005)
15. Zhang, J., Li, W., Han, N., Kan, J.: Forest fire detection system based on a zigbee wireless sensor network. Journal of Beijing Forestry University 29(4) (2007)
16. Zuniga, M., Krishnamachari, B.: Analyzing the transitional region in low power wireless links. In: SECON, Santa Clara, USA (2004)

Uninterrupted Coverage of a Planar Region with Rotating Directional Antennae

Evangelos Kranakis[1], Fraser MacQuarie[2], Oscar Morales-Ponce[3], and Jorge Urrutia[4]

[1] School of Computer Science, Carleton University, Ottawa, Canada
Supported in Part by NSERC and MITACS Grants
kranakis@scs.carleton.ca

[2] School of Computer Science, Carleton University, Ottawa, Canada
frasermacquarrie@gmail.com

[3] School of Computer Science, Carleton University, Ottawa, Canada
Supported by MITACS Postdoctoral Fellowship
omponce@connect.carleton.ca

[4] Instituto de Matemáticas, Universidad Nacional Autónoma de México, México
Supported in Part by Conacyt
urrutia@matem.unam.mx

Abstract. Assume that n directional antennae located at distinct points in the plane are rotating at constant identical speeds. They all have identical range and sensor angle (or field of view). We propose and study the *Rotating Antennae Coverage Problem*, a new problem concerning rotating sensors for the uninterrupted coverage of a region in the plane. More specifically, what is the initial orientation of the sensors, minimum angle, and range required so that a given (infinite or finite) line or planar domain is covered by the rotating sensors at all times? We give algorithms for determining the initial orientation of the sensors and analyze the resulting angle/range tradeoffs for ensuring continuous coverage of a given region or line in the plane with identical rotating sensors of given transmission angle and range. We also investigate other variants of the problem whereby for a given parameter T (representing time) there is no point in the domain that is left unattended by some sensor for a period of time longer than T. Despite the apparent simplicity of the problem several of the algorithms proposed are intricate and elegant. We have also implemented our algorithms in C++ and the code can be downloaded on the web.

Keywords and Phrases: Angle, Antenna, Constant Speed, Coverage, Floodlights, Rotating, Sensors.

1 Introduction

Assume n directional antennae with identical range and beam width and located at distinct points in a planar finite or infinite domain. The antennae are rotating continuously at constant identical speeds. A point in the domain is called *covered* by a sensor if it is within the range and coverage area of at least one of the n sensors. The domain may well represent a critical region all of whose points need to be covered so as to monitor important events (such as animal migration, military activity, navigation guidance, etc.)

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 56–68, 2012.

which is taking place within this domain. In this setting it is required that specific events that may occur at some point within this domain be detected, located and reported by at least one of the sensors at all times. More specifically we consider the following *Rotating Antennae Coverage Problem* concerning the monitoring of a region.

> Assume we are given a finite or infinite planar region. We have n sensors modelled as directional antennae with given identical ranges and beam widths. The sensors are rotating continuously with constant identical speeds. We are concerned with providing an algorithm for determining the initial orientation of the antennae so as to ensure that no point in the domain is ever left unmonitored at any time. In addition, we are also interested in algorithms for attaining optimal antennae angle/range tradeoffs for accomplishing this monitoring task.

In a further (and natural) generalization, we may also be interested in two additional parameters. 1) *Gap Time T:* for some real number $T \geq 0$, it is required that specific events that may occur at some point in this domain be detected, located and reported by at least one sensor within any specified time interval whose length does not exceed a certain gap T, and 2) *Number of Monitoring Antennae k:* for some integer $k \geq 1$, every point in the region is monitored by at least k antennae at all times. We use the notation $RAC_k(T)$ to denote this Rotating Antennae Coverage problem with monitoring time T and number of monitors k. When $k = 1$ we use the abbreviation $RAC(T)$, when $T = 0$ the abbreviation RAC_k, and when both $k = 1, T = 0$ we simply use the abbreviation RAC. In particular, in RAC_k we want to ensure that every point in the region is always monitored by at least k sensors at all times. Thus, despite the fact that the coverage provided by each individual sensor may be intermittent (due to limitations on the antenna angle and range) and may result in insufficiently covered "corridors" within the plane region during the antenna rotation, the coverage provided by the ensemble of all the rotating sensors when taken together guarantees complete coverage of the region at all times.

To address the Rotating Antennae Coverage problem we propose a rotation model whereby directional antennae rotate at constant identical speeds in the same direction. This same model could also be used if it was required to locate the activities and report events if sensors were also location aware (i.e., they knew their geographic coordinates).

1.1 Preliminaries, Definitions, and Notation

In the sequel we define our coverage problem precisely and provide basic terminology, definitions and notation. Throughout the paper we assume that we have n identical directional sensors. Each sensor consists of a rotating directional antenna with range (also called radius) $r > 0$, beam width (also called angle) $0 \leq \phi \leq 2\pi$ that rotates around its apex (which is at a fixed position) with constant speed in clockwise order. All antennae rotate in the same direction at constant identical speeds. The antennae are set at some initial orientation (determined by an algorithm) that depends on the particular location of its sensor in relation to the remaining sensors in the set of points. The coverage area of a sensor at time t is the circular sector of radius r and angle ϕ determined by the sensor during its rotation at time t. A point in a given planar region $\mathcal{R}$, is called covered at time t if it is within the range of at least one of the n sensors at time t. We study the problem of covering $\mathcal{R}$ with a set of rotating directional sensors of identical angle ϕ

and range r. We distinguish two types of sensors: 1) directional sensors with given angle and finite range, for example, video cameras, and 2) directional sensors with given angle but unlimited (or infinite) range, which we refer to in the sequel as *floodlights*.

Note that although floodlights (i.e., sensors with infinite range) may not be technically realistic, nevertheless they will prove to be quite convenient in subsequent discussions in that they will simplify proofs and mathematical presentation. With these explanations in mind we are ready to give the main definitions.

Definition 1 (Angles $\Phi_r(P,\mathcal{R})$ and $\Phi(P,\mathcal{R})$). *Let P be a set of points in the plane and $\mathcal{R}$ be a planar region. Let $\Phi_r(P,\mathcal{R})$ be the infimum over all angles $\phi \leq 2\pi$ such that if sensors of angle ϕ and range r are located at the points P then there is an initial orientation of the sensors so that the whole region $\mathcal{R}$ is covered at all times under continuous rotation of the directional antennae. For the case of floodlights we have infinite range $r = +\infty$ in which case we use the notation $\Phi(P,\mathcal{R})$.*

Definition 2 (Angles $\Phi_r(n,\mathcal{R})$ and $\Phi(n,\mathcal{R})$). *Let $\mathcal{R}$ be a region in the plane. Let $\Phi_r(n,\mathcal{R})$ be the infimum over all $\Phi_r(P,\mathcal{R})$ where P is any set of n directional sensors in the plane. For the case of floodlights we have infinite range $r = +\infty$ in which case we use the notation $\Phi(n,\mathcal{R})$.*

We note that although in the sequel we will be assuming that the sensors lie in the region $\mathcal{R}$ under consideration the definitions make sense even without this assumption. A similar definition can be given for covering a line $\mathcal{L}$ (i.e., only for points located on the line) using rotating antennae and the corresponding notation is $\Phi(n,\mathcal{L})$. The coverage problems we are interested in can be formulated precisely as follows.

Problem 1. Determine the beam width $\Phi_r(P,\mathcal{R})$ such that there is an initial orientation of the sensors in P with range r so that the whole region $\mathcal{R}$ is covered under continuous rotation of the directional sensors. Similar problem for $\Phi(P,\mathcal{L})$.

Problem 2. Determine the beam width $\Phi_r(n,\mathcal{R})$ such that there is an initial orientation of n directional sensors with range r so that the whole region $\mathcal{R}$ is covered under continuous rotation of the directional sensors. Similar problem for $\Phi(n,\mathcal{L})$.

We will see in the sequel that the coverage problems for infinite and finite range are related. As usual, $\angle(ABC)$ denotes the angle between the line segments AB and BC. Assume we have a point $K = (x,y)$. For any angle ρ define the point $K_\rho = (x,y) + re^{i\rho}$.

Definition 3 (Rotating Antenna Sector). *Consider a directional antenna located at a point K with beam width ϕ and radius r as it rotates clockwise. We define as follows the sector delimited by the antenna at time t.*

-Let $F_K(r,\rho;0)$ denote the initial sector defined by the sensor when its orientation is ρ; this is the circular sector defined in a circle of radius r, centered at K and delimited by the radii KK_ρ and $KK_{\rho+\phi}$.

-At time t the sensor will rotate by an angle of t radians. Let $F_K(r,\rho;t)$ denote the circular sector at time t which is defined in a circle of radius r, centered at K and delimited by the radii $KK_{\rho-t}$ and $KK_{\rho-t+\phi}$.

Although we omit the details, a similar definition can be given when r is infinite and we simply denote the sector defined by the sensor located at the point K by $F_K(\rho;t)$. Observe that the orientation at time t is invariant to the initial orientation, i.e., $F_K(r,\rho;t) = F_K(r,\rho - t;0)$.

1.2 Related Work

There exists research in computational geometry that is somewhat related to our problem. For example, the art gallery problem which is concerned with placing the minimum number of guards in a planar domain so as to cover a given region or perimeter and has been studied in various different settings. For the art gallery problem, Chvatal [2] proved that $n/3$ guards are always sufficient and sometimes necessary to guard a simple polygon with n vertices and later Fisk [3] gave a shorter proof. In these works, guards have an omnidirectional field of view. For additional details on art gallery problems the reader is referred to [7,9], as well as to [4] for a more recent randomized algorithm for sensor placement in a simple polygon. Closely related is research with floodlights which corresponds to our antenna model with fixed angle but infinite range. For example, [11] proposes the problem of illuminating the plane with floodlights and proves that the infinite plane can be illuminated with n floodlights if and only if the sum of angles is at least 2π.

There is extensive literature in mobile and sensor networks concerning coverage, e.g., see [10,1]. The k–coverage problem with isotropic sensors was studied in [6]. In [12] and [5] the authors studied the k-coverage problem and the relationship between coverage and connectivity. Additional research can also be found in [8].

It is important to point out that all the literature mentioned above differs from our setting in that the antennae are static while we are concerned with a dynamic model of rotating antennae.

1.3 Results of the Paper

We provide several algorithms depending on the number of points and their relative location that determine for a given set of points in the plane the initial orientation of the sensors, as well as minimum angle, and range required so that a given (infinite or finite) line or planar domain is covered at all times regardless of the fact that the sensors are rotating. We give algorithms for determining the initial orientation of the sensors and study angle/range tradeoffs given that the sensors rotate with identical speeds and have a given field of view and range. Section 2 is concerned with lattice configurations, and Section 3 with arbitrary configurations of points in the plane. In both cases we consider algorithms for orienting the antennae so as to cover a given line or region provided the sensors are located in lattice configurations or arbitrary positions in the plane. In Section 4 we look at other variants of the problem for a given parameter T (representing the gap time) whereby no point in the domain is left unattended by a sensor for a period longer than T. Several of the algorithms proposed are intricate and elegant. We conclude with discussion of open problems. The main results of the paper are summarized in Table 1, for infinite, and Table 2 for finite antennae, respectively.

Table 1. Summary of results with finite range. We use the notation $r_{DT(P)} = 2\max_{u,v}(d(u,v) : \{u,v\} \in DT(P))$, where $DT(P)$ is the Delaunay Triangulation of the set of points.

Points P in	Range	Beam Width
Line $\mathcal{L}$ of size r	r	$\Phi_r(P,\mathcal{L}) = \frac{3\pi}{n}$
Lattice $\mathcal{L}$ of size $m \times n$	$r \leq 2\max(n,m)/3$	$\Phi_r(P,\mathcal{L}) \geq \frac{2\pi}{r}$
Lattice $\mathcal{L}$ of size $m \times n$	$r \leq 2\max(n,m)/3$	$\Phi_r(P,CH(\mathcal{L})) \geq \frac{2\pi}{\sqrt{r^2-1}}$
General Position	$r_{DT(P)}$	$\Phi_r(P,CH(P)) \geq \pi$

Table 2. Summary of results with infinite range. We use the notation $\mathcal{H}_u(\mathcal{L})$ to denote the upper half-plane determined by $\mathcal{L}$.

Points P in	Coverage Region	Beam Width
Line $\mathcal{L}$	$\mathcal{L}$	$\Phi(P,\mathcal{L}) = \frac{3\pi}{n}$
Line $\mathcal{L}$	$\mathcal{H}_u(\mathcal{L})$	$\Phi(P,\mathcal{H}_u(\mathcal{L})) = \frac{3\pi}{n}$
General Position	Plane $\mathcal{P}$	$\Phi(2,\mathcal{P}) = 2\pi$
General Position	Plane $\mathcal{P}$	$\Phi(3,\mathcal{P}) = \pi$

2 Lattice Configurations

In this section we consider sensors located in lattice positions, namely the $1 \times n$ and $m \times n$ grid.

2.1 Infinite Line

Theorem 1. *For any set P of $n \geq 2$ floodlights on a line $\mathcal{L}$ we have that $\Phi(P,\mathcal{L}) = \frac{3\pi}{n}$.*

Proof. Without loss of generality assume that the line $\mathcal{L}$ to be covered is horizontal. Let $P = \{p_0, p_1, ..., p_{n-1}\}$ be the set of n sensors on the line $\mathcal{L}$ and let the points be such that the x-coordinate of p_i is less than the x-coordinate of p_{i+1}, for $i = 0,1,\ldots,n-2$.

First we prove that an angle of $\frac{3\pi}{n}$ is always sufficient. Let the initial orientation of the sensor at p_i be $F_{p_i}(i \cdot 3\pi/n;0)$, for $i = 0,1,\ldots,n-1$; see Figure 1. We define a dual plane as follows: each sensor i is the circular sector of a unitary circle C delimited by $i \cdot 3\pi/n$ and $(i+1) \cdot 3\pi/n$, and at time t, the line $\mathcal{L}$ is represented as a directed line segment $\overrightarrow{\mathcal{L}}$ such that $\overrightarrow{\mathcal{L}}$ crosses the center of C and the head of $\overrightarrow{\mathcal{L}}$ forms an angle t with the horizontal; see Figure 2a.

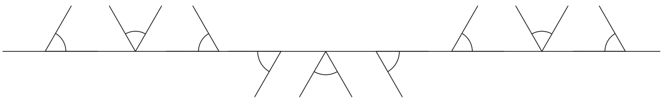
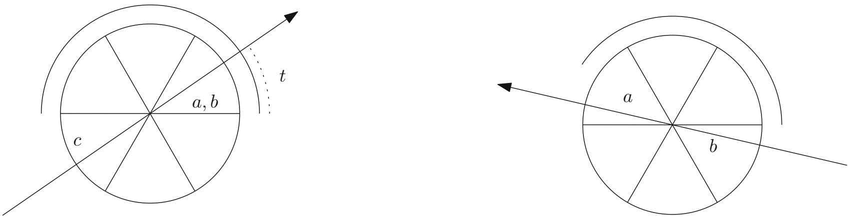


Fig. 1. Initial orientation of the directional sensors on $\mathcal{L}$



(a) Orientation at t.

(b) If $\Phi(n,\mathcal{L}) < \frac{3\pi}{n}$, $\mathcal{L}$ is not always fully covered.

Fig. 2. Directional sensors at a unique point

In the dual plane, sensors are static while it is the line $\mathcal{L}$ that rotates all the time. The orientation $\overrightarrow{\mathcal{L}}$ of $\mathcal{L}$ preserves the sensor rotations in the original plane. The head of $\overrightarrow{\mathcal{L}}$ represents ∞ and the tail represents $-\infty$ in the original plane.

Since the sum of the angles is 3π, the circular sector $[0,\pi)$ of C in the dual plane is always covered by two sets $S_1, S_2 \subseteq P$ of sensors while the circular sector $[\pi, 2\pi)$ of C in the dual plane is covered by one set $S_3 \subseteq P$ of sensors. Observe that each sensor in S_3 is between S_1 and S_2 in the original plane. Let $a \in S_1$, $b \in S_2$ and $c \in S_3$ be the sensors that cover a segment of $\overrightarrow{\mathcal{L}}$ at time t in the dual plane. If a and b cover the head of $\overrightarrow{\mathcal{L}}$, c covers the tail. Therefore, $\mathcal{L}$ is fully covered by c and b in the original plane. Similarly, if a and b cover the tail of $\overrightarrow{\mathcal{L}}$, c covers the head. Therefore, $\mathcal{L}$ is fully covered by a and c in the original plane.

Now we prove that an angle of $\frac{3\pi}{n}$ is always necessary. Assume on the contrary that the sum of angles is less than 3π. Therefore, there exists a time t when only two sensors, say a and b, cover a segment of $\overrightarrow{\mathcal{L}}$ in the dual plane as depicted in Figure 2b. Assume a covers the tail and b covers the head of $\overrightarrow{\mathcal{L}}$ in the dual plane. Therefore, $\mathcal{L}$ is fully covered in the original plane. However, at time $t+\pi$, a covers the head and b covers the tail of $\overrightarrow{\mathcal{L}}$ in the dual plane. Therefore, the line segment ab of $\mathcal{L}$ in the original plane is not covered. This contradicts the assumption. The pseudocode is presented in Algorithm 1.

Algorithm 1. Initial orientation of sensors on a line $\mathcal{L}$ that covers $\mathcal{L}$.

```
input : $\{p_0, p_1, \ldots, p_{n-1}\}$ : sensors on the horizontal line
output: Initial orientation of $\{p_0, p_1, \ldots, p_{n-1}\}$
Let the $x$-coordinate of $p_i$ be less than the $x$-coordinate of $p_{i+1}$;
for $i \leftarrow 0$ to $n-1$ do
    Orient the antenna at $p_i$ as $F_{p_i}(i \cdot 3\pi/n; 0)$;
```

This completes the proof of the theorem. □

Observe that if $\mathcal{L}$ is finite, then it is sufficient to use a range equal to the length of $\mathcal{L}$. Thus, we have the following corollary to Theorem 1 when $\mathcal{L}$ is finite.

Corollary 1. *For a set P of $n \geq 2$ sensors on a line $\mathcal{L}$ of length r, we have that* $\Phi_r(P, \mathcal{L}) = \frac{3\pi}{n}$.

2.2 Rectangular Lattice

Theorem 2. *Consider a set P of nm directional sensors located in a lattice $\mathcal{L}$ of size $m \times n$ and let the antennae have range r such that* $\max(n,m) \geq \lceil 3r/2 \rceil$. *Then, we have that* $\Phi_r(P, \mathcal{L}) \geq \frac{2\pi}{r}$.

Proof. Assume without loss of generality that $n \leq m$. It is sufficient to orient the antennae and provide coverage for a single row of the lattice and apply the result to each row so as to cover each point in P; see Figure 3.

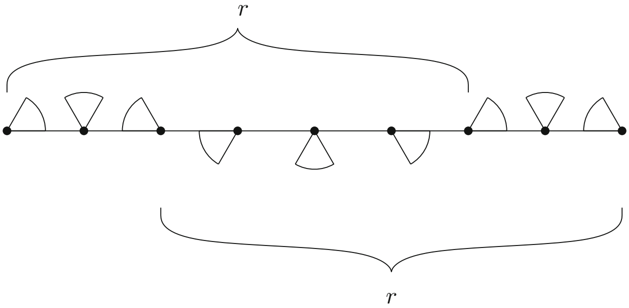


Fig. 3. $r = 6$, a lattice of size $n = 9$ and the initial positions

We orient (i, j)-th sensor as $F_{p_{i,j}}(r, j \cdot 2\pi/r; 0)$. To prove that $\mathcal{L}$ is covered, consider a pair of sensors $p_{i,j}$ and $p_{i,k}$ at distance $\lceil 3r/2 \rceil$. Since $m \geq \lceil 3r/2 \rceil$, such a pair exists. Furthermore, there are $3r/2$ sensors between $p_{i,j}$ and $p_{i,k}$. Observe that the sensors between $p_{i,j}$ and $p_{i,k}$ are oriented consecutively as Corollary 1. Therefore, the line segment $p_{i,j}p_{i,k}$ is always covered. The pseudocode is presented in Algorithm 2. This completes the proof of the theorem. □

3 Planar Configurations

In this section we consider configurations of sensors in the plane and study coverage for half-plane, infinite plane, and the convex hull of a set of points.

Algorithm 2. Initial orientation of sensors on a lattice $\mathcal{L}$ of size $m \times n$ that covers $\mathcal{L}$.

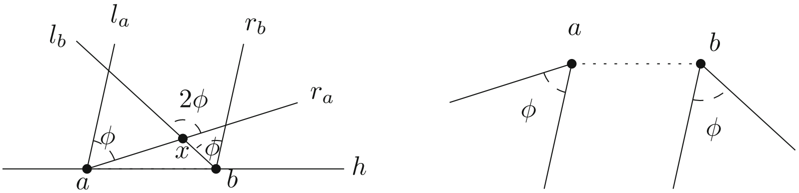
```
input  : $P, r$: $P$ points on a lattice of size $m \times n$ such that $\max(n,m) \geq \lceil 3r/2 \rceil$
output: Initial orientation of $P$
for $i \leftarrow 0$ to $n-1$ do
    for $j \leftarrow 0$ to $m-1$ do
        Orient the antenna at $p_{i,j}$ as $F_{p_{i,j}}(r, j \cdot 2\pi/r; 0)$;
```

3.1 Covering the Half-Plane

First we consider orientation algorithms for covering a half-plane determined by an infinite line. We say that two sensors a and b with sensor angle ϕ form a dark corridor at time t if $F_a(\rho;t) \cap F_b(\rho+\phi;t) = \emptyset$.

Lemma 1. *Let a and b be two directional sensors of angle ϕ on a horizontal line. Assume that the initial orientations of the antennae at a, b are $F_a(\pi;0), F_b(\pi-\phi;0)$, respectively. Further, assume that the x-coordinate of a is less than the x-coordinate of b. If $0 \leq t \leq \pi$, the intersection of the sensors covers a circular sector 2ϕ. If $\pi < t < 2\pi$, they leave a black corridor.*

Proof. Let l_a, l_b, r_a and r_b be the left and right rays that define the wedges of the sensors at a and b respectively. Let h be the horizontal. At time t, $\angle(r_a,h) = \pi - t$, $\angle(l_a,h) = \pi + \phi - t$, $\angle(r_b,h) = \pi - \phi - t$ and $\angle(l_b,h) = \pi - t$. Observe that $\angle(r_a,r_b) = -\phi$ and $\angle(l_b,l_a) = \phi$. Therefore, when $\pi < t < 2\pi$, the rays of r_a and l_b do not intersect, i.e., $F_a(\pi;t) \cap F_b(\pi-\phi;t) = \emptyset$ since the x-coordinate of a is less than the x-coordinate of b and a black corridor is formed; see Figure 4b. However, when $0 \leq t \leq \pi$, r_a intersects l_b since the x-coordinate of a is less than the x-coordinate of b. Consider the intersection point x between r_a and l_b; see Figure 4a. It is not difficult to see that l_a and r_b determine a coverage wedge incident to x of angle 2ϕ. □



(a) $0 \leq t \leq \pi$

(b) $\pi < t < 2\pi$ a dark corridor is formed

Fig. 4. Two directional sensors

Theorem 3. *For a set P of $n \geq 3$ floodlights on a line $\mathcal{L}$ and the upper half-plane $\mathcal{H}_u(\mathcal{L})$ determined by $\mathcal{L}$ we have that $\Phi(n, \mathcal{H}_u) = \frac{3\pi}{n}$.*

Proof. We will prove that the initial orientation depicted in Figure 1 of Theorem 1 also covers the half-plane $\mathcal{H}_u(\mathcal{L})$ determined by $\mathcal{L}$. Similarly to the proof of Theorem 1 we assume that $\mathcal{L}$ is horizontal and the x-coordinate of the sensor p_i is less than the x-coordinate of the sensor p_{i+1}, for all $i = 0, 1, \ldots, n-2$. As before, we define a dual plane as follows: 1) each sensor i is the circular sector of a unitary circle C delimited by $i \cdot 3\pi/n$ and $(i+1) \cdot 3\pi/n$, 2) at time t, the line $\mathcal{L}$ is represented as a directed line segment $\overrightarrow{\mathcal{L}}$ such that $\overrightarrow{\mathcal{L}}$ crosses the center of C and the head of $\overrightarrow{\mathcal{L}}$ forms an angle t with the horizontal, and 3) the upper half-plane $\mathcal{H}_u(\mathcal{L})$ determined by $\mathcal{L}$ is represented by the left half-plane determined by $\overrightarrow{\mathcal{L}}$; see Figure 2a.

In the dual plane sensors are static and $\mathcal{L}$ rotates during the time. The orientation $\overrightarrow{\mathcal{L}}$ of $\mathcal{L}$ preserves the sensor rotations and the upper half-plane $\mathcal{H}_u$ of the original plane.

Since the sum of the angles is 3π, the circular sector $[0,\pi)$ of C in the dual plane is always covered twice and the circular sector $[\pi,2\pi)$ of C is covered once. Let S_1 be the set of circular sectors that covers the head of $\overrightarrow{\mathcal{L}}$ and let S_2 be the set of circular sectors that covers the tail of $\overrightarrow{\mathcal{L}}$. Observe that either $|S_1| = 1$ and $|S_2| = 2$ or $|S_1| = 2$ and $|S_2| = 1$. We will prove that there exists an increasing subsequence $p_j, p_{j+1}, ..., p_i$ so as by Lemma 1 it covers the left half-space determined by $\overrightarrow{\mathcal{L}}$. If $|S_1| = 1$, let $p_i \in S_1$ and $p_j \in S_2$ such that j is the min label in S_2. Otherwise if $|S_1| = 2$, let $p_i \in S_2$ and $p_j \in S_1$ such that j is the min label in S_1. Since $j < i$, the increasing subsequence is determined by the sensors $p_j, p_{j+1}, ..., p_i$.

To prove the bound is tight, assume by contradiction that $\Phi(n,\mathcal{H}_u(\mathcal{L})) < 3\pi/n$; see Figure 2b. Since $\Phi(n,\mathcal{H}_u) < 3\pi/n$ they cover less than 3π. Therefore, there exists a time t such that only two sensors fully cover $\overrightarrow{\mathcal{L}}$. Assume a covers the tail and b covers the head of $\mathcal{L}$. Therefore, $\mathcal{L}$ is fully covered. However, at time $t+\pi$, a covers the head and b covers the tail of $\mathcal{L}$. Therefore, the line segment ab is not covered. This contradicts the assumption. This completes the proof of the theorem. □

Observe that if the points are uniformly distributed in the lower line of a rectangle of size $l \times 1$, it is sufficient to have a range equal to $\sqrt{l^2+1}$. Thus, we have a corollary to Theorem 3.

Corollary 2. *For a set P of $n \geq 2$ sensors uniformly distributed in the lower line of a rectangle $\mathcal{R}$ of size $l \times 1$, we have that $\Phi_r(P,\mathcal{R}) = \frac{3\pi}{n}$; where $r = \sqrt{l^2+1}$.*

Theorem 4. *Assume we are given a set P of mn directional sensors of radius r located in a $m \times n$ lattice $\mathcal{L}$ where $\min(n,m) > 1$ and $\max(n,m) \geq \lceil 3r/2 \rceil$. Let $CH(\mathcal{L})$ be the convex hull of $\mathcal{L}$. Then, we have that $\Phi_r(P,CH(\mathcal{L})) \geq \frac{2\pi}{\sqrt{r^2-1}}$.*

Proof. Without loss of generality assume that $m \geq n$. Let $p_{i,j}$ be the sensor at row i and column j for $0 \leq i < n$ and $0 \leq j < m$. Orient $p_{i,j}$ as $F_{p_{i,j}}(r, j \cdot \frac{2\pi}{\sqrt{r^2-1}}; 0)$; see Figure 5. To prove that it is always sufficient, consider a pair of sensors $p_{i,j}$ and $p_{i,k}$ in the row $i < n-1$ at distance $\left\lceil \frac{3\sqrt{r^2-1}}{2} \right\rceil$. Since $m \geq \lceil \frac{3r}{2} \rceil$, such a pair always exists. Furthermore, there are $\left\lceil \frac{3\sqrt{r^2-1}}{2} \right\rceil$ sensors between $p_{i,j}$ and $p_{i,k}$. Observe that the sensors between $p_{i,j}$ and $p_{i,k}$ are consecutively oriented as Corollary 2. Let $R(p_{i,j}, p_{i+1,k})$ be the rectangle formed by $p_{i,j}$ and $p_{i+1,k}$. From Corollary 2 $R(p_{i,j}, p_{i+1,k})$ is fully covered with range r since $\left\lceil \frac{3\sqrt{r^2-1}}{2} \right\rceil \frac{2\pi}{\sqrt{r^2-1}} > 3\pi$. We give below the pseudocode of the main algorithm. This completes the proof of the theorem. □

3.2 Covering the Plane

Next we consider antennae orientation algorithms for covering the entire plane. The case of coverage with two antennae is simple, but coverage with three antennae turns out to be quite intricate and elegant.

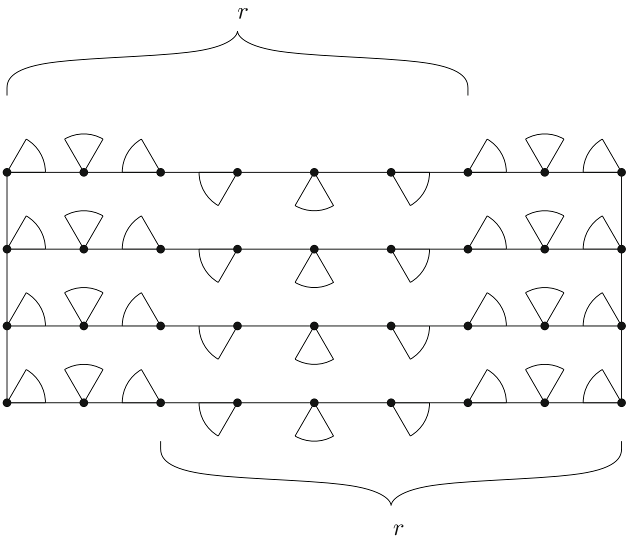


Fig. 5. $r = 4$, a grid of size 4×9 and the initial position

Algorithm 3. Initial orientation of sensors on a lattice $\mathcal{L}$ of size $m \times n$ that covers $CH(\mathcal{L})$.

```
input  : $P, r$: $P$ points on a lattice of size $m \times n$ such that $n \geq \lceil 3r/2 \rceil$
output: Initial orientation of $P$
for $i \leftarrow 0$ to $m-1$ do
    for $i \leftarrow 0$ to $n-1$ do
        Orient the antenna at $p_{j,i}$ as $F_{p_{j,i}}(r, i \cdot \frac{2\pi}{\sqrt{r^2-1}}; 0)$;
```

Theorem 5. *Let $\mathcal{P}$ be the entire plane. We have that $\Phi(2,\mathcal{P}) = 2\pi$.*

Proof. Assume by contradiction that $\omega := \Phi(2,\mathcal{P}) < 2\pi$; see Figure 6.

Let p_1, p_2 be two floodlights of angle ω. Assume that there is an initial orientation of p_1 and p_2 such that every point in the plane is covered at all times. However, there

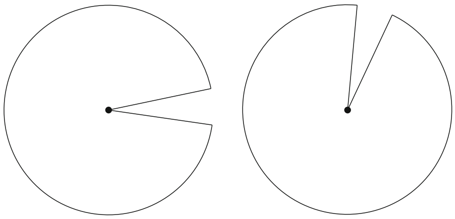


Fig. 6. $\Phi(2,\mathcal{P}) = 2\pi$

exists an uncovered wedge w_1 forming an angle $2\pi - \omega$ emanating from p_1 and another uncovered wedge w_2 forming an angle $2\pi - \omega$ emanating from p_2. Clearly, at some time t as the sensors rotate with identical constant speeds the sensor p_2 will be within the wedge w_1. But then it is not difficult to see that a planar region is left which is covered by neither p_1 nor p_2, which is a contradiction. □

Theorem 6. *Let $\mathcal{P}$ be the entire plane. We have that $\Phi(3,\mathcal{P}) = \pi$.*

Proof. Let p,q,r be three directional sensors in the plane. If the sensors are collinear then the initial configuration depicted in Figure 7 can be easily seen to be correct.

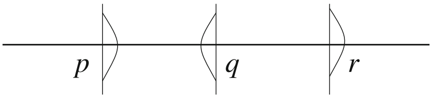


Fig. 7. Initial orientation for three sensors in collinear position

Therefore we may assume, without loss of generality, that the three sensors are not in collinear position. Further we may assume that the line segment pr is horizontal and q is above pr. Let C be the circumcircle C of p,q,r. Orient p as $F_p(l;0)$, where l is the tangent of C at p, q as $F_q(\pi + \angle(qpr);0)$ and r as $F_r(0;0)$ as depicted in Figure 8a. Consider any point a in the circumference of C of pqr. Observe that the angle that each sensor forms with a is equal to the arc; see Figure 8b. Therefore, they intersect at a. It can be verified that when a is in the arc pr, qr leave an uncovered wedge with apex at p. However, p covers the uncovered wedge. When a is in the arc rq, the roles change to p,q and r respectively and when a is in the arc qp, the roles change to pr and q respectively. This proves the upper bound if the points are not collinear.

Assume now that p,q,r are collinear. Without loss of generality assume that they are on a horizontal line and the x-coordinate of q is greater than the x-coordinate of p and smaller than the x-coordinate of r. Orient p,q,r as $F_p(0;0)$, $F_q(\pi;0)$ and $F_r(0;0)$. By Lemma 1, p and q cover the plane at time $t < \pi$ and q and r cover the plane at time $\pi \leq t < 2\pi$.

To prove that the bound is tight, assume by contradiction that $\Phi(3,\mathcal{P}) = \pi - \varepsilon$. Assume that at time t the sensors cover the plane. Therefore, there exists a point a in the coverage area of p where two line wedges incident to q and r intersect since two sensor cannot cover the plane as depicted in Figure 8c. However, a is not covered at time $t+\pi$ since $\Phi(3,\mathcal{P}) = \pi - \varepsilon$. □

Theorem 7. *Let P be a set of $n \geq 3$ points in general position and $CH(P)$ be the convex hull on P. We have that $\Phi_r(P,CH(P)) \geq \pi$ where r is twice the longest edge of the Delaunay Triangulation of P.*

Proof. Consider the Delaunay triangulation $DT(P)$ of P. Let G be the dual graph of $DT(P)$ where each triangle $\triangle(u)$ of $DT(P)$ is a vertex u in G and two vertices u,v are adjacent in G if and only if $\triangle(u) \cap \triangle(v) \neq \emptyset$ in $DT(P)$. Observe that unlike Voronoi diagrams, there is no vertex in the dual for the outer face and G is not planar. Let I be

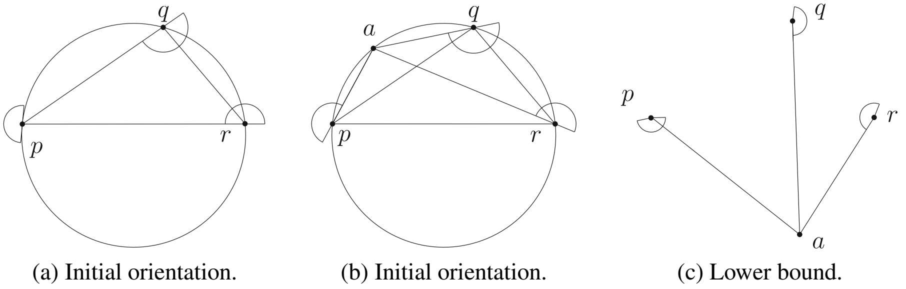


Fig. 8. Three points covering the plane

a maximal independent set of G. For each vertex $u \in I$ we orient the directional sensors that form the triangle $\triangle(u)$ as in Theorem 6. Let r be twice the longest edge of $DT(P)$. We claim that r is always sufficient to cover $CH(P)$. To prove the claim assume on the contrary that it is not sufficient. Therefore, there exists a time where a triangle $\triangle(v)$ is not fully covered. From Theorem 6, v is not a neighbor of $u \in I$ since the sensors of $\triangle(u)$ cover all the adjacent triangles at all times. Therefore I is not maximal. This contradicts the assumption. □

4 Coverage with Gap Time at Most T

In this section we study a variant of the problem in which we allow points to be uncovered for a period of time no longer than T. Let $\Phi_r(P;\mathcal{R},T)$ be the infimum over all angles $\phi \leq 2\pi$ such that if sensors of angle ϕ and range r are located at the points then there is an initial orientation of the sensors so that every point is left uncovered for a period of time no longer than $T < 2\pi$ under continuous rotation of the directional sensors. We will prove that in fact the two problems are equivalent.

Theorem 8. $\Phi_r(P,\mathcal{R};T) = \Phi_r(P,\mathcal{R}) - T$

Proof. Assume an initial orientation of the sensor in P with angle $\Phi_r(P,\mathcal{R})$. For each sensor p of P, we will show how to orient p with angle $\Phi_r(P,\mathcal{R}) - T$ such that every point is uncovered for a period of time no longer than T. Let $F_p(r,\rho;0)$ be the initial orientation of p with angle $\Phi_r(P,\mathcal{R})$ such that $\mathcal{R}$ is fully covered at all times. Let the initial orientation of p as $F_p(r,\rho+T;0)$ with angle $\Phi_r(P,\mathcal{R}) - T$ We claim that the initial orientation does not leave any point unattained for longer than time T. Assume on the contrary that there exists a point a such that it is uncovered for a time greater than T. Therefore, a is not covered by any sensor p_i with angle $\Phi_r(P,\mathcal{R})$. This contradicts the assumption. □

5 Software

We implemented our algorithms in C++ to confirm our results. The programs can be downloaded from http://people.scs.carleton.ca/∼omponce/floodlights/index.html.

6 Conclusion and Open Problems

We have studied the problem of determining the initial orientation of rotating directional sensors so as to ensure uninterrupted coverage of a planar region under continuous rotation of the antennae. We studied the problem in several settings, including sensors located in lattice and arbitrary configurations as well as for various types of regions. Several open problems remaining concern angle/range tradeoffs. Additional problems concern determining tight bounds on the angle $\Phi(P)$ for arbitrary and specific configurations of points P, e.g., points in convex position, etc. In this paper we proved that $\Phi(n,\mathcal{P})$ is equal to 2π for $n = 2$, and equal to π for $n = 3$. However, nothing non-trivial is known for $n \geq 4$. Additional interesting questions arise by considering alternative settings concerning the speeds and rotation directions of the antennae, as well as k-coverage whereby k antennae are required to monitor all points at all times.

References

1. Cardei, M., Wu, J.: Energy-efficient coverage problems in wireless ad-hoc sensor networks. Computer Communications 29(4), 413–420 (2006)
2. Chvatal, V.: A combinatorial theorem in plane geometry. Journal of Combinatorial Theory, Series B 18(1), 39–41 (1975)
3. Fisk, S.: A short proof of chvátal's watchman theorem. Journal of Combinatorial Theory, Series B 24(3), 374 (1978)
4. González-Baños, H.: A randomized art-gallery algorithm for sensor placement. In: Proceedings of the Seventeenth Annual Symposium on Computational Geometry, pp. 232–240. ACM (2001)
5. Gupta, H., Das, S.R., Gu, Q.: Connected sensor cover: self-organization of sensor networks for efficient query execution. In: Proceedings of the 4th ACM International Symposium on Mobile Ad hoc Networking & Computing, pp. 189–200. ACM (2003)
6. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. Mobile Networks and Applications 10(4), 519–528 (2005)
7. Lee, D., Lin, A.: Computational complexity of art gallery problems. IEEE Transactions on Information Theory 32(2), 276–282 (1986)
8. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: INFOCOM 2001: Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1380–1387. IEEE (2001)
9. O'Rourke, J.: Art gallery theorems and algorithms, vol. 57. Oxford University Press, Oxford (1987)
10. Poduri, S., Sukhatme, G.S.: Constrained coverage for mobile sensor networks. In: 2004 Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 1, pp. 165–171. IEEE (2004)
11. Urrutia, J.: Art gallery and illumination problems. In: Handbook of Computational Geometry, pp. 973–1027 (2000)
12. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 28–39. ACM (2003)

Social Aspects to Support Opportunistic Networks in an Academic Environment

Radu Ioan Ciobanu, Ciprian Dobre, and Valentin Cristea

University Politehnica of Bucharest, Romania
Faculty of Automatic Control and Computers
radu.ciobanu@cti.pub.ro, {ciprian.dobre,valentin.cristea}@cs.pub.ro

Abstract. As wireless and 3G networks become more crowded, users with mobile devices have difficulties in accessing the network. Opportunistic networks, created between mobile phones using local peer-to-peer connections, have the potential to solve such problems by dispersing some of the traffic to neighboring smartphones. Recently various opportunistic routing or dissemination algorithms were proposed and evaluated in different scenarios emulating real-world phenomena as close as possible. In this paper we present an experiment performed at the Politehnica University of Bucharest in which we collected social and mobiltity data to evaluate opportunistic routing and dissemination algorithms. We present an analysis of our findings, highlighting key social and mobility behavior factors that can influence such opportunistic solutions. Most importantly, we show that by adding knowledge such as social links between participants in an opportunistic network routing and dissemination algorithms can be greatly improved.

1 Introduction

Opportunistic mobile networks consist of human-carried mobile devices that communicate with each other in a store-carry-and-forward fashion, without any infrastructure. Compared to classical networks, they present distinct challenges. In opportunistic networks, disconnections and highly variable delays caused by human mobility are the norm. The solution consists of dynamically building routes, as each node acts according to the store-carry-and-forward paradigm. Thus, contacts between nodes are viewed as an opportunity to move data closer to the destination. Such networks are therefore formed between nodes spread across the environment, without any knowledge of a network topology. The routes between nodes are dynamically created, and nodes can be opportunistically used as a next hop for bringing each message closer to the destination. Nodes may store a message, carry it around, and forward it when they encounter the destination or a node that is more likely to reach the destination.

In order for researchers to be able to implement dissemination algorithms for opportunistic networks, real-life traces can be used to offer information about the patterns that people carrying mobile devices follow. Traces are taken using different types of mobile devices for various kinds of communication as well as for

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 69–82, 2012.

different scenarios. This paper describes a social tracing experiment that took place between November and December 2011 at the Politehnica University of Bucharest and the way it was implemented. Furthermore, we analyze the results obtained and try to gather information that may be relevant in designing a good dissemination algorithm for opportunistic networks.

In an opportunistic network, the members are people that carry mobile devices. These people are organized into communities, according to common professions, workplaces, interests, etc. Generally, members of the same community interact with each other more often than with members of outside communities, so the community organization should be taken into consideration when designing algorithms for opportunistic networks. In recent years, due to the advent of social networks and applications, researchers have started showing interest in the use of such elements in opportunistic algorithms. We show here that adding knowledge about social links between opportunistic network nodes to routing and dissemination algorithms greatly improves their effect.

2 Related Work

There are two ways of testing the performance of a data dissemination algorithm in an opportunistic network. First of all, traces such as the ones presented in this paper can be taken from various situations. Such traces have been performed for WiFi [1,2] or Bluetooth [3]. The Bluetooth trace experiment is similar to the one presented in this paper in terms of number of participants, but the duration differs greatly, as the traces in [3] were performed for 3 and 5 days (compared to 35 in our case, as presented in Sect. 3). Another difference between our trace and the ones from [3] is that the iMote traces are performed by nodes that interact with each other for long times during the day, as the carriers work in the same enclosed place for large parts of the day. We show in Sect. 3 that our trace covers a larger array of node types because the participants are not grouped together. A good place for finding mobility traces for various situations is CRAWDAD [4], a community resource for archiving wireless data. The second way of testing a dissemination algorithm is to use mobility models. There have been several such models proposed in recent years. The research began with random models such as the waypoint model, and continued with mobility models that take into consideration the social aspect of human movement [5] as well as the attraction of physical locations [6]. Existing human mobility models for oppotunistic networks, including models that explore location preference or use the social graph, are reviewed in detail in [7], and a taxonomy for classifying such models is proposed.

A thorough review of opportunistic networking is presented in [8]. The analysis, developed in the context of the EU Haggle project, highlights the properties of main networking functions, including message forwarding, security, data dissemination and mobility models. The authors also propose various solutions for communication in opportunistic networks, and introduce HCMM, a mobility model that merges the spatial and social dimensions. Several well-known opportunistic forwarding algorithms are also presented, such as BUBBLE Rap [9], PROPICMAN [10] and HIBOp [11].

There are several papers that propose dissemination algorithms for opportunistic networking. Authors of [12] propose Socio-Aware Overlay, an algorithm that creates an overlay for an opportunistic network with publish/subscribe communication. The overlay is composed of nodes with high values of centrality, so that the chosen broker node maintains a higher message delivery rate. The Socio-Aware Overlay algorithm is socially-aware, having its own community detection methods. Thus, the authors of the article propose two algorithms for distributed community detection, named Simple and k-CLIQUE. Another dissemination algorithm is proposed in [13]. Choosing the next-hop node is a scheduling hard problem, with fault-tolerant requirements [14]. Wireless Ad Hoc Podcasting has the purpose of wireless ad hoc delivery of content among mobile nodes. The technique enables the distribution of content using opportunistic contacts whenever podcasting devices are in wireless communication range. Authors of [15] propose a dissemination technique called ContentPlace, that attempts to deal with data dissemination in resource-constrained opportunistic networks by making content available in regions where interested users are present, without overusing available resources. In order to optimize content availability, ContentPlace exploits learned information about users' social relationships, to decide where to place user data. ContentPlace's design is based on two assumptions: that the users can be grouped together logically, according to the type of content they are interested in, and that their movement is driven by social relationships. In order to be able to select data from an encountered node, nodes from ContentPlace use a utility function by means of which each node can associate a utility value to any data object. When a node encounters a peer, it computes the utility values of all the data objects stored in the local and in the peer's cache. Then, it selects the set of data objects that maximizes the local utility of its cache.

A taxonomy for data dissemination algorithms is proposed in [16]. The authors propose splitting such algorithms in four large categories. The first category deals with the infrastructure of the network, meaning the way the network is organized into an overlay for the nodes. Then, the dissemination techniques are also split according to the characteristics of their nodes, such as node state and node interaction (which includes node discovery, content identification and data exchange). The third category of the taxonomy is represented by content characteristics, meaning the way content is organized and analyzed, and finally the last category (and the most important one) is social awareness. Social awareness is considered to be the future of opportunistic networks, because the nodes in such a network are mobile devices carried by humans, which interact with each other according to social relationships.

Similar to the approach proposed in this paper, the addition of social network information to opportunistic routing has been studied in [17]. The authors consider two types of networks: a detected social network (DSN) as given by a community detection algorithm such as k-CLIQUE and a self-reported social network (SRSN) as given by Facebook relationships. When two nodes meet in their simulation, they exchange data only if they are in the same network (either DSN or SRSN). The authors show that using SRSN information instead of DSN decreases the delivery cost and produces comparable delivery ratio. Several other papers address the issue of using social information in opportunistic networks. In [7], an analytical model for the expected number of hops and delay of messages delivered in a social-based opportunistic routing algorithm is proposed, where the forwarding process is modeled as a semi-Markov process. Social information about the participants in an opportunistic network can be used not only for data forwarding, but also for content sharing. Thus, the authors of [18] propose a context- and social-aware middleware that learns context and social information about the nodes in the network, which is then used to predict their future movement. The middleware was integrated with the Haggle architecture and was used for content sharing, yielding up to 200% improvement in terms of hit rate and 99% reduction in resource consumption in terms of traffic generated in the network.

3 Social Tracing

In order to obtain trace information regarding the mobility of the members of a faculty, a real-world tracing experiment has been performed at the Politehnica University of Bucharest, in the autumn-winter season of 2011. This section presents the setup and additional details about this experiment.

3.1 Social Tracer

Tracing was performed using an Android application we developed entitled Social Tracer, which is presented in more detail in [19]. The participants have been asked to run the application whenever they are in the faculty grounds, as we were interested in collecting data about the mobility and social traces in an academic environment. Social Tracer sends regular Bluetooth discovery messages at certain intervals, looking for any type of device that has its Bluetooth on. These include the other participants in the experiment, as well as phones, laptops or other type of mobile devices in range. The reason Bluetooth was preferred to WiFi is mainly the battery use [20]. For example, in 4 hours of running the application on a Samsung I9000 Galaxy S with discovery messages sent at every 5 minutes, the application used approximately 10% of the battery's energy. The period between two successive Bluetooth discovery invocations can be set from the application, ranging from 1 to 30 minutes (the participants have been asked to keep it as low as possible, in order to have a more fine-grained view of the encounters).

When encountering another Bluetooth device, the Social Tracer application logs data containing its address, name and timestamp. The address and name are used to uniquely identify devices, and the timestamp is used for gathering contact data. Data logged is stored in the device's memory, therefore every once in a while participants were asked to upload the data collected so far to a central server located within the faculty premises. All gathered traces were then parsed and merged to obtain a log file with a format similar to the ones in [4]. Successive encounters between the same pair of devices within a certain time interval were considered as continuous contacts, also taking into consideration possible loss of packets due to network congestion or low range of Bluetooth. Data gathered this way is presented and analyzed in Sect. 4.

3.2 Experimental Setup

The experiment was performed for a period of 35 days at the Politehnica University of Bucharest between November 18 and December 22 2011. There were a total of 22 participants, chosen to be as varied as possible in terms of year, in order to obtain a better approximation of mobility in a real academic environment. Thus, there were twelve Bachelor students (one in the first year, nine in the third and two in the fourth), seven Master students (four in the first year and three in the second) and three research assistants. The participating members were asked to start the application whenever they arrived at the faculty and to turn it off when they left, because we were only interested in the mobility patterns and social interaction in the academic environment. However, this did not always happen, but the outcome of the experiment was not affected because the only devices seen after leaving the faculty were external devices.

4 Trace Analysis

This section presents a detailed analysis of the logs obtained at the end of the experiment described in Sect. 3.

4.1 Details

We define internal devices as the ones carried by the participants in the experiment, while external devices are represented by other nodes encountered during the course of the experiment. There were 22 internal devices numbered from 0 to 21. The total number of contacts between two internal devices (i.e. internal contacts) was 341, while the number of external contacts was 1127. A contact is considered to start at the first time a certain device was seen and to end at the last time it was seen in a given time interval. There were 655 different external devices sighted during the course of the experiment. This means that in average each different external device has been seen about 2 times. External devices may be mobile phones carried by other students or laptops and notebooks found in the laboratories at the faculty. Some of these external devices have high contact times because they may belong to the owner of the internal device that does the discovery, therefore being in its proximity for large periods of time. However, external contacts are in general relatively short.

4.2 Contact and Inter-contact Times

Encounters in an opportunistic network are characterized by two important notions: contact time and inter-contact time. The contact time represents the duration of a contact between two devices from the moment they discover they are in range until the moment the link between them is gone. This represents the time window in which the two participating nodes can send data to each other. Inter-contact times are intervals between two successive encounters of the same two devices. They are relevant in deciding whether data should be sent directly between two nodes when they are in range or whether it should be relayed to a third node for forwarding.

Figure 1 shows the distribution of contact and inter-contact times for the entire duration of the experiment (ranging from 2 minutes to 35 days) for all internal devices. Axis Y presents the percentage of time values that are greater than the time on axis X. As shown in [3], the distribution of contact times follows an approximate power law for both internal and external devices, as well as contact time and inter-contact time. The contact time data series are relevant when discussing the bandwidth required to send data packets between the nodes in an opportunistic network, because they show the time in which a device can communicate with other devices. As stated before, the number of internal contacts is 341, with the average contact duration being 30 minutes, which means that internal contacts have generally been recorded between devices belonging to students attending the same courses or lecturers and research assistants teaching those courses. External contacts also follow an approximate power law, with an average duration of 27 minutes. However, in this case there are certain external contacts that have a duration of several hours. This situation is similar to the one previously described, where these devices belong to the same person carrying the internal device. The inter-contact time distribution shows a heavy tail property, meaning that the tail distribution function decreases slowly. The impact of such a function in opportunistic networking has been studied in more detail in [21] for four different traces. The authors conclude that the probability of a packet being blocked in an inter-contact period grows with time and that there is no stateless opportunistic algorithm that can guarantee a transmission delay with a finite expectation.

Figure 2 shows contact and inter-contact times for encounters with any nodes. Thus, contact time in this case (called any-contact time in [3]) represents the time in which any internal or external node is in range with the current observer, while the inter-any-contact time is the time when the current device does not see anyone in range. These any-contact times are greater than regular contact times, but the shape of the distribution is also a power law function. A conclusion that can be drawn from these charts is, as observed in [3], that durations of contact times are bigger and intervals between contacts are smaller, so if a node wants to perform a multicast or to publish an object in a publish/subscribe environment it has a great chance of being able to do so.

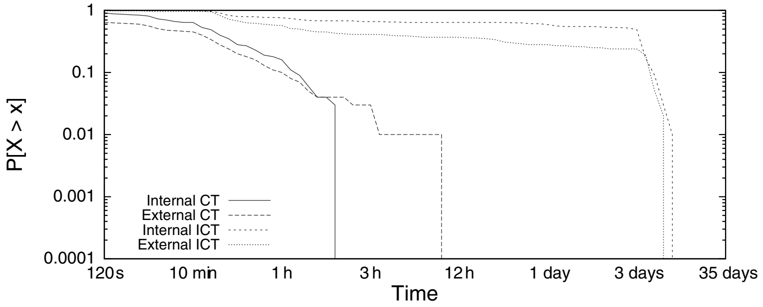


Fig. 1. Probability distribution of contact and inter-contact times (CT = contact time, ICT = inter-contact time)

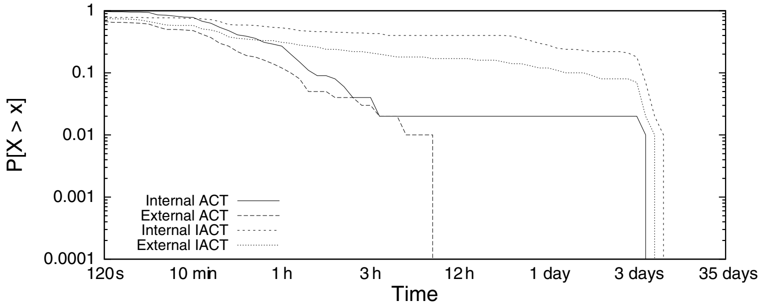


Fig. 2. Probability distribution of any-contact and inter-any-contact times (ACT = any-contact time, IACT = inter-any-contact time)

4.3 Contact Distribution

Figure 3 shows the distribution of the number of times a node (internal or external) was sighted by a device participating in the experiment. It can be seen that the maximum number of encounters of an internal device is 55 during the course of the 35 days of the experiment, whereas some internal nodes have never been seen. Most internal devices have been seen from 16 to 20 times. As for external devices, the majority of them have been encountered less than 5 times, with 534 of them having been sighted only once. There are few exceptions, as three external devices have been encountered more than 16 times. The conclusion is that there is a large number of nodes available in such an environment that can be used to relay a message, meaning that there is a lower chance of traffic congestion. Figure 4 presents the number of times specific pairs of devices saw each other. It shows that the maximum number of contacts between two internal nodes or an internal and an external node is 17. Generally the number of contacts with external devices is larger than the number of contacts with internal devices. This shows that the participants in the experiment have been chosen well so that

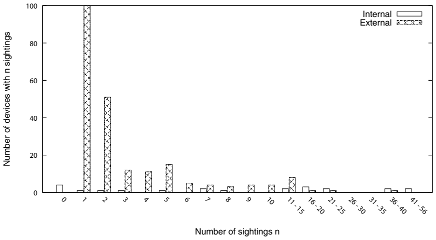


Fig. 3. Distribution of the number of sightings of a device

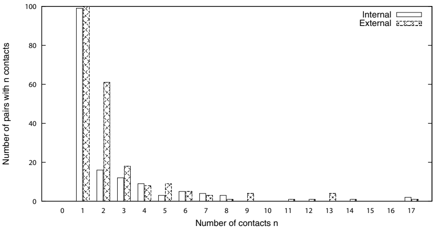


Fig. 4. Distribution of the number of contacts between pairs of devices

they represent various groups from the social and logical grouping of nodes in a network based on mobile device carriers in an academic environment.

4.4 Communities and Social Structures

As stated in Sect. 2, the social aspect has become very important in the world of opportunistic networking, because mobile devices are carried by people that are organized into communities and social circles. Users from the same community or social circle tend to interact more with each other, so relaying a packet by taking into consideration the community an encountered node belongs to could lead to a lower latency and a better hit rate. Human mobility models such as CMM [5] and HCMM [6] have been proposed and implemented, but an experimental approach could show the interaction patterns better.

The environment represented by the faculty grounds already has a logical organization into communities, namely the groups of students and the teachers or lecturers. At the University Politehnica of Bucharest there are four years for Bachelor students, each split into several groups of about 30 students each. For Master's students, the two years are formed of seven directions with about 20 students each. We tried to choose the participants in the experiment so that the distribution would be as good as possible, as shown in Sect. 3. However, because there were only 22 participants in our experiment, we decided that the logical grouping should be done by year instead of group. We applied the k-CLIQUE algorithm for community detection [22] on the traces collected by Social Tracer. k-CLIQUE dynamically detects the community of a node by analyzing its encounters with other devices. There are two important parameters to the k-CLIQUE algorithm: the contact threshold and the community threshold. The contact threshold specifies the amount of time that two nodes have to be in contact before being considered as part of the same community, while the community threshold is used to specify the number of community nodes two encountering devices must have in common in order for them to belong to the same community. The community graph obtained after applying k-CLIQUE is presented in Fig. 5, along with the logical organization of participants into year groups. In the figure, Bachelor students are represented as triangles (first year), circles (third year) and double circles (fourth year), Master's students are shown as pentagons (first year) and diamonds (second year), while assistants and lecturers are squares. An arrow from node A to node B means that A sees B as part of its community. The algorithm was applied only for internal nodes, using a contact threshold of fifteen minutes and a community threshold of five nodes, values that were chosen after analyzing the traces. Because social networks and groups of communities are represented as matrices, we define the similarity value between two matrices as the percentage of values that are equal in both of them. Thus, the similarity value between the k-CLIQUE graph and the logical distribution of participants into year groups is 79.95%. This shows that k-CLIQUE functions correctly in the case of our trace.

Because a logical grouping into communities may not always be as straightforward as in this case, the social relationships between device owners can be taken into account. The social graph of the participants in our experiment is shown in Fig. 6, where the year group representation is the same as in Fig. 5 and the edges symbolize a social link.. Some nodes (such as 1, 4, 8 and 20) are represented by students that participated in the experiment but did not have or did not provide a Facebook account. It can be observed from Fig. 6 that the node with the most social links (12) is 11, which is followed by nodes 3 and 7 with 11 links each. It can also easily be seen that most nodes that are in the same community share a social link between them, as well as the fact that in most cases the number of social links of a node is close to the number of communities it belongs to according to k-CLIQUE. This means that a more popular node in terms of social relationships will belong to more communities, which makes it a better candidate for relaying data for other nodes in the opportunistic network. The similarity value for the social network organization and k-CLIQUE is 83.06%, showing that k-CLIQUE is even better at detecting social communities than is it for logical grouping.

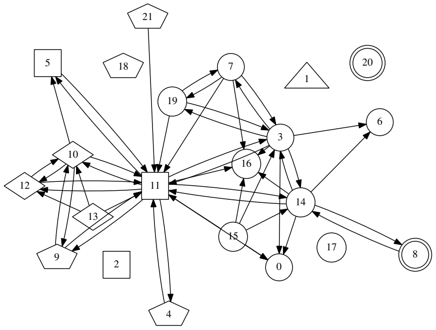


Fig. 5. The graph of participants in the experiment as computed by k-CLIQUE

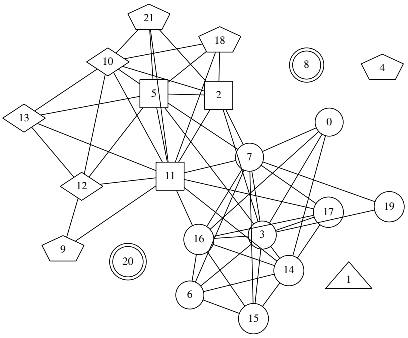


Fig. 6. The social graph of participants in the experiment (via Facebook)

5 Opportunistic Networking Using Social Organization

The social organization of members in an opportunistic network can be used to improve the effectiveness of routing algorithms. In order to prove this, we have implemented some modified versions of the distributed BUBBLE Rap algorithm (DiBuBB [9]) that take social links into consideration when routing. This section describes our modified versions of DiBuBB along with the simulation setup, the metrics used for analyzing the performance of our improvements and the results obtained through the simulation.

5.1 Simulation Setup

BUBBLE Rap [9] is a routing algorithm for opportunistic networks that uses knowledge about nodes' communities and centralities to deliver messages. It uses a node's betweenness centrality, which represents the number of times a node is on the shortest path between two other nodes in the network. Each node in BUBBLE Rap has two centrality values, a local one (for its own community) and a global one. A node sends a message to nodes that have a higher global centrality than it until the message arrives at the destination community. Then, the local ranking is used to send the message up the community hierarchy until it reaches the destination. Community detection is done using k-CLIQUE, while the centralities are computed by carrying out an emulation that replays collected mobility traces, applies a flooding algorithm, and then computes the number of times a node acts as a relay on a shortest path. However, such a method is not feasible in real life, so a distributed version of BUBBLE Rap entitled DiBuBB was also proposed by the authors. It uses distributed k-CLIQUE [22] for community detection and a cumulative or single window algorithm for distributed centrality computation.

We have also implemented a version of DiBuBB to test the trace data presented in this paper. As stated in Sect. 3, we use k-CLIQUE with a contact threshold of fifteen minutes and a community threshold of five nodes to detect the communities. For computing the centrality values for each node in a distributed fashion we implemented the cumulative window (C-window) method, which counts the number of individual nodes encountered for each six-hour time window and then performs an exponential smoothing on the cumulated values. We will refer to this version from now on as “base”.

We consider that knowledge about the social relationships between members of an opportunistic network can increase the effectiveness of routing. Therefore, we modified the base version of DiBuBB to use the social network matrix instead of k-CLIQUE. Thus, when two nodes meet, instead of checking if they belong in the same community according to k-CLIQUE, they look for a social link between them. If that social link exists, then the nodes will compare their community centralities, and the one with the lower value will send its messages to the other one. If there is no link between the nodes, the global centralities will be verified. This will be refered to as the "social" version.

We then tried to take this approach one step further, by using the social network in the computation of centrality values as well. When there is an encounter between two nodes, they are considered to be in the same community if either they are seen as such by k-CLIQUE, or if they have a social distance of less than 3 (i.e. they are directly connected or they share a common friend). The centrality value is computed according to the following formula: $centrality = w_1 * S_{window} + w_2 * popularity$, where S_{window} is the original value of the centrality as computed by DiBuBB, *popularity* is the number of social links a node has, and w_1 and w_2 are weight values that follow the conditions $w_1 + w_2 = 1$ and $w_1 > w_2$. For nodes that are not part of the social network (e.g. participants in the tracing experiment that do no own or have not provided a Facebook account link), the centrality will be computed the same as in the base version of DiBuBB. Having weights for the two components of the centrality value allows us to fine-tune the algorithm according to the trace it is applied on. We will refer to this version as the "popularity" version.

In the simulation scenario, each node sends 11 messages to other nodes in the opportunistic network. We managed uncertainty while testing by eliminating from the list of internal devices the ones that have few or no encounters with other internal devices. Such nodes are participants in the experiment that either have not turned off the Social Tracer application when arriving at the faculty, or they have not been attending classes. There were four test cases in which nodes sent messages to other randomly chosen nodes (the destinations were kept the same between the three versions of DiBuBB for each test case).

5.2 Results

We applied the base, social and popularity versions of DiBuBB on the trace collected in our experiment. The first and most important metric that we chose is *hit rate*, which is computed as the ratio between successfully delivered and total messages. It suggests the efficiency of a routing algorithm and ideally it would be 100%. It shows the fraction of requests that can be served by a routing algorithm. Another used metric is the *delivery cost*, represented by the ratio between the total number of exchanged messages during the course of the experiment and the number of generated messages. It should be as low as possible and it shows the congestion of the network. The *latency* values show the time (in seconds) passed between generating a message and delivering it to the destination. In an opportunistic network, which is a type of delay tolerant network (DTN), delivery latency is not as important, but nonetheless it should be improved when possible. Finally, the *hop count* is the number of nodes that carried a message until it reached the destination on the shortest path.

The results of our testing scenario are shown in Table 1. It can be seen that the hit rate increases from base to social to popularity in each of the four test cases. The improvement caused by the social version is very significat, going up to 16% in some cases. The popularity version offers a smaller but very important

Table 1. Results of applying the three versions of DiBuBB (base - B, social - S, popularity - P) on four test cases. Latencies are in the DD:HH:MM:SS format

Test Case	**Run 1**			**Run 2**		
Metric	**B**	**S**	**P**	**B**	**S**	**P**
Hit Rate	81.81%	97.72%	98.48%	82.57%	98.48%	99.24%
Delivery Cost	5.89	8.76	15.71	5.87	9.06	15.43
Avg Latency	01:08:35:48	01:21:37:42	01:21:41:48	01:03:36:10	01:18:17:12	01:21:14:01
Min Latency	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12
Max Latency	31:10:53:00	31:10:53:00	32:10:14:29	20:10:32:01	20:10:32:01	32:10:14:29
Hop Count	1.22	3.13	6.33	1.18	3.23	6.01
Test Case	**Run 3**			**Run 4**		
Metric	**B**	**S**	**P**	**B**	**S**	**P**
Hit Rate	88.63%	96.21%	99.24%	90.15%	93.93%	97.72%
Delivery Cost	4.87	7.62	14.57	4.23	7.53	14.586
Avg Latency	01:23:43:41	02:20:04:19	02:17:33:00	02:03:05:56	02:23:47:10	02:20:50:52
Min Latency	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11
Max Latency	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43
Hop Count	2.57	3.87	6.16	2.73	4.04	6.17

increase in hit rate, which is brought close to 100%, the ideal value for this metric. However, having such a good hit rate comes with certain costs, which is clear when looking at the other metrics in the table. The increase in delivery cost and average hop count are correlated, because if a message passes through more nodes it is transferred more times. These values are highest for the popularity version, but given the size of a message in an environment such as the one from our trace experiment, they are not so significant. On the other hand, the average latency grows by a maximum of about 14 hours, which may prove to be too big of a difference for such an algorithm to be feasible. Nevertheless, by analyzing the maximum latency from the table, we observe that it increases by an order of up to approximately 12 days. The explanation is that there are some nodes that very rarely interact with other internal nodes from the network, so the opportunity of delivering data to them comes at very large time intervals, thus increasing the average latency. By eliminating such nodes from the simulation, the latency values fall into acceptable ranges. Another way of solving this issue is to use the external nodes as message carriers when running the three DiBuBB versions. In a real-life situation, the number of participants in an opportunistic network represented by an academic environment is much larger, thus the chance of getting data to even these remote nodes in a timely manner greatly increases. It should also be noted that the scenario presented here assumes that nodes send messages to random destinations. However, in reality there is a greater chance that a node will send data to someone from its social circle or community.

6 Conclusions and Future Work

We have presented a social tracing experiment that took place at the Politehnica University of Bucharest in the winter of 2011, with the purpose of gathering contact information between mobile devices participating in an opportunistic network. We analyzed the traces and showed that the contact and inter-contact times follow approximate power law functions. We applied a community detection algorithm on the traces and compared the results obtained with the social network. The conclusion was that nodes with more social links belong to more communities from the perspective of *k*-CLIQUE and that the social and logical grouping of nodes are in direct correlation with their interactions.

We have also shown that, by including knowledge about the social relationships between nodes in an opportunistic routing algorithm, the hit rate can be significantly increased at low latency and hop count costs under certain conditions. This has been proven on the trace presented in the paper, by comparing it to the distributed implementation of BUBBLE Rap, DiBuBB. Further tests to cement these affirmations will be performed, where nodes will send data to nodes in their own community or social circle with a higher probability. As future work, we plan to implement an opportunistic data dissemination algorithm of our own, specialized for scenarios like the one presented in this paper (i.e. an academic environment). We consider that opportunistic social networks are the future of mobile communication, especially in a world with more and more content available and with a higher degree of connectivity between individuals. Therefore, having real world traces of human movement and knowing that social relationships govern human interaction are paramount to creating suitable routing and dissemination algorithms.

Acknowledgement. This work was partially supported by project “ERRIC - Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1”, ID: 264207, and by national project “TRANSYS Models and Techniques for Traffic Optimizing in Urban Environments”, Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

References

1. McNett, M., Voelker, G.M.: Access and mobility of wireless PDA users. SIGMOBILE Mob. Comput. Commun. Rev. 7, 55–57 (2003)
2. Henderson, T., Kotz, D., Abyzov, I.: The changing usage of a mature campus-wide wireless network. In: Proc. of the 10th Annual Int. Conf. on Mobile Computing and Networking, MobiCom 2004, pp. 187–201. ACM, New York (2004)
3. Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., Diot, C.: Pocket switched networks and human mobility in conference environments. In: Proc. of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, WDTN 2005, pp. 244–251. ACM, New York (2005)

4. CRAWDAD, http://crawdad.cs.dartmouth.edu/
5. Musolesi, M., Mascolo, C.: Designing mobility models based on social network theory. SIGMOBILE Mob. Comput. Commun. Rev. 11, 59–70 (2007)
6. Boldrini, C., Passarella, A.: HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships. Comput. Commun. 33, 1056–1074 (2010)
7. Karamshuk, D., Boldrini, C., Conti, M., Passarella, A.: Human mobility models for opportunistic networks. IEEE Comm. Magazine 49(12), 157–165 (2011)
8. Conti, M., Giordano, S., May, M., Passarella, A.: From opportunistic networks to opportunistic computing. Comm. Mag. 48, 126–139 (2010)
9. Hui, P., Crowcroft, J., Yoneki, E.: BUBBLE Rap: social-based forwarding in delay tolerant networks. In: Proc. of the 9th ACM Int. Symp. on Mobile ad Hoc Networking and Computing, MobiHoc 2008, pp. 241–250. ACM, New York (2008)
10. Nguyen, H.A., Giordano, S., Puiatti, A.: Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman). In: 2007 IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks, pp. 1–6 (2007)
11. Boldrini, C., Conti, M., Jacopini, J., Passarella, A.: HiBOp: a History Based Routing Protocol for Opportunistic Networks. In: IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007, pp. 1–12 (2007)
12. Yoneki, E., Hui, P., Chan, S., Crowcroft, J.: A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In: Proc. of the 10th ACM Symp. on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWiM 2007, pp. 225–234. ACM, New York (2007)
13. Lenders, V., May, M., Karlsson, G., Wacha, C.: Wireless ad hoc podcasting. SIGMOBILE Mob. Comput. Commun. Rev. 12, 65–67 (2008)
14. Pop, F.: A fault tolerant decentralized scheduling in large scale distributed systems. In: Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (eds.) Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications. Info. Science Ref., pp. 566–589 (2010)
15. Boldrini, C., Conti, M., Passarella, A.: Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. Pervasive Mob. Comput. 4, 633–657 (2008)
16. Ciobanu, R., Dobre, C.: Data dissemination in opportunistic networks. In: 18th Int. Conf. on Control Systems and Computer Science, CSCS-18, pp. 529–536 (2011)
17. Bigwood, G., Rehunathan, D., Bateman, M., Henderson, T., Bhatti, S.: Exploiting self-reported social networks for routing in ubiquitous computing environments. In: Proc. of the 2008 IEEE Int. Conf. on Wireless & Mobile Computing, Networking & Comm., pp. 484–489. IEEE Computer Society, Washington, USA (2008)
18. Boldrini, C., Conti, M., Delmastro, F., Passarella, A.: Context- and social-aware middleware for opportunistic networks. J. Netw. Comput. Appl. 33(5), 525–541 (2010)
19. Social Tracer, http://code.google.com/p/social-tracer/
20. Ferro, E., Potorti, F.: Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. IEEE Wireless Comm. 12(1), 12–26 (2005)
21. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge, Computer Lab (2005)
22. Hui, P., Yoneki, E., Chan, S.Y., Crowcroft, J.: Distributed community detection in delay tolerant networks. In: Proc. of 2nd ACM/IEEE Inter. Workshop on Mobility in the Evolving Internet Architecture, MobiArch 2007, pp. 7:1–7:8. ACM, New York (2007)

Analysing Delay-Tolerant Networks with Correlated Mobility

Mikael Asplund and Simin Nadjm-Tehrani

Department of Computer and Information Science
Linköping University
{mikael.asplund,simin-nadjm.tehrani}@liu.se

Abstract. Given a mobility pattern that entails intermittent wireless ad hoc connectivity, what is the best message delivery ratio and latency that can be achieved for a delay-tolerant routing protocol? We address this question by introducing a general scheme for deriving the routing latency distribution for a given mobility trace. Prior work on determining latency distributions has focused on models where the node mobility is characterised by independent contacts between nodes. We demonstrate through simulations with synthetic and real data traces that such models fail to predict the routing latency for cases with heterogeneous and correlated mobility. We demonstrate that our approach, which is based on characterising mobility through a colouring process, achieves a very good fit to simulated results also for such complex mobility patterns.

Keywords: Latency, Delay-tolerant networks, Correlated Mobility, Connectivity.

1 Introduction

Delay- and disruption-tolerant networks represent an extreme end of systems in which a connected network cannot be relied upon. Instead, messages are propagated using a store-carry-forward mechanism. Such networks can have applications for disaster area management [4], vehicular networks [19], and environmental monitoring [17]. These systems offer many challenges and have been extensively studied by the research community [1,22,23,26].

Recent results indicate that to the extent that delay-tolerant networks will be found on a larger scale, they will definitely be composed of islands of connectivity, that is, some parts that are well-connected and some parts that are sparse. This in turn implies correlated contact patterns [2,11]. Most existing analytical delay performance models fail to capture such scenarios, since they assume independent node contacts. Moreover, although there are analyses done also for quite complex mobility models [8,9], it is not obvious how one should go about to map such models from real traces.

We extend previous results by studying the routing latency distribution for heterogeneous mobility movements. Our analytical model incorporates a colouring technique for information propagation to derive the latency distribution for

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 83–97, 2012.

an epidemic routing algorithm for a quite general case. The key strength of our approach compared to other models of heterogeneous mobility is that we are able to extract the relevant data from a real trace and produce the routing latency distribution (not just expected latency). The results are verified with a simulation-based study where we consider both synthetic and real-life mobility traces. We show that while a model that assumes independent inter-contact times works well for simple synthetic models such as random waypoint it is not able to predict the routing performance for a heterogeneous mobility model whereas our analytical results match very well.

There are two main contributions in this paper. First, a scheme for deriving the routing latency distribution for complex heterogeneous mobility models and, second, an experimental evaluation and validation of our model and a comparison with a model that assumes homogeneous and independent mobility is presented. The key insight of the evaluation is that heterogeneous mobility can result in such a high correlation of contacts that theoretical results based on independent inter-contact times are no longer valid.

The rest of the paper is organised as follows. Section 2 describes the system model and the basic assumptions we make. Section 3 describes how to derive the routing latency distribution given knowledge of the colouring rate distribution. This latter distribution is discussed in Section 4, and we explain how it can be determined from mobility traces. Section 5 contains the experimental evaluation. Finally, Section 6 gives an overview of the related work and Section 7 concludes the paper.

2 System Model

Consider a system composed of N mobile nodes (some possibly stationary). Nodes can communicate when they are in contact[1] with each other. During the contact both nodes can send and receive messages. We focus on connection patterns and ignore effects of queueing and contentions. Moreover, since we are interested in intermittently connected networks, the time taken to transmit a message is assumed negligible in relation to the time taken to wait for new contacts. We call this assumption A.

We characterise the pattern with which contacts occur using a simple colouring process (similar to [22,23]). Note that the colouring does not necessarily correspond to message dissemination, and should be seen only as an indication of node contact patterns. The basic idea is that if node A is coloured and subsequently comes in contact with node B, then node B will also become coloured (if not already coloured).The only restriction we make on the contact pattern (and thereby on the mobility of the nodes) is that the *incremental colouring times* should be independent. More specifically, given a colouring process that has coloured i nodes, the time to colour one more node is independent from the time taken to colour the earlier i nodes. We call this assumption B. Note that

[1] A contact is defined by a start and an end time between which two nodes are within communication range.

Table 1. Notation

N	Number of nodes in the system	$P(X)$	Probability of X being true
T_i	Random variable, the time taken for a randomly chosen colouring process to colour i nodes	Δ_i	Random variable, the time taken for a randomly chosen colouring process to colour one more node given i coloured nodes
R	Random variable, the message delivery time	$f_i(t)$	PDF of the random variable T_i
$f_{\Delta i}(t)$	PDF of the random variable Δ_i	$F_i(t)$	CDF of the random variable T_i
$F_{\Delta i}(t)$	CDF of the random variable Δ_i	$F_R(t)$	CDF of the random variable R

this is a much weaker restriction on the set of allowed mobility models compared to assuming independent inter-contact times.

We use a number of random variables to describe the colouring and routing processes, Table 1 summarises the most important notation. PDF is an abbreviation for probability density function and CDF stands for cumulative density function, these abbreviations are used throughout the paper.

Our analysis builds on ideal epidemic routing since it corresponds to the optimal performance any routing algorithm can achieve. Thus, these results provide a useful theoretical reference measure on what is good performance for a given mobility model. Such a reference can also be of practical use to decide whether the measured performance in some network is due to the network characteristics or to the protocol implementation. Moreover, this scheme can be extended to other routing protocols, for example using the techniques described by Resta and Santi [22].

3 Routing Latency

We now proceed to characterise the routing latency for epidemic routing in intermittently connected networks. We begin by determining the colouring time distribution which is then used to express the routing latency distribution.

3.1 Colouring Time

A colouring process (t_0, s) is characterised by a start time t_0 and a source node s from which the colouring process begins (thus, s becomes coloured at time t_0). Every time a coloured node comes in contact with an uncoloured node, the uncoloured node becomes coloured. Let T_i denote the random variable representing the time taken for a randomly chosen colouring process to colour i nodes.

Moreover, we let Δ_i denote the random variable that describes the time taken for a randomly chosen colouring process to colour *one more node* given that i nodes are already coloured. This means that we can express the time taken for a colouring process to reach $i+1$ nodes as $T_{i+1} = T_i + \Delta_i$.

Note that since we start the process with one coloured node, the time to colour the first node is $T_1 = 0$, and the time to colour the second node is $T_2 = \varDelta_1$. Slightly abbreviating standard notation we let $f_i(t)$ denote the PDF of the random variable T_i and let $f_{\varDelta i}(t)$ be the PDF of $\varDelta_i$. For the purpose of this presentation we assume that the latter of these functions is given since it depends on the mobility of the nodes in the system. In Section 4 we show how to extract $f_{\varDelta i}(t)$ from an existing contact trace. Assumption B from Section 2 states that T_i and $\varDelta_i$ are independent, so the PDF of their sum can be expressed as the convolution of the PDFs of the respective variables [10]:

$$f_{i+1}(t) = (f_i * f_{\varDelta i})(t) \tag{1}$$

Since we know the characteristic of $f_2(t)$ we can use equation (1) to iteratively calculate $f_3(t), f_4(t), f_5(t)$ and so on. The CDF for the variable T_i, here denoted by $F_i(t)$, can be computed in the standard manner from the PDF by integrating over all time points. Thus, assuming that colouring times are independent, it is straightforward to express the colouring time distribution $F_i(t)$ given knowledge of the PDF $f_{\varDelta i}(t)$. In the next subsection, we show how to derive the routing latency distribution from $F_i(t)$.

3.2 Routing Latency and Delivery Ratio

Our aim now is to find the latency distribution for an ideal routing algorithm. So, consider a randomly chosen time t_0, source node s and destination node $d \neq s$. Let R be the random variable that models the time to route a message from s to d using ideal epidemic routing. We will try to find the CDF of R, $F_R(t) = P(R \leq t)$. Clearly, given assumption A (i.e., that the queueing and transmission times can be neglected), this probability is the same as for d being one of the coloured nodes by the colouring process (t_0, s) after t time units.

Let C_t be the random variable that models the number of coloured nodes after t time units. If $C_t = i$ then the probability that d is coloured after t time units is $(i-1)/(N-1)$ since if we remove the source node s, there are $i-1$ coloured nodes and $N-1$ nodes in total. Thus, we can express $F_R(t)$ as:

$$F_R(t) = P(R \leq t) = \sum_{i=1}^{N} P(C_t = i) \cdot \frac{i-1}{N-1} \tag{2}$$

Now let's consider the probability $P(C_t = i)$ that the number of coloured nodes at time t equals i. This is the same as the probability that the time taken to inform i nodes is less than or equal to t minus the probability that $i+1$ nodes can be reached in this time:

$$P(C_t = i) = P(T_i \leq t) - P(T_{i+1} \leq t) \tag{3}$$

Combining equations (2) and (3), and rewriting gives:

$$F_R(t) = \frac{1}{N-1} \sum_{i=2}^{N} F_i(t) \tag{4}$$

Listing 1. GetRoutingLatencyDistribution

Input: $f_{\Delta i}$: Vector representing the PDF of Δ_i

```
$f_2 \leftarrow f_{\Delta 1}$
for i = 3 ... N
    $f_i \leftarrow \textsc{conv}(f_{i-1}, f_{\Delta i-1})$   /* equation (1) */
$F_i \leftarrow \textsc{cumsum}(f_i)$
$F_R \leftarrow \frac{1}{N-1}\sum_{i=2}^{N} F_i$              /* equation (4) */
return $F_R$
```

In summary, if we know the probability PDFs of the random variables Δ_i, we can use equation (1) to determine $f_i(t)$. Equation (4) will then give us the cumulative distribution function for the epidemic routing latency. Listing 1 shows an algorithmic representation of how to derive the distribution for R using discrete distributions. The procedure CONV and CUMSUM are standard Matlab functions and compute the convolution between two vectors and cumulative vector sum respectively.

By knowing R we can easily deduce the delivery ratio of a protocol given a certain time-to-live (TTL) for each packet. The probability that a message with TTL of T will reach its destination is simply $F_R(T)$ (i.e., the probability that the message will be delivered within time T).

4 Colouring Rate

Having derived the routing latency distribution based on knowledge of the distribution of the incremental colouring time Δ_i we now proceed to show how to find this latter distribution.

We consider two cases, when the mobility is homogeneous, and the more interesting heterogeneous case. By homogeneous we mean that the pairwise inter-contact times (i.e., the time between contacts) are identical and independently distributed (often abbreviated iid). The homogeneous case is not really novel in this context and is provided here briefly in order to explain the baselines we have used and to show that this case is also covered by our general approach.

4.1 Homogeneous Mobility

For the particular case of homogeneous mobility we make three additional assumptions commonly used to analyse homogeneous mobility [7,12]. (H1) The duration of contacts is negligible compared to the waiting times, (H2) the inter-contact time has a finite expectation, and (H3) pair-wise contacts are independent.

Now consider a set of coloured nodes that wait for a new contact to appear so that a new node can become coloured. The time they have to wait is the smallest of all pairwise waiting times for all pairs where one node in the pair is coloured

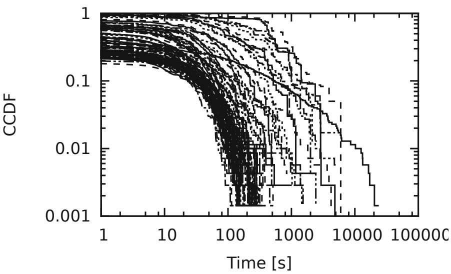


Fig. 1. Complementary Cumulative Distribution Functions (CCDF) of Δ_i

and one node is uncoloured. If i nodes are coloured, then there are $i(N-i)$ such pairs. Given assumption H3, we can express the CDF of Δ_i as:

$$F_{\Delta i}(t) = P(\Delta_i \leq t) = 1 - (1 - F_\tau(t))^{i(N-i)} \tag{5}$$

where $F_\tau(t)$ is the cumulative distribution of the residual[2] inter-contact time between two nodes. We refer to Karagiannis et al. [12] for further explanation and how to derive the residual distribution from the inter-contact distribution. If the inter-contact time is exponentially distributed with rate λ, then the residual waiting time is also exponentially distributed with the same rate and the incremental colouring time Δ_i will be exponentially distributed with rate $\lambda i(N-i)$.

4.2 Heterogeneous Mobility

If node contacts are not independent, then deriving an expression for the colouring distribution Δ_i will be more challenging. We now proceed to present a first simple model for approximating it from real heterogeneous traces.

In order to explain the rationale behind the model we first show some data from a real-life trace based on the movement of taxis in the San Francisco area. The trace was collected by Piorkowski et al. [21] based on data made available by the cabspotting project during May 2008 and we used a subset of the first 100 vehicles from the trace. In the simulation each taxi was assumed to have a wifi device with a range of 550m.

Fig. 1 shows the Complementary Cumulative Distribution Function (CCDF) of each Δ_i (recall that i corresponds to the number of already coloured nodes) for the San Fransisco cab scenario. We obtained this data by running 700 of colouring processes on the contact trace and logging the time taken to colour the next node. The plot uses a logarithmic scale on both axes to highlight the characteristics of the distribution. This shows that they exhibit an exponential decay (i.e., it approaches 0 fast, indicated by the sharp drop of the curves.).

[2] The residual inter-contact time refers to the time left to the next contact from a randomly chosen time t, as opposed to the time to the next contact measured from the previous contact time.

The second phenomena that we have observed is that due to clustering of nodes, it is often the case that the next node can be coloured without any waiting time at all. Based on these two basic principles we conjecture that the colouring time can be modelled as either being zero with a certain probability, or with a waiting time that is exponentially distributed.

If i nodes have been coloured, then we let $Con(i)$ denote the probability that one of those i nodes is connected to an uncoloured node (thereby allowing an immediate colouring of the next node). Further we let $f_{\mathrm{Exp}}(t, \lambda_i)$ denote the PDF of the exponential distribution with rate λ_i. Then, we let the PDF of the the simple colouring distribution model be expressed as:

$$f_{\Delta i} = \begin{cases} Con(i) & \text{if } t = 0 \\ (1 - Con(i)) * f_{\mathrm{Exp}}(t, \lambda_i) & \text{otherwise} \end{cases} \tag{6}$$

While this is clearly a simple model, it can be seen as a first step towards modelling the colouring distribution and seems to work well enough for the scenarios we have studied. We believe that further work is needed to better understand how the colouring distribution is affected by different mobility conditions. Note also that our general scheme is not tied to this particular model and allows further refinements.

5 Evaluation

To validate our model and to test whether it actually provides any added value compared to existing models we performed a series of simulation-based experiments. We used three different mobility models, the random waypoint mobility model, a model based on a map of Helsinki and a real-world trace from the cabs in the San Francisco area. After explaining the experiment setup we give the details and results for each of these models. Finally, we relate our findings on the effects of heterogeneity for these cases.

We used the ONE Simulator [14] to empirically find the ideal epidemic routing latency distribution for the three different mobility models. For each mobility model we ran the simulation 50 times. For the first 40000 seconds a new message with random source and destination was sent every 50 to 100 seconds. The simulation length was sufficiently long for all messages to be delivered. We used small messages of size 1 byte, and channel bandwidth of 10Mb/s.

In addition to the simulated results we used two different theoretical models to predict the latency distribution:

Colouring Rate: This model uses equation (6) from Section 4.2 to model the colouring times. The necessary parameters $Con(i)$ and λ_i are estimated from the trace file by sampling.

Homogeneous: This model assumes independent and exponentially[3] distributed inter-contact times which are used to compute $f_{\Delta i}$ as described in Section 4.1. This has been a popular model for analysing properties of delay-tolerant networks [22,23,26].

In order not to get a biased value for the inter-contact time distributions due to a too short sampling period, we analysed contacts from 200 000 seconds of simulation. To further reduce the effect of bias we use Kaplan-Meier estimation as suggested by Zhang et al. [26].

5.1 Effect of Mobility

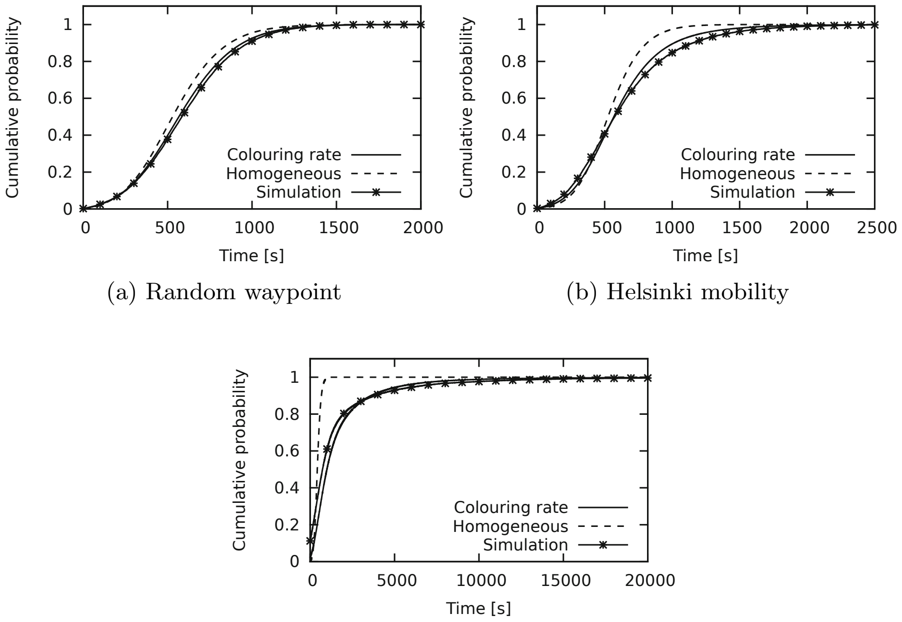
Random Waypoint Mobility. In order to validate our model against already known results, we start with considering the random waypoint mobility model. Despite its many weaknesses [2,25], this model of mobility is still very popular model for evaluating ad hoc communication protocols and frameworks. The network was composed of 60 nodes moving in an area of $5km \times 5km$, each having a wireless range of $100m$. The speed of nodes was constant $10m/s$ with no pause time.

Fig. 2a shows the results of the two theoretical models and the simulation. The graph shows the cumulative probability distribution (i.e., the probability that a message will has been delivered within the time given on the x axis). As expected, both models manage to predict the simulated results fairly well. In fact, the exponential nature of the inter-contact times of RWP is well understood and since the heterogeneous model is more general, we were expecting similar results.

Helsinki Mobility. We now turn to a more realistic and interesting mobility model, the Helsinki mobility model as introduced by Keränen and Ott [13]. The model is based on movements in the Helsinki downtown area. The 126 nodes is a mix of pedestrians, cars, and trams, and the move in the downtown Helsinki area (4500x3400 m). We used a transmission range of 50 meters for all devices. Fig. 2b shows the results. Again both theoretical models achieve reasonable results. However, due to the partly heterogeneous nature of the mobility model, the homogeneous model differs somewhat more from the simulated result. In particular, we see that the s-shape is more sharp compared to the observed data. We further discuss possible explanations for this in Section 5.2.

San Francisco Cabs. Finally, the last mobility trace we have analysed is a real-life trace based on the movement of taxis in the San Francisco area as explained in Section 4.2. Fig. 2c shows the results. In this case the homogeneous model fails to capture the routing latency that can be observed in simulation. However, the heterogeneous model based on equation (6) is still quite accurate. We were surprised to find such a big difference between the simulated data and the homogeneous model. Something is clearly very different in this trace compared to the synthetic mobility models. An estimate of the fraction of messages

[3] We also obtained nearly identical results when estimating the inter-contact distribution from the mobility trace, which we have excluded for lack of space.



(a) Random waypoint

(b) Helsinki mobility

(c) San Francisco cab trace

Fig. 2. Routing latency

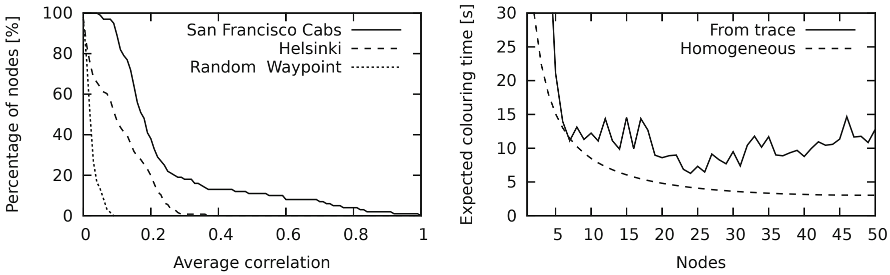
being delivered within an average latency of 2500s in such a scenario would be misleadingly optimistic by 20%.

5.2 The Effects of Heterogeneity

In the previous subsection we have seen that the accuracy of the homogeneous model is high for the random waypoint model, but is lower for the Helsinki model and completely fails for the San Francisco cab trace. In this subsection we present our investigation into why this is the case. We proceed by identifying four different aspects of how this model differs from reality.

Correlation. We begin with the most striking fact of the results presented so far. The homogeneous model is way off in predicting the routing latency distribution in the San Francisco case. There are a number of different ways that one can try to explain this, but we believe that the most important one has to do with correlation (i.e., non-independence) of events. The main assumption that makes equation (5 possible, and thereby the homogeneous model is that the contacts between different pairs of node are independent from each other. However, this seems to be a false assumption.

We analysed the contact patterns of the three different mobility models by considering the residual inter-contact times for each node during a period of 20000 seconds. Fig. 3a shows the percentage of nodes who's average correlation



(a) (b)

Fig. 3. (a) Correlation of contacts, (b) Time to colour one more node in the San Francisco trace

among its contacts is higher than a given value (i.e., it is the complementary CDF of nodes having a given average correlation). If the pairwise contacts are independent, they will have no (or very low) average correlation and we would expect to see a sharp decay of the curve in the beginning of the graph. This is also what we see for the random waypoint model. Since the nodes move around completely independently from each other, the contacts also become independent. The Helsinki trace shows a higher degree of correlation, but not as significant as for the San Francisco cab case. In this case 40% of the nodes have an *average* correlation of their contacts which is higher than 0.2 (a correlation of 1 would mean that all contacts are completely synchronised). This shows a high degree of dependence and we believe provides an explanation of the result we have seen in Section 5.1.

Note that correlated mobility does not necessarily lead to slower message propagation, in fact there are results indicating the contrary [8]. What we have seen is that the *prediction* of the latency becomes too optimistic when not taking correlation into account. If the model assumes that contacts are "evenly" spread out over time, whereas in reality they come in clusters, the results of the model will not be accurate.

Lack of Expansion. The second prominent effect is what we choose to call lack of expansion (motivated by the close connection to expander graphs [3]). This means that the rate of the colouring process seems not to correspond to the number of coloured nodes. Fig. 3b shows the expected time to colour one more node for the San Francisco trace. The x-axis represents the number of nodes already coloured (up to half the number of nodes). We can see that the homogeneous model predicts that the time decreases (i.e., the rate of colouring increases) as the number of coloured nodes increase. On the other hand, the data based on sampling the distribution of Δ_i from the mobility trace file (indicated as "From trace" in the figure) shows that after the first 5-10 nodes have been coloured, the rate is more or less independent from the number of coloured nodes. We believe that this is partly due to the fact that most of the node mobility is relatively local and that nodes are often stationary for long periods of time.

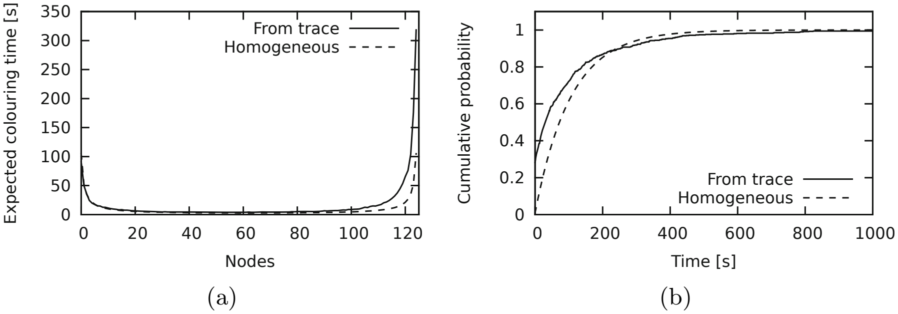


Fig. 4. (a) Time to colour one more node with the Helsinki mobility model, (b) CDF of the time to colour the second node

Slow Finish. Another effect that can be observed is that in some rare cases it can take a very long time for a message to reach its destination. For example in Fig. 2c, even after 10000 seconds not all messages have been delivered to their destinations. This has to do with the fact that the time to colour all nodes take significantly longer time than to colour *almost* all the nodes. The models based on independent contacts predict that it takes the same amount of time to colour the second node as it takes to colour the last node. In both cases there are $N - 1$ possible node pairs that can meet and result in a colouring. However, we have seen that in reality colouring the last node takes significantly longer (on average). Fig. 4a shows the effect for the Helsinki trace, by plotting the expected colouring time as a function of the number of coloured nodes. While the homogeneous model is completely symmetrical around the middle, the actual data shows that it takes roughly three times longer to reach the last node than to reach the second node.

Fast Start. Finally, we consider why the homogeneous model predict a lower probability for delivering messages fast. This can be seen in both the Helsinki and San Francisco cases, but is more distinct in the former case. It can be seen visually in Fig. 2b in that the homogeneous model has a slightly flatter start compared to the other curves. This is because there is a chance that when a message is created, the node at which it is created has a number of neighbours. Thus, the message will not need to wait any time at all before being transmitted. Or if we express it as a colouring process, the time to colour the second node is sometimes zero. For a model based on inter-contact times, this is not considered.

Fig. 4b shows the CDF of T_2, (i.e., the time taken to colour the second node) for the Helsinki case with the colouring rate and homogeneous model. We see that both curves are similar (the expected value for T_2 is the same for both models) but that the start value differs. That is, in the homogeneous model, it is predicted that the chance that the second node is immediately coloured is zero, whereas in fact it is roughly 0.3. Recall that the colouring time only reflects the contact patterns of the mobility and does not consider message transmission delays.

In this section we have seen how heterogeneous mobility causes correlated contacts and how that affects predictions of routing latency. Our model which is based on colouring rate of nodes was the only model able to accurately predict the routing latency distribution in these cases.

6 Related Works

There is a rich body of work discussing detailed analytical models for latency and delivery ratio in delay-tolerant networks. The work ranges from experimentally grounded papers aiming to find models and frameworks that fit to observed data to more abstract models dealing with asymptotic bounds on information propagation. Many of these approaches are based on or inspired by epidemiological models [15]. We have previously characterised the worst-case latency of broadcast for such networks using expander graph techniques [3].

Closest to our work in this paper is that of Resta and Santi [22], where the authors present an analytical framework for predicting routing performance in delay-tolerant networks. The authors analyse epidemic and two-hops routing using a colouring process under similar assumptions as in our paper. The main difference is that our work considers heterogeneous node mobility (including correlated inter-contact times), whereas the work by Resta and Santi assumes independent exponential inter-contact times.

Zhang et al. [26] analyse epidemic routing taking into account more factors such as limited buffer space and signalling. Their model is based on differential equations also assuming independent exponentially distributed inter-contact times. A similar technique is used by Altman et al. [1], and extended to deal with multiple classes of mobility movements by Spyropoulos et al. [24].

Kuiper and Nadjm-Tehrani [16] present a quite different approach for analysing performance of geographic routing. Their framework can be used based on abstract mobility and protocol models as well as extracting distributions for arbitrary mobility models and protocols from simulation data. The main application area for this model is geographic routing where waiting and forwarding are naturally the two modes of operation in routing.

The assumption of exponential inter-contact times was first challenged by Chaintreau et al. [7] who observed a power law of the distribution for a set of real mobility traces (i.e., meaning that there is a relatively high likelihood of very long inter-contact times). Later work by Karagiannis et al. [12] as well as Zhu et al. [27] showed that the power law applied only for a part of the distributions and that from a certain time point, the exponential model better explains the data. Pasarella and Conti [20] present a model suggesting that an aggregate power law distribution can in fact be the result of pairs with different but still independent exponentially distributed contacts. Such heterogeneous but still independent contact patterns have also been analysed in terms of delay performance by Lee and Eun [18].

Our work on the other hand, suggests that the exact characteristic of the inter-contact distribution is less relevant when contacts are not independent.

Correlated and heterogeneous mobility and the effect on routing have recently been discussed in several papers [6,5,8,11], but to our knowledge, we are the first to provide a framework that accurately captures the routing latency distribution for real traces with heterogeneous and correlated movements.

7 Conclusions and Future Work

We have presented a mathematical model for determining the routing latency distribution in intermittently connected networks based on trace analysis. The basic idea that we have built upon is that the speed of a colouring process captures the dynamic connectivity of such networks. This was confirmed by a set of simulation-based experiments where we demonstrated that our model matched the simulation results very well. On the other hand, the models based on independent and homogeneous contacts did not provide accurate results except for the case with the random waypoint mobility model.

Our scheme allows accurate analysis of a much wider range of mobility models than previously possible. This analytical technique also has the possibility to increase our understanding of the connection between mobility and routing performance, potentially leading to new mobility metrics and classifications. We used a rough estimation-based model for the colouring distribution, and there is certainly room for considering other ways of expressing these distributions.

There are several possible extensions to this work. First, it would be interesting to study the accuracy of the analysis in the context of other routing paradigms such as social and geographic routing, as well as considering effects of limited bandwidth and buffers. Moreover, the effects of correlation of node contacts should be further investigated by analysing other real-life traces, also considering under which circumstances our assumption of independent colouring times is valid.

Acknowledgements. This work was supported by the Swedish Research Council (VR) grant 2008-4667. During the final stages of preparing the manuscript the first author was supported, in part, by Science Foundation Ireland grant 10/CE/I1855.

References

1. Altman, E., Basar, T., Pellegrini, F.D.: Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks. Perform. Eval. 67(4) (2010), doi:10.1016/j.peva.2009.09.001
2. Aschenbruck, N., Munjal, A., Camp, T.: Trace-based mobility modeling for multi-hop wireless networks. Comput. Commun. 34(6) (2010), doi:10.1016/j.comcom.2010.11.002
3. Asplund, M.: Disconnected Discoveries: Availability Studies in Partitioned Networks. PhD thesis, Linköping University (2010), http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-60553

4. Asplund, M., Nadjm-Tehrani, S.: A partition-tolerant manycast algorithm for disaster area networks. In: 28th International Symposium on Reliable Distributed Systems (SRDS). IEEE (2009), doi:10.1109/SRDS.2009.16
5. Bulut, E., Geyik, S., Szymanski, B.: Efficient routing in delay tolerant networks with correlated node mobility. In: 2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems, MASS (2010), doi:10.1109/MASS.2010.5663962
6. Cai, H., Eun, D.Y.: Toward stochastic anatomy of inter-meeting time distribution under general mobility models. In: Proceedings of the 9th ACM International Symposium on Mobile ad hoc Networking and Computing, MobiHoc 2008. ACM (2008), doi:10.1145/1374618.1374655
7. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on opportunistic forwarding algorithms. IEEE Trans. Mobile Comput. 6(6) (2007), doi:10.1109/TMC.2007.1060
8. Ciullo, D., Martina, V., Garetto, M., Leonardi, E.: Impact of correlated mobility on delay-throughput performance in mobile ad hoc networks. IEEE/ACM Transactions on Networking 19(6) (2011), doi:10.1109/TNET.2011.2140128
9. Garetto, M., Giaccone, P., Leonardi, E.: Capacity scaling in delay tolerant networks with heterogeneous mobile nodes. In: Proc. 8th ACM International Symposium on Mobile ad hoc Networking and Computing (MobiHoc). ACM (2007), doi:10.1145/1288107.1288114
10. Grinstead, C.M., Snell, J.L.: Introduction to Probability. American Mathematical Society (1997)
11. Hossmann, T., Spyropoulos, T., Legendre, F.: Putting contacts into context: Mobility modeling beyond inter-contact times. In: Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2011). ACM (2011)
12. Karagiannis, T., Boudec, J.-Y.L., Vojnović, M.: Power law and exponential decay of intercontact times between mobile devices. IEEE Transactions on Mobile Computing 9 (2010), doi:10.1109/TMC.2010.99
13. Keränen, A., Ott, J.: Increasing reality for dtn protocol simulations. Technical report, Helsinki University of Technology, Networking Laboratory (2007)
14. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for dtn protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools, ICST (2009), doi:10.4108/ICST.SIMUTOOLS2009.5674
15. Khelil, A., Becker, C., Tian, J., Rothermel, K.: An epidemic model for information diffusion in manets. In: Proc. 5th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM). ACM (2002), doi:10.1145/570758.570768
16. Kuiper, E., Nadjm-Tehrani, S., Yuan, D.: A framework for performance analysis of geographic delay-tolerant routing. EURASIP Journal on Wireless Communications and Networking (to appear, 2012)
17. Lahde, S., Doering, M., Pöttner, W.-B., Lammert, G., Wolf, L.: A practical analysis of communication characteristics for mobile and distributed pollution measurements on the road. Wireless Communications and Mobile Computing 7(10) (2007), doi:10.1002/wcm.522
18. Lee, C.-H., Eun, D.Y.: Exploiting heterogeneity in mobile opportunistic networks: An analytic approach. In: 2010 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks, SECON (2010), doi:10.1109/SECON.2010.5508265

19. Lu, R., Lin, X., Shen, X.: Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In: 2010 Proceedings of IEEE INFOCOM (2010), doi:10.1109/INFCOM.2010.5462161
20. Passarella, A., Conti, M.: Characterising Aggregate Inter-contact Times in Heterogeneous Opportunistic Networks. In: Domingo-Pascual, J., Manzoni, P., Palazzo, S., Pont, A., Scoglio, C. (eds.) NETWORKING 2011, Part II. LNCS, vol. 6641, pp. 301–313. Springer, Heidelberg (2011), doi:10.1007/978-3-642-20798-3_23
21. Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M.: A parsimonious model of mobile partitioned networks with clustering. In: First International Conference on Communication Systems and Networks (COMSNETS). IEEE (2009), doi:10.1109/COMSNETS.2009.4808865
22. Resta, G., Santi, P.: A framework for routing performance analysis in delay tolerant networks with application to non cooperative networks. IEEE Transactions on Parallel and Distributed Systems 23(1) (2011), doi:10.1109/TPDS.2011.99
23. Spyropoulos, T., Psounis, K., Raghavendra, C.: Efficient routing in intermittently connected mobile networks: The single-copy case. IEEE/ACM Trans. Netw. 16(1) (2008), doi:10.1109/TNET.2007.897962
24. Spyropoulos, T., Turletti, T., Obraczka, K.: Routing in delay-tolerant networks comprising heterogeneous node populations. IEEE Trans. Mobile Comput. 8(8) (2009), doi:10.1109/TMC.2008.172
25. Yoon, J., Liu, M., Noble, B.: Random waypoint considered harmful. In: Proc. INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE (2003), doi:10.1109/INFCOM.2003.1208967
26. Zhang, X., Neglia, G., Kurose, J., Towsley, D.: Performance modeling of epidemic routing. Comput. Netw. 51(10) (2007), doi:10.1016/j.comnet.2006.11.028
27. Zhu, H., Fu, L., Xue, G., Zhu, Y., Li, M., Ni, L.: Recognizing exponential inter-contact time in vanets. In: Proceedings of IEEE INFOCOM (2010), doi:10.1109/INFCOM.2010.5462263

Study on the Effect of Network Dynamics on Opportunistic Routing

Waldir Moreira[1], Manuel de Souza[1], Paulo Mendes[1], and Susana Sargento[2]

[1] SITI, University Lusófona
{waldir.junior,manuel.desouza,paulo.mendes}@ulusofona.pt
[2] Instituto de Telecomunicações, University of Aveiro
susana@ua.pt

Abstract. There has been an effort to employ social similarity inferred from user mobility patterns in opportunistic routing solutions to improve forwarding. However, the dynamics of the networks are still not fully considered when devising solutions based on social similarity metrics. To address this issue, we propose two utility functions which consider the daily life routines of users and the intensity of their social interactions to take forwarding decisions: *Time-Evolving Contact Duration* (TECD) that weights social interactions among nodes considering the duration of contacts; and *TECD Importance* (TECDi) which estimates the importance of nodes. We compare our utility functions against contact- and social-based solutions, and we show that the use of daily life routines information (i.e., using *TECD* and *TECDi*) has a positive effect on opportunistic routing.

Keywords: daily routines, network dynamics, contact duration, social structures, opportunistic routing.

1 Introduction

Most of the proposed opportunistic routing solutions are built based on inter-contact times [1], despite the increasing need to fully understand the nature of such statistics. In addition, the resulting proximity graphs are rather instable [2] as they need to be created based on the users' mobility, which constantly changes.

There is a current trend that employs the use of stable social structures (inferred from user mobility) in order to improve opportunistic routing [3]. These social structures are built based on social similarity metrics such as centrality which identifies nodes that can increase the probability of message delivery given their popularity within the system. Such centrality is used to perform message delivery both inside and outside of a cluster/community under the assumption that users meet each other according to the social strength between them.

However, the proposals which are based on the identification of social structures (e.g., communities) do not take into account the dynamics of networks, that is, the evolution of the network structure (the making and braking of network ties) since users meet other users in different moments during their daily

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 98–111, 2012.

routines. So, the global structure of the users' social network changes as their personal networks change.

When considering dynamic social similarity, Hossmann et al. [4] show that it is imperative to correctly map real node interactions (resulting from the mobility process) into a cleaner social representation (i.e., comprising only stable social contacts) to create proximity graphs based on the daily life routine of nodes to aid routing. This encourages us to study the effect that network dynamics have on opportunistic routing. Additionally, Eagle and Pentland [5] show that users have routines that can be used to identify future behavior and interaction with others with whom they share similar behavior and potentially share the same community. These studies suggest that opportunistic routing should mimic social behavior, and the creation of social structures should consider the oscillations of user behavior.

To address this challenge, we propose the use of time-evolving social structures to reflect the different behavior that users have in different daily periods of time. This is achieved through two novel utility functions: *Time-Evolving Contact Duration* (TECD) that weights social interactions based on the statistical contact duration that nodes have in the same daily period of time, over consecutive days; and *TECD Importance* (TECDi) which estimates the importance of nodes based on its node degree, and the social strength towards its neighbors, in different periods of time.

To evaluate our approach, we compare opportunistic routing based on our utility functions against benchmark contact- and social-based proposals. Results show that capturing the dynamism of networks based on social daily behavior (by using *TECD* and *TECDi*, for instance) can improve routing performance in terms of delivery probability, cost and latency.

This paper is structured as follows. Section 2 analyzes the most significant opportunistic routing trends. In Section 3, the *TECD* and *TECDi* utility functions are presented along with an algorithm used to implement them. Section 4 presents the evaluation methodology, setup, and results. In Section 5 we conclude the paper and present directions for future work.

2 Related Work

We previously identified [3] that most of the opportunistic routing prior-art considered the replication-based forwarding scheme, while only 15% were based on single-copy and flooding-based forwarding schemes. Among the replication-based solutions, approximately 69% consider a contact-based approach (e.g., frequency of encounters) and 31% (the latest ones) investigate a new trend based on social similarity metrics (e.g., community detection).

Contact-based proposals consider every contact among nodes to update the proximity graph and implement metrics such as the number of times nodes meet, contact frequency and the last time a contact occurs. Besides *PROPHET* [6], the most cited replication-based proposal [3], other examples based on contact metric are *Prediction* [7], and *Encounter-Based Routing* [8].

Social-based proposals build proximity graphs based mostly on social similarity metrics such as inter-contact times, and therefore they can identify social structures such as communities in *Bubble Rap* [2], interest shared by nodes as in *SocialCast* [9], and node popularity (i.e., importance) as in *PeopleRank* [10].

On one hand, contact-based proposals can achieve acceptable performance, but with the complexity of updating proximity graphs according to node movements [2]. On the other hand, social-based proposals have shown routing improvements based on proximity graphs more stable than those created based on every contact. However, these proposals are complex when subject to form communities prior to routing [2], or are based on strong assumptions about nodes sharing the same interests spending most of the time co-located [9,10].

Independently of being based on contacts or social similarities, none of the analyzed approaches consider the daily life routine of people (i.e., the dynamics of network) carrying communicating devices, which certainly influences their performance. Additionally, it has been shown that people's routines can be rather useful to determine future behavior [5], and that considering the dynamics of social ties (based on an analysis of contact duration) from different daily routines is important to achieve a correct mapping of real social interactions into a proximity graph with a clean (i.e., more stable) social representation able to aid data forwarding [4].

3 Our Approach

Forwarding based on social interactions has great potential as less volatile proximity graphs are created. However, existing approaches fail to consider the oscillations of social interactions, which are mostly influenced by daily routines. We believe that the accuracy level of social interactions is mainly dependent on the statistical duration of contacts over different periods of time, since people have daily habits that lead to a periodic repetition of behavior [5]. Moreover, they will more accurately reflect the real evolution of social ties than relying solely on the contact between nodes or well-defined social structures. This section starts by describing the *TECD* and *TECDi* utility functions. After that, we describe the algorithm that implements such utility functions.

3.1 Time-Evolving Contact Duration (TECD)

TECD aims to capture the evolution of social interactions in the same daily period of time (hereafter called daily sample) over consecutive days, by computing social strength based on the average duration of contacts.

Fig. 1 shows how social interactions (from the point of view of user A) vary during a day. For instance, it illustrates a daily sample (8 pm - 12 am) over which the social strength of user A to users D, E, and F is much stronger (less intermittent line) than the strength to users B and C. Fig. 1 aims to show the dynamics of a social network over a one-day period, where users' behavior in different daily samples lead to different social structures.

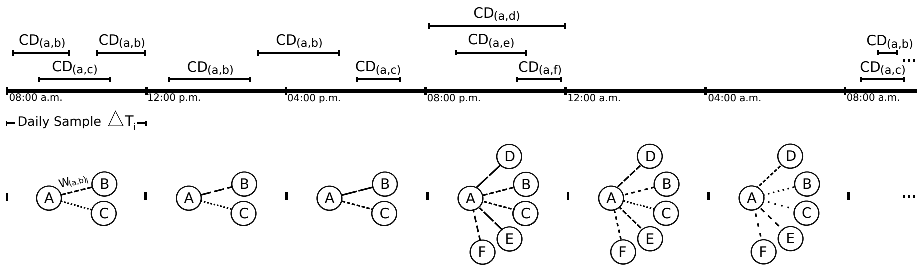


Fig. 1. Contacts of A with a set of nodes x ($CD_{(a,x)}$) in different daily samples ΔT_i

As illustrated in Fig. 1, users' social strength in a given daily sample depends on the average contact duration that they have in such time period: if user x has n contacts with user y in a daily sample ΔT_i, having each contact k a certain duration (*Contact Duration* - $CD_{(x,y)_k}$), at the end of ΔT_i the *Total Contact Time* ($TCT_{(x,y)_i}$) between them is given by Eq. 1.

$$TCT_{(x,y)_i} = \sum_{k=1}^{n} CD_{(x,y)_k} \tag{1}$$

Each ΔT_i represents a different daily sample in the routine of a person. Since a behavior pattern can be observed, we can consider that each daily sample represents a specific behavior at work/study place, home, or somewhere else (e.g., out of town, friends' houses). In [5] it is shown that people can have their future behaviors predicted by considering previous ones. Thus, we try to capture such behavior considering the time that nodes spend together (i.e., *Total Connected Time*) in the same daily sample ΔT_i along different days j.

The *Total Contact Time* between users in the same daily sample over consecutive days can be used to estimate the average duration of their contacts for that specific daily sample [5]: the average duration of contacts between users x and y during a daily sample ΔT_i in a day j ($AD^j_{(x,y)_i}$) is given by a cumulative moving average of their TCT in that daily sample ($TCT^j_{(x,y)_i}$) and the average duration of their contacts during the same daily sample ΔT_i in the previous day (cf. Eq. 2)

$$AD^j_{(x,y)_i} = \frac{TCT^j_{(x,y)_i} + (j-1)AD^{j-1}_{(x,y)_i}}{j} \tag{2}$$

The social strength between users in a specific daily sample may also provide some insight about their social strength in consecutive k samples in the same day, ΔT_{i+k}. This is what we call *Time Transitive Property.* This property increases the probability of nodes being capable of transmitting large data chunks, since transmission can be resumed in the next daily sample with high probability.

The *TECD* utility function (cf. Eq. 3) is able to capture the social strength ($w_{(x,y)_i}$) between any pair of users x and y in a daily sample ΔT_i based on the *Average Duration* ($AD_{(x,y)_i}$) of contacts between them in such daily sample and in consecutive $t-1$ samples, where t represents the total number of daily samples. When $k > t$, the corresponding $AD_{(x,y)}$ value refers to the daily sample $k-t$. In Eq. 3 the time transitive property is given by the weight $\frac{t}{t+\text{k-i}}$, where the highest weight is associated to the average contact duration in the current daily sample, being it reduced in consecutive samples.

$$TECD = w_{(x,y)_i} = \sum_{k=i}^{i+t-1} \frac{t}{t+k-i} AD_{(x,y)_k} \tag{3}$$

3.2 TECD Importance (TECDi)

TECDi, shown in Eq. 4, aims to capture the *Importance* (I_x^i) of any user x in a daily sample ΔT_i, based on its social strength ($TECD$) towards each user that belongs to its neighbor set (N_x) in that time interval, in addition to the importance of such neighbors.

$$TECDi = I_x^i = (1-d) + d \sum_{y \epsilon N_x} w_{(x,y)_i} \frac{I_y^i}{N_x} \tag{4}$$

TECDi is based on the *PeopleRank* function [10]. However, *TECDi* considers the social strength between a user and its neighbors encountered within a specific ΔT_i, while *PeopleRank* computes the importance considering all neighbors of x at any time. It is worth mentioning that the dumping factor (d) in *TECDi* has a similar meaning as in *PeopleRank:* to introduce some randomness while taking forwarding decisions.

3.3 Distributed Algorithm

The operation of the algorithm is quite simple as shown in Algorithm 1. Its functionality is different according to the utility function being used, *TECD* or *TECDi.* In the former case, when the $CurrentNode$ meets a $Node_i$ in a daily sample ΔT_k, it will get a list of all neighbors of $Node_i$ in that daily sample and its weights towards them, ($Node_i$.WeightsToAllneighbors computed based on Eq. 3). Then, every $Message_j$ in $CurrentNode$'s buffer is replicated to $Node_i$ if the latter's weight towards the destination (getWeightTo($Destination_j$)) is greater than $CurrentNode$'s weight towards the same destination.

If *TECDi* is used, then when the $CurrentNode$ meets a $Node_i$, the $CurrentNode$ obtains $Node_i$'s importance and replication occurs if such importance is greater than or equal to that of the $CurrentNode$.

Algorithm 1. Forwarding with *TECD* and *TECDi*

```
begin
  foreach $Node_i$ encountered by $CurrentNode$ do
    if ($TECD$ being used) then
      receive($Node_i$.WeightsToAllneighbors)
      foreach $Message_j \in$ buffer.($CurrentNode$) & $\notin$ buffer($Node_i$) do
        if ($Node_i$.getWeightTo($Destination_j$) >
                    $CurrentNode$.getWeightTo($Destination_j$))
        then $CurrentNode$.replicateTo($Node_i$, $Message_j$)
    else if ($TECD_i$ being used) then
      receive($Node_i$.Importance)
      foreach $Message_j \in$ buffer.($CurrentNode$) & $\notin$ buffer($Node_i$) do
        if ($Node_i$.importance $\geq$ $CurrentNode$.importance)
        then $CurrentNode$.replicateTo($Node_i$, $Message_j$)
end
```

4 Performance Evaluation

This section presents the evaluation methodology, implementations and simulation settings, and the evaluation results which point out advantages and constraints of capturing the dynamics of the network, by using time-evolving contact duration and node importance in opportunistic routing. Results are analyzed in two stages: we start by the performance analysis between the *TECD*-based algorithm and contact-based approaches, *PROPHET* and *Epidemic*. Next, we analyze the performance of the *TECDi*-based algorithm against social-based approaches, *Bubble* Rap and *Rank* (a *PeopleRank*-like solution). For the sake of simplicity, we refer to our algorithm as *TECD* or *TECDi* hereafter.

4.1 Evaluation Methodology

Performance analysis is carried out on Opportunistic Network Environment (ONE) [11] simulator with simulations representing a 12-day interaction period between nodes. Each simulation is run ten times (with different random number generator seeds for the used movement models) to provide a 95% confidence interval for the results.

The metrics used to assess the performance of *TECD* and *TECDi* against the contact- and social-based benchmark solutions are the average delivery probability (i.e., ratio between the number of delivered messages and total number of created messages), average cost (i.e., number of replicas per delivered message), and average latency (i.e., time elapsed between message creation and delivery). Regarding the ONE simulator, the time step size is of 2 seconds [11], and simulations are performed in batch mode with 2 GB RAM dedicated memory.

4.2 Implementations and Simulation Settings

TECD and *TECDi* are compared with representatives of the identified contact- and social-based approaches in Section 2. In what concerns contact-based solutions, *PROPHET* [6] and *Epidemic* [12] were chosen for being the most cited proposals: the first is widely recognized by the Delay Tolerant Networks (DTN) research community, and the second represents a solution that is merely interested in replicating messages in order to increase delivery. For the social-based approaches, *Bubble Rap* [2] and *PeopleRank* [10] were selected as representative of solutions based on social structures and node popularity notions.

Regarding the implementations, ONE encompasses all benchmarks except *Rank*. Hence, we have implemented it considering the node degree and the importance of its neighbors to make it having a *PeopleRank*-like behavior in each time interval of a daily routine. Thus, instead of determining the overall importance of a node at each encounter, like *PeopleRank* does, *Rank* estimates the node importance at the end of every daily sample.

It is worth noting that the dumping factor d of *Rank,* which is also used in *TECDi*, was set to 0.8, since it lies among the values where *PeopleRank* [10] showed the best success rates. Regarding *Bubble Rap*, we used it with K-clique and single window algorithms for community formation and node centrality computation [2]. We chose parameter k to be 5, since it results in the best overall performance for *Bubble Rap* in terms of delivery probability, cost and latency.

Regarding the simulation settings, although not using real traces, we aim at creating a scenario close to the one found in people's daily activities. Such concern is also considered in the evaluation of considered benchmarks where community-based scenarios are devised and real world traces are used.

The simulation scenario is part of the Helsinki city and has 150 nodes distributed in 8 groups of people and 9 groups of vehicles. One of the vehicle groups, with 10 nodes, follows the *Shortest Path Map Based Movement*, where they randomly choose a place and use the shortest path to reach it. These nodes represent police patrols equipped with Bluetooth (250 kbps in a 10 m range), with speed between 7 to 10 m/s and a pause time between 100 and 300 seconds.

The other vehicle groups represent buses. Each group is composed of 2 vehicles, equipped with Bluetooth (250 kbps/10 m) and WiFi (11 Mbps/250 m), following the *Bus Movement* with speeds between 7 to 10 m/s and pause times between 10 and 30 seconds. Due to limitations of the *Bubble Rap* implementation, the bus group is equipped with only WiFi (11 Mbps/100 m) interface.

The people groups follow the *Working Day Movement* with walking speeds between 0.8 and 1.4 m/s and may also use buses. Each group has different meeting spots, offices, and home locations. People spend 8 hours at work and present 50% probability of having an evening activity after work. In the office, nodes move around and have a pause time between 1 min and 4 hours. Evening activities can be done alone or in group, and can last between 1 and 2 hours.

The traffic load corresponds to approximately 500 messages generated per day among a subset of preset node pairs, which results in 6000 considered for the performance assessment.

TTL values were set at 24 hours and unlimited. With a 24-hour TTL, messages have a short time in the system and shall impact delivery probability, and unlimited TTL increases the cost as messages are allowed to be replicated for the duration of the simulations. This way the studied proposals are analyzed in two extreme cases. Message size ranges from 1 kB to 100 kB. The buffer space is of

2 MB. Message and buffer size comply with the universal evaluation framework that we proposed previously [3] based on the evidence that opportunistic routing prior-art follow completely different evaluation settings, making assessment a challenging task.

4.3 Comparison against Contact-Based Algorithms

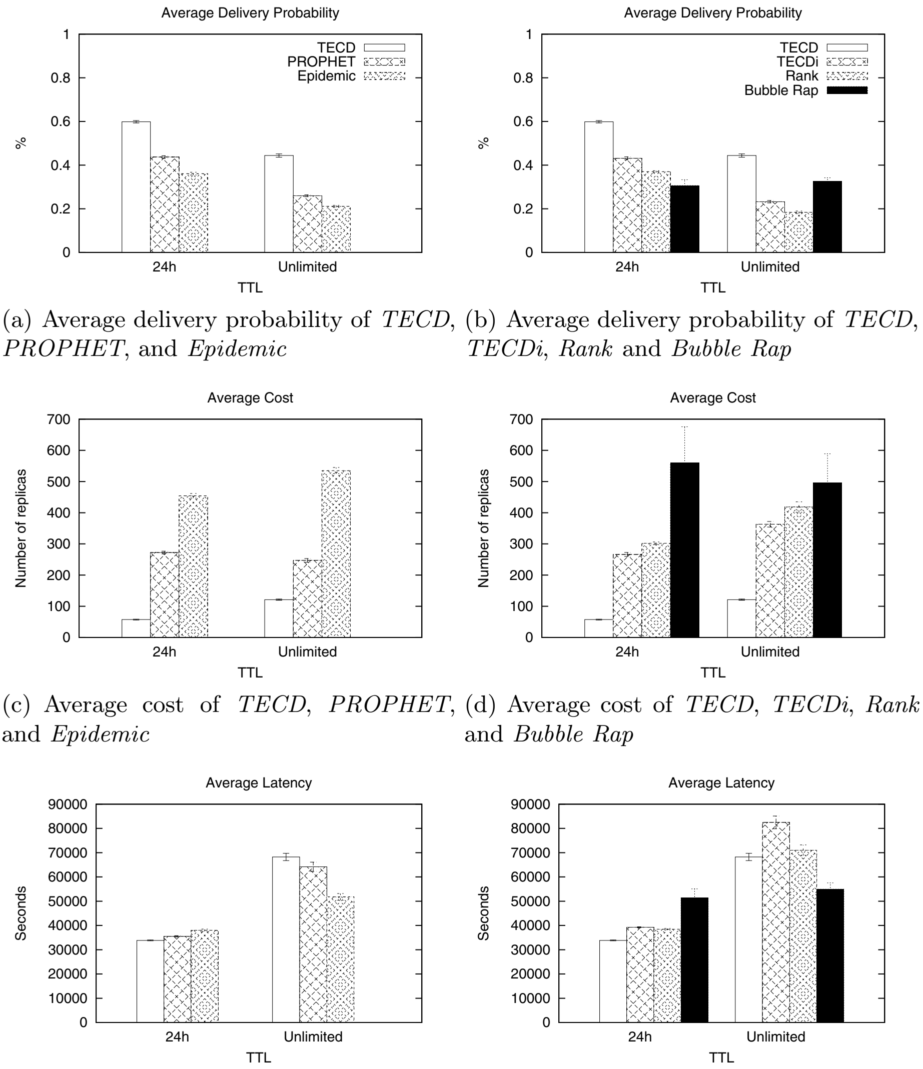
As shown in Fig. 2(a), *TECD* has a gain of 16.17 percentage points over *PROPHET* (59.87% against 43.70%) and a gain of 23.77 percentage points over *Epidemic* (59.87% against 36.10%) for a 24-hour TTL. The average delivery probability gain of *TECD* with unlimited TTL increases to 18.41 percentage points over *PROPHET* (44.39% against 25.97%), and to 23.26 percentage points over *Epidemic* (44.39% against 21.13%).

As *TECD* is able to reflect the daily routine and intensity of social ties, it only forwards messages to nodes that actually increase the probability of reaching the destination (i.e., are well socially connected) even if the carrier needs to keep the message for a little longer. *PROPHET*'s performance is affected by the presence of finite loops (happening occasionally) that keep messages in the system for longer times, which in turn use transmission opportunities that would be needed to keep a good delivery probability of other messages. *Epidemic*'s performance is explained by an aggressive replication strategy, which quickly exhausts buffer space, limited to 2 MB per node.

Regarding the average cost, Fig. 2(c) shows that for the 24-hour TTL, *TECD* produces 57.37 replicas to perform a delivery against 272.35 of *PROPHET* and 454.12 of *Epidemic*. For unlimited TTL, *TECD* has a higher number of replicas (121.30), but it is still lower than the ones created by *PROPHET* (247.11), and almost 4.5 times smaller than the ones created by *Epidemic* (535.04).

TECD presents a lower cost than *PROPHET*, since it wisely chooses the next best hop based on the social strength, avoiding a higher number of replicas (i.e., messages can be held in the carrier node for a longer time, while finding stronger social ties). Although the advantage of *TECD* decreases with unlimited TTL, this is acceptable since the cost is expected to increase as messages are allowed to live longer in the system. The cost reduction of *PROPHET* with unlimited TTL is related to the limited buffer (i.e., 2 MB), which does not allow it to further replicate as long-lived messages quickly exhaust nodes' buffer. As mentioned before, buffer limitation is not an issue for *TECD* since it wisely chooses next hops, leading to lower replications and to lower buffer occupancy. Regarding *Epidemic*, the trend is for the number of replicas to increase with TTL as messages take longer to expire: with unlimited TTL cost increases almost 18%.

For the average latency, Fig. 2(e) shows that with a 24-hour TTL, *TECD* has a lower average latency than *PROPHET* (1672.82 s lower) and *Epidemic* (4173.53 s lower). This is due to the fact that *TECD* is able to deliver messages with a lower number of hops (3.23 against 3.66 of *PROPHET* and 11.10 of *Epidemic*) and due to occasional finite loops that add delay in *PROPHET*. For unlimited TTL, *TECD* has slightly higher latency because nodes tend to hold messages for



(a) Average delivery probability of *TECD*, *PROPHET*, and *Epidemic*

(b) Average delivery probability of *TECD*, *TECDi*, *Rank* and *Bubble Rap*

(c) Average cost of *TECD*, *PROPHET*, and *Epidemic*

(d) Average cost of *TECD*, *TECDi*, *Rank* and *Bubble Rap*

(e) Average latency of *TECD*, *PROPHET*, and *Epidemic*

(f) Average latency of *TECD*, *TECDi*, *Rank* and *Bubble Rap*

Fig. 2. Comparison results for the considered performance metrics

longer time based on the expectation of having a better transmission opportunity in a later time interval.

Fig. 2(e) also shows that *PROPHET* has similar behavior as reported in [6]: latency increases with TTL. This happens since *PROPHET* has no mechanism to erase copies of messages that were already delivered, which occupy buffer and use forwarding opportunities of undelivered messages. The same justification helps to explain the higher latency of *TECD* and *Epidemic* with unlimited TTL. The difference in latency values with unlimited TTL for the three approaches is due to the fact that *Epidemic* puts much more messages in the system, which increases the probability of one message arriving first to the destination. Nevertheless, the cost of this strategy is very high (cf. Fig. 2(c)).

These first results show that representing social interactions based on the average duration of contacts in time intervals defined based on people daily habits leads to an overall better performance than just using inter-contact time information such as time since last encounter and frequency of encounters.

In summary, with a 24-hour TTL, *TECD* presents an average delivery probability 16% and 23% higher than *PROPHET* and *Epidemic*, producing almost $\frac{1}{5}$ and $\frac{1}{8}$ less replicas with a subtle gain in latency, respectively. With unlimited TTL, the advantage of using *TECD* in relation to *PROPHET* and *Epidemic* is still clear. Although *TECD* presents an average latency 6.29% and 31.79% higher than *PROPHET* and *Epidemic*, it presents an average delivery probability 18.41 and 23.26 percentage points higher, producing nearly 126 and 414 less replicas.

4.4 Comparison against Social-Based Algorithms

In this section, we analyze the performance of *TECD* and *TECDi* against *Bubble Rap* and *Rank*. The goal is two-fold: first, to analyze the potential benefits of considering social relationships as a complement to the importance of nodes (metric used by *Rank*) for the identification of accurate social interactions; and, second, to analyze how an approach able to capture the network dynamics based on a time-evolving social-based solution behaves when compared to solutions that rely solely on the identification of social structures, as with *Bubble Rap*.

In what concerns the average delivery probability (cf. Fig. 2(b)), *TECD* overcomes *Rank* and *Bubble Rap* in 22.84 and 29.26 percentage points, respectively, for a 24-hour TTL (59.87% against 37.03% and 30.61%, respectively). For unlimited TTL, all approaches present lower delivery probability (while keeping the same performance order), with the exception of *Bubble Rap*, which improves as TTL increases.

It is also observed that *TECDi* has a gain of 6.06 percentage points over *Rank* for a 24-hour TTL (43.09% against 37.03%). This gain is higher than the one of *Rank* for a scenario with unlimited TTL (23.25% against 18.37%).

Regarding *Bubble Rap*, *TECDi* has a gain of more than 12 percentage points for the 24-hour TTL case (43.09% against 30.61%), since *Bubble Rap* needs some time to create communities, which affects its delivery rate. However, for the unlimited TTL case, *Bubble Rap* overcomes *TECDi* by more than 9 percentage points, which was expected since it has now more time to form communities increasing the delivery capability as reported in Hui et al.

Based on Fig. 2(b), our findings show that, considering the strength of social ties in addition to the importance of nodes, it brings benefits in terms of delivery probability. With *TECDi* the importance of nodes is influenced by the social strength they have with neighbors. This means for instance that a node, with a high set of weakly related neighbors, will have a lower importance with *TECDi* than with *Rank*. Moreover, next hops are selected among the important nodes that have a stronger social tie with the message carrier, which helps in the case where more than one contact is needed to transfer a message.

In what concerns the average cost (cf. Fig. 2(d)), *TECD* achieves the lowest cost with 24-hour TTL when compared to *Rank* and *Bubble Rap* (57.37 replicas against 302.05 and 559.69, respectively) as it chooses the best next hop based on the social strength. The same behavior occurs with unlimited TTL as *TECDi* is able to reduce the number of potential next hops from the set of important neighbors (being more selective in relation to nodes with high degree but weak social ties with their neighbors). However, cost increases with TTL, since messages stay longer in the system and are subjected to more replication. *Bubble Rap* presents the highest costs independently of TTL, as replication occurs based on global centrality when communities are not yet fully formed, and nodes with higher global rank (i.e., those which move throughout the entire scenario like buses) are always receiving messages in an attempt to have them readily delivered. The decrease in cost of *Bubble Rap* with unlimited TTL is due to messages starting to exhaust buffer in carrier nodes, which means less messages to be replicated.

Regarding the average latency (cf. Fig. 2(f)), *TECD* manages to have the lowest latency (less 4598 s and 17476 s than *Rank* and *Bubble Rap*, respectively) whereas *TECDi* has a subtle higher latency (more 818.37 s) than *Rank,* but still has a lower latency (less 12059 s) than *Bubble Rap* with a 24-hour TTL. With unlimited TTL, *TECD* still has a lower latency (less 2752 s) than *Rank,* but a higher latency (more 13299 s) when compared to *Bubble Rap,* while *TECDi* has a higher latency than *Rank* and *Bubble Rap* (11492.86 s and 27545 s, respectively). Increase in latency of *TECD* is due to messages being held longer in attempt to find better next hops. As for *TECDi,* messages are replicated based on node importance, and as they reach top-ranked nodes, they take more time to reach destination especially if those do not interact much outside their social groups.

The high latency of *Bubble Rap* with 24-hour TTL is expected as messages are subject either to the formation of communities or to the fact that their holders must come in contact to higher popularity ranked nodes to perform forwarding, thus influencing the overall latency experienced. As the number of delivered messages increased in the unlimited TTL case, so did its overall latency.

When compared to *TECD*, both *Rank* and *TECDi* present a higher average latency, since nodes tend to hold messages longer, especially if the current node has a high importance factor. The impact of the importance factor also explains the higher latency of *TECDi* in relation to *Rank*, since the former also consider the social strength between nodes, in addiction to the importance of nodes.

In summary, with a 24-hour TTL, *TECD* overcomes *Rank* and *Bubble Rap* in terms of delivery probability for more than 22.84 and 29.26 percentage points respectively, creating $\frac{1}{5}$ and $\frac{1}{10}$ less replicas with 13.58% and 51.63% lower latency. Yet, *TECDi* presents an average delivery probability of 6.06 and 12.47

percentage points higher than *Rank* and *Bubble Rap* respectively, producing almost 36 and 294 less replicas with a 2.12% higher latency than *Rank* and 30.7% lower latency than *Bubble Rap*.

With unlimited TTL, *TECD* has still better performance in terms of delivery probability and cost than *Rank* and *Bubble Rap*, although its latency is 24.20% higher than *Bubble Rap*. In the case of *TECDi*, it brings more advantages than *Rank*, because although the former presents an average latency 16.18% higher than *Rank*, it has an average delivery probability 4.88 percentage points higher than *Rank*, producing 55 less replicas. With *Bubble Rap*, although it has a gain of 9 percentage points regarding delivery probability and a lower latency, *TECDi* manages to spare resources by creating almost 37% less replicas.

4.5 Scalability Analysis

We analyze the memory needed for computing *TECD* and *TECDi*. Considering a worst case scenario with k time slots and n nodes, where every node meets all other nodes in each ΔT_i, we have:

- $n \times (n-1)$ variables to store the starting time for every new connection.
- $n \times (n-1)$ variables to store TCT computations.
- $k \times n \times (n-1)$ variables to store AD computations.

If each variable has X bits, *TECD*'s needed resources is given by Eq. 5.

$$TECD_{alloc} = n \times (n-1) \times (k+2) \times X\, bits \tag{5}$$

In our scenario we have 150 nodes, 6 time slots, and 64 bit double for storing, which results in 1.364 MB of total memory usage in the system, which means that in average each node needs up to 4 MB (including the 2 MB of buffer space).

To use *TECDi*, nodes need to store their importance and the importance of nodes they meet. Thus, the amount of needed resources is given by Eq. 6.

$$TEDCi_{alloc} = n^2 \times X + TECD_{alloc}\, bits \tag{6}$$

Assuming the aforementioned worst case, *TECDi* needs a storage capacity of 1.536 MB, which means that in average a node needs to reserve the same 4 MB.

In case of rather limited buffer, a solution is to keep track of the best weights, eliminating those under a threshold. A configuration with pre-defined thresholds will be considered to investigate which is the most suitable value to be used.

5 Conclusions and Future Work

Prior art proved that social relationships are useful for data exchange in opportunistic networks. Thus, it is clear that solutions based on the structure of the network have much better performance than solutions based solely on contacts. However, we can also observe that the dynamics of the network (i.e., based on daily routines) should also be addressed in order to improve even more opportunistic routing. Hence, we propose two utility functions which are based on the daily routine of people, and can transcribe movement patterns resulting from social ties into equivalent social weighted representations relevant for forwarding: *Time-Evolving Contact Duration* (TECD), where social interactions are weighted reflecting the social daily routines of users, and *TECD Importance*, which includes the social strength between nodes and their importance.

As presented in Sections 4.3 and 4.4, network dynamics can indeed improve the performance of opportunistic routing when compared to solutions based on the structure of the network and on node contact. *TECD* stands out amongst the contact- and social-based proposals and it even has advantages over *TECDi*: delivery probability gains up to 21.1 percentage points producing ~242 less replicas with a lower latency (~17.3%). This is due to the fact that *TECD* replicates messages based on the social strength among nodes (which is very reliable), while *TECDi* does it based on node importance. This dependency on node importance results in useless replications which increases *TECDi*'s cost contributing to its lower delivery probability and higher latency. Still *TECDi* has great potential (when compared to the remaining solutions) in improving forwarding, and this approach should be further investigated.

This paper showed the advantages of using solutions based on the dynamics of the network for routing. We prove this by comparing a solution based on utility functions that are aware of users' daily routines against prior art.

As future work, we aim to investigate a hybrid utilization, with *TECD* and *TECDi,* creating a new opportunistic routing proposal to forward messages. We also plan to finetune both utility functions to improve delivery probability, cost and latency, and evaluate such proposals considering human traces to show its potential in real scenarios.

Acknowledgment. To FCT for financial support to Waldir Moreira's PhD grant (SFRH/BD/62761/2009) and to the User-Centric Routing project (PTDC/EEA-TEL/103637/2008).

References

1. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on the design of opportunistic forwarding algorithms. In: Proceedings of INFOCOM, Barcelona, Spain (April 2006)
2. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: social-based forwarding in delay tolerant networks. IEEE Transactions on Mobile Computing 10(11), 1576–1589 (2011)
3. Moreira, W., Mendes, P., Sargento, S.: Assessment model for opportunistic routing. In: Proceedings of the IEEE LATINCOM, Belem, Brazil (October 2011)
4. Hossmann, T., Spyropoulos, T., Legendre, F.: Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing. In: Proceedings of IEEE INFOCOM, San Diego, USA (March 2010)
5. Eagle, N., Pentland, A.: Eigenbehaviors: identifying structure in routine. Behavioral Ecology and Sociobiology 63, 1057–1066 (2009)

6. Lindgren, A., Doria, A., Schelén, O.: Probabilistic Routing in Intermittently Connected Networks. In: Dini, P., Lorenz, P., Souza, J.N.d. (eds.) SAPIR 2004. LNCS, vol. 3126, pp. 239–254. Springer, Heidelberg (2004)
7. Song, L., Kotz, D.F.: Evaluating opportunistic routing protocols with large realistic contact traces. In: Proceedings of ACM MobiCom CHANTS, Montreal, Canada (September 2007)
8. Nelson, S., Bakht, M., Kravets, R.: Encounter-based routing in DTNs. In: Proceedings of INFOCOM, Rio de Janeiro, Brazil (April 2009)
9. Costa, P., Mascolo, C., Musolesi, M., Picco, G.P.: Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. IEEE Journal on Selected Areas in Communications 26, 748–760 (2008)
10. Mtibaa, A., May, M., Ammar, M., Diot, C.: Peoplerank: Combining social and contact information for opportunistic forwarding. In: Proceedings of INFOCOM, San Diego, USA (March 2010)
11. Keränen, A., Ott, J., Kärkkäinen, T.: The one simulator for dtn protocol evaluation. In: Proceedings of SIMULTools, Rome, Italy (March 2009)
12. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Tech. Rep. CS-200006, Duke University (2000)

Autonomic Cooperative Networking for Vehicular Communications

Michał Wódczak

Ericsson, ul. Umultowska 85, 61–614 Poznań, Poland,
michal.wodczak@ericsson.com

Abstract. As vehicular systems are expected to become the key element of the global networking ecosystem, it is crucial to ensure that the relevant technologies are included in their development from the very outset. A distinctive feature of vehicular networks is their envisaged high complexity in terms of network composition. Therefore, certain dose of automation is required for the purposes of guaranteeing smooth and robust system operation. First, it is expected that there will be a need for the vehicles to express capabilities of autonomic configuration in order to address the issues of rapid topology changes and distributed nature of the network. Second, a very relevant question of self-management needs to be answered so it is possible to understand how network nodes, i.e. vehicles, can express cooperative behaviors, manifested through, for example, the ability to perform autonomic cooperative communications and routing.

Keywords: cooperative communications, autonomic system design, vehicular networks.

1 Introduction

This paper outlines the evolution of the technologies leading to the instantiation of autonomic cooperative networking for vehicular systems [13], [12]. The discussion covers not only spatio-temporal transmission and its successful transition into cooperative relaying but also puts emphasis on the very important role of network layer routines eventually enhanced with the concept of autonomic system design. Perceived from a general perspective, cooperative transmission seems to be emerging as the key enhancement to vehicular systems in terms of providing reliable data transmission. As this approach capitalises on the exploitation of spatio-temporal processing techniques, it can offer very promising improvements in performance of wireless communications through the use of transmission diversity provided by relay nodes. Such relay nodes may form virtual antenna arrays [7] and act for example as a distributed space-time block or trellis encoder [11]. In the case of vehicular networks, however, the system dynamics requires special logic able to handle the continually changing network configuration. This can be achieved through cross-layering with the aid of network layer routines [22]. For this purpose one may, in turn, employ for example the

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 112–125, 2012.

Multi-Point Relay station selection heuristic of the Optimised Link State Routing protocol which can be easily exploited to organise virtual antenna arrays [5]. Last but not least, the system needs to be stable and scalable, and so large-scale vehicular networks should implement the notion of self-management [4]. A future vehicular network may be actually even expected to somewhat imitate the behavior of a living organism meaning it should be capable of functioning by itself without any necessity for a specific external human intervention during most of the time of operation [13], [12]. Such autonomic network architectures are currently being developed with the aid of the concept of decision elements and control loops and are being pushed onto the standardisation path [3].

The paper has been organised in the following way. Section 2 provides all the necessary background on spatio-temporal processing so the idea of applying cooperative relaying to vehicular systems can be introduced in Section 3. Next, the role of network layer routines is explained in Section 4 in context of organising vehicles into virtual antenna arrays. Then the concept of involving the rationale behind autonomic system design is advertised for in Section 5 and the simulation results for autonomic equivalent distributed space-time encoder, working in both the block and trellis mode, are presented in Section 6. The paper is concluded in Section 7.

2 Spatio-temporal Transmission

There are a few space-time processing techniques which may be employed for the purposes of pre-processing the transmitted signals in such a way that they are more robust to the wireless radio channel impairments. Among them there is space-time block coding, introduced by Alamouti [1], which offers diversity gain but no coding gain. Therefore, despite its name, space-time block coding may be generally perceived as a modulation rather than a coding technique. It was designed to provide an additional spatio-temporal diversity in wireless systems, employing multi-element antenna arrays for the purposes of enhancing the reliability of transmission. When compared to the classic solutions based on receive diversity, space-time coding allows to shift the complexity connected with multiple antennas from small mobile user terminals to base stations. Among the most significant advantages of this approach, there is the reduced complexity of user terminals, lower cost of installing one multi-element antenna array at the base station only, as well as the possibility of guaranteeing reasonable spacing among elements of such an antenna array. The base space-time block code G_2 is defined with the use of the matrix (1):

$$G_2 = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix} \tag{1}$$

This code may be used in a system employing two transmit and any number of receive antennas. More specifically, in the first timeslot the x_1 and x_2 symbols are sent by the first and second transmit antenna respectively and then, in the

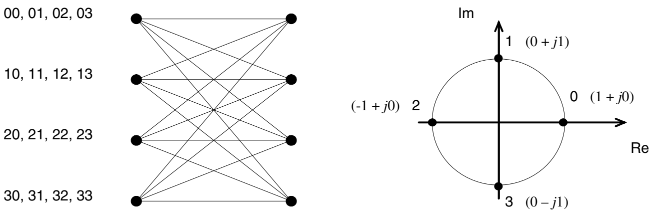


Fig. 1. Base space-time trellis code exploiting 4-PSK modulation

second timeslot, the $-x_2^*$ and x_1^* symbols are transmitted in the same manner, being the complex conjugates of the original ones.

The other technique taken into account in this paper is space-time trellis coding [18], [19]. In contrast to space-time block coding, it introduces additional relations among specific sequences transmitted by distinct antennas, as well as the symbols constituting these sequences, so that apart from diversity gain, additional coding gain may be also observed. The base space-time trellis code exploiting the 4-PSK modulation scheme, is presented in Figure 1 [18]. The numbers placed to the left of the trellis diagram should be interpreted in the following way: the most significant digit represents the current state, whereas the least significant one corresponds to the input and therefore also to the next state [18]. It means that the consecutive pairs of the encoder input bits determine the transition from the current state to the following one. Consequently, two symbols are relayed to the transmit antennas, and more specifically the first antenna transmits the channel symbol informing about the current state, whereas the second antenna transmits the channel symbol informing about the next state.

3 Cooperative Relaying

The issue of cooperative relaying emerged as a very promising method for improving the process of transmission in wireless mobile networks [14]. This is where the application of the discussed space-time processing techniques to cooperative relaying can be advantageous as in general it true that thanks to significant separation between Relay Nodes (RN) a Virtual Multiple Input Multiple Output (MIMO) radio channel may be formed in the cooperation phase. This way cooperative transmission may improve the reliability of communication through the use of diversity provided by network nodes available to assist in the transmission between a given pair of source and destination nodes [10]. This is possible because the rationale behind spatial-temporal processing discussed in the preceeding section can be easily mapped onto networking, as long as sufficiently tight synchronization is guaranteed. Cooperative transmission evolved from of the conventional relaying scheme [24] and usually it consists of two phases. First the Source Node (SN) sends its information to the RN, which fully decodes the received signal, then re-encodes it and sends to the Destination Node (DN). Such

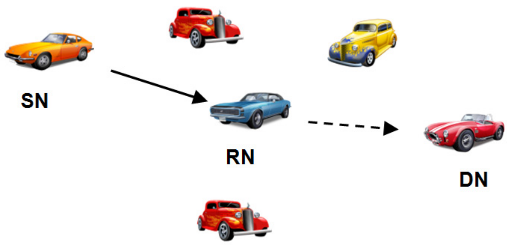


Fig. 2. Conventional relaying

an approach makes it feasible to either reduce the transmit power or extend the transmission range. As it was explained in the description of space-time processing techniques, no diversity gain can be offered here. This approach is also known as L3DF because taking into account the Open Systems Interconnection layered structure the operation of relaying is performed at the network layer [9].

An extension is the basic cooperative relaying case where additionally a direct link between SN and DN is available as in Figure 3. Here, both the DN and RN receive the transmitted signal and then the RN may additionally resend its copy towards the DN to enhance the system performance by exploiting the aforementioned diversity gain. The most advanced approach actually consists in application of distributed space-time block coding [11]. SN broadcasts its signal, which is consequently received by the DN as well as by the potential RNs. Afterwards, this signal is processed by these intermediate nodes and eventually resent towards the DN. The concept of distributed space-time block coding may be perceived as a special case of virtual antenna arrays. It is then crucial to note that the latter can be generalised even further to encompass multi-hop set-ups [6], [7].

Cooperative transmission is not only applicable to mobile ad hoc networks, mesh networks or sensor networks but there already exist interesting applications to the next generation cellular systems as well [8]. Regardless the environment,

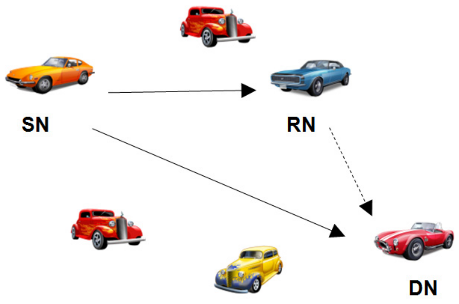


Fig. 3. Cooperative relaying

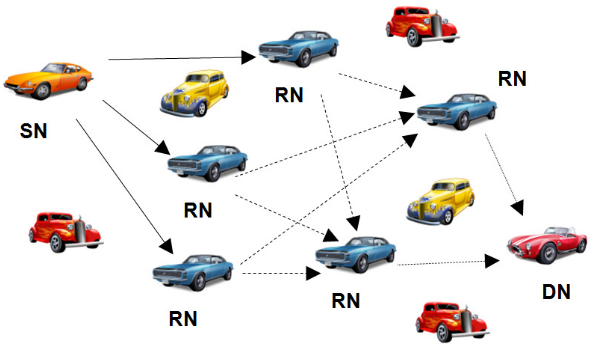


Fig. 4. Generalised virtual antenna array concept

however, in most of the cases there is question of the selection of relay nodes to be included in a virtual antenna array. This is somewhat similar to transmit antenna selection in the case of plain space-time coding systems where one could observe benefits from proper antenna selection [20]. For networked systems, however, this issue becomes even more substantial and complex as the network topology may be changing rapidly and it is very unpredictable. One of the proposed approaches assumes employing specific existing routing layer mechanisms for the purposes of gaining access to and capitalising on topology information readily available at the network layer [22]. This approac is further discussed in Section 6 where the evaluation assumptions are outlined.

4 Network Layer Routines

The Optimised Link State Routing protocol was designed for Mobile Ad-hoc Networks (MANETs). As it belongs to the link-state class of protocols, it somewhat resembles the classic solution in the form of the Open Shortest Path First (OSPF) protocol, however, it is better suited to mobile environments. Such environments are usually characterised by very dynamic changes in the topology of the mobile network and therefore the protocol should be tailored accordingly so that, keeping the overhead at a reasonable level, it was able to follow those changes and provide accurate routing information.

Although, due to the broadcast nature of the wireless channel, many intermediate intermediary nodes typically receive the radio transmission originated by one of the neighboring nodes, it makes sense only for some of them to participate in cooperative relaying. Routing mechanisms can be used to facilitate this process, so information available at the network layer may be exploited to for the coordinatione of the transmission at the link layer. The selected relay nodes may form the aforementioned virtual antenna array(s) and then perform the operation of distributed space-time processing to orthogonalise the wireless radio channel. A perfect candidate solution addressing these needs is actually the

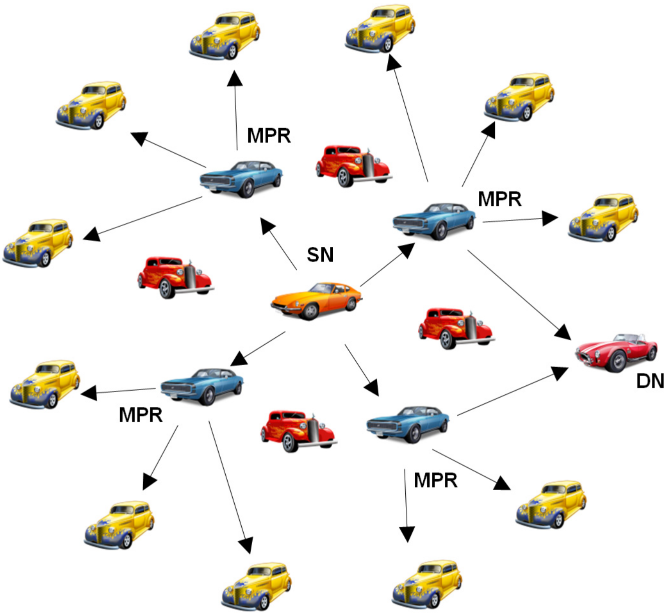


Fig. 5. Multi-point relay selection

aforementioned OLSR protocol. It is characterised by proactive topology discovery and an optimised broadcasting mechanism of the Multi-Point Relay (MPR) station selection heuristic. As described in [22] and [21], [23] the modification of this mechanism allows for a seamless integration of the concept of virtual antenna aided cooperative transmission based on space-time block coding with the routines of network layer protocol. It means that, thanks to careful extensions to the OLSR protocol ensuring its backward compatibility, one is able to capitalise on the routing mechanisms and additional information readily available at the network layer for the purposes of optimising the performance of a link layer system employing the aforementioned virtual antenna arrays.

The MPR selection heuristic [15] is aimed at optimising the protocol overhead during the phase of topology recognition (Figure 5). It is performed with the use of both the set of one-hop and the set of two-hop neighbours, identified with the aid of Hello messages. To this end, each node x issues such messages containing the addresses of all the neighbours it has discovered within its one-hop neighbourhood $N(x)$ together with the corresponding link codes [5]. Following, a node n in $N(x)$ should be able to acquire knowledge about its two-hop neighbourhood $N^2(x)$, reachable through the node x, in the same manner. Having identified its both one-hop and two-hop neighbourhoods, the node x may elect MPRs with the aid of the heuristic summarised in short below [5]. First, the node x includes in the $MPR(x)$ set all its symmetric one-hop neighbours being the only ones to provide reachability to a node n^2 in the strict symmetric two-hop neighbourhood, and always willing to carry and forward traffic [15].

The heuristic continues selecting this node in the $N(x)$ set, which has not been inserted into the $MPR(x)$ set so far and is characterised by the highest willingness to carry and forward traffic, as long as there still exist any uncovered nodes in $N^2(x)$. For multiple choice the one is chosen which provides the highest reachability, i.e. through which the highest number of still uncovered nodes in $N^2(x)$ can be reached. Otherwise, if it is impossible to select one node only, the one with the highest degree is chosen, where the degree of a one-hope neighbour denotes the number of its symmetric neighbours, excluding all the members of $N(x)$ and the node x, performing the computation [5].

In particular, OLSR appears of so special interest here because it is not only already integrated with cooperative transmission, but also very well aligned with the paradigm of autonomic networking, as it allows for a network node to decide on its willingness to carry and forward traffic, which is exactly what is meant by an enabler for autonomic behaviors.

5 Autonomic Networking

As the network complexity increases, the network layer per se is no longer sufficient when it comes to the general scope coordination. In particular, to accommodate the routing information enhanced cooperative transmission, a specific architectural extension depicted in Figure 6 is necessary as a component aligned with the rationale behind the Generic Autonomic Network Architecture (GANA) [4]. Autonomic networking has emerged as one of the most promising approaches towards the instantiation of the self-managing future networked systems [4] such as vehicular communications [13], [12]. Autonomicity is defined as the ability to self-configure without a need for any external intervention, while the flavour of a system being autonomous means its ability to display certain dose of cognition. The inherent feature of autonomic networking is a need for continuous

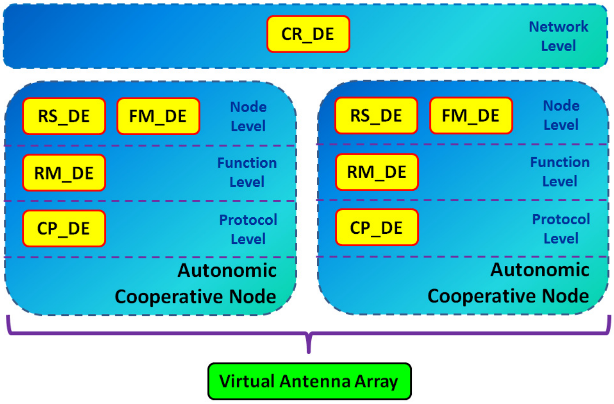


Fig. 6. Autonomic cooperative node from architectural perspective

monitoring so the network is able to self-configure according to the imposed policies and taking into account additional monitoring related information [2]. The aforementioned factors are particularly important for vehicular networks which depending on the scenario might be characterized by a very dynamically changing topology, affecting the possibility for efficient cooperation. Generally, such an autonomic network should behave like a living organism or rather imitate its internal processes remaining in close correlation but not requiring external intervention during most of the time of operation. In particular, such systems are on control loops, where a Decision Element (DE) is controlling a managed entity (ME) based on a closed information flow and with the use of external monitoring and policies related data (Figure 7).

Such decision elements interact among themselves in hierarchical management structures based on the input from control loops. This is a very important aspect because monitoring is necessary for acquiring up-to-date information data regarding the network topology, which may positively affect the safety situation in the network. At the same time, policing might be used to impose certain behaviors on distinct vehicles, groups of vehicles or even the whole network in order to address certain objectives, related for example to the traffic load optimisation. In particular, starting from the protocol level as described in Figure 6 a new Cooperative Processing decision element (CP_DE) needs to be introduced having responsibility over controlling the aspects of cooperative transmission protocol related to physical emulation of the distributed space-time encoder. This operation is equivalent to the processing of the relayed signal according to the operation of a given space-time encoder, either block or trellis one. The operation of CP_DE needs to be aligned with the already existing Routing Management decision element RM_DE. This is necessary for the proper synchronisation of

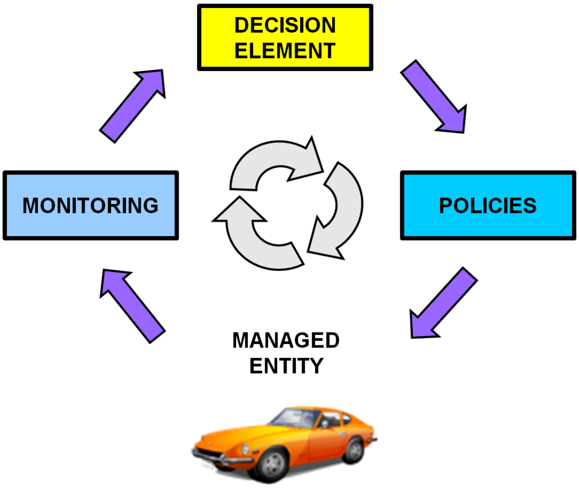


Fig. 7. Control loop

the routing tables maintained at the cooperating nodes [21], [23]. Moreover, the RM_DE also needs to act pursuant to the interactions with other existing DEs, i.e. the Resilience and Survivability decision element RS_DE and Fault Management decision element FM_DE. In this case, the RS_DE covers the aspects of service resilience and survivability. For this it interacts with FM_DE which controls the symptoms suggesting that a failure, e.g. in terms of service, may be imminent. Finally, while these DEs are located at autonomic cooperative nodes, it is still necessary to provide substantial coordination at the network level. This task is accomplished by the Cooperative Routing decision element CR_DE which is responsible for overseeing the situation from a higher level perspective and orchestrating the concurrent cooperative and non-cooperative transmissions among vehicles [13], [12].

6 System Assumptions and Evaluation

The evaluation of a system comprising all the above components is a challenging task. Therefore, it is assumed that the operation of the routing information enhanced cooperative transmission as described in [23], [21], [22] is enhanced with the notion of an autonomic equivalent distributed space-time encoder. In particular, using the notation introduced for the MPR selection heuristic, each neighbour n having zero degree is removed by the source vehicular node x from the set $N(x)$. Then the classic MPR selection heuristic is executed iteratively over the set $N(x)$, until all the potential MPR nodes have been subdivided into disjoint $MPR^i(x)$ sets. Each such iteration should result in additional multipoint relay set, i.e. secondary, ternary, etc. At the same time, all the nodes from these sets, are expected to be assigned to the most relevant virtual antenna arrays $VAA(x, n^{(2)})$, providing cooperative connectivity between the source node x and the destination node $n^{(2)}$. This means that any intermediate node n may be included in more than one virtual antenna array operating according to the definition of an autonomic equivalent distributed space-time encoder outlined below [21], [23].

Definition: *A set of perfectly synchronized distributed relay nodes connected to the source node via error-free links and able to cooperatively encode the received signals as if they constituted a given space-time block or trellis encoder, conceptually forms and is defined as an autonomic equivalent distributed space-time encoder.*

Following, the operation of such encoder has been analysed with the aid of simulations. Different configurations were validated where the power emitted by each transmitting vehicle was always normalised for the total transmitted power to be equal to 1. Always 10 million bits were transmitted and the destination vehicle was assumed to have from 1 up to 3 receiving antennas. In particular, the signal (2) received by a receive antenna j may be then written as [1], [18]:

$$r_t^j = \sum_{i=1}^{N} h_{i,j} s_t^i + \eta_t^j \tag{2}$$

where $h_{i,j}$ denotes a channel coefficient between the antenna of the relaying vehicle i and the receive antenna j at the destination vehicle, s_t^i represents the symbol transmitted by antenna of vehicle i and the noise samples η_t^j are modeled by the complex Gaussian process with zero mean and $N_0/2$ variance per dimension. For space-time block coding the following metric (3) was used [1]:

$$z = \sum_{t=1}^{L} \sum_{j=1}^{M} \left| r_t^j - \sum_{i=1}^{N} h_{i,j} s_t^i \right|^2 \tag{3}$$

where t denotes the time slot and i denotes the transmit antenna. For space-time trellis coding, transitions was assigned weights (4) according to [18]:

$$w_{x,y} = \sum_{j=1}^{M} \left| r_t^j - \sum_{i=1}^{N} h_{i,j} s_t^i \right|^2 \tag{4}$$

The results achieved for the base space-time block and space-time trellis codes, introduced in Section 2, are presented in Figure 8 and Figure 9, respectively.

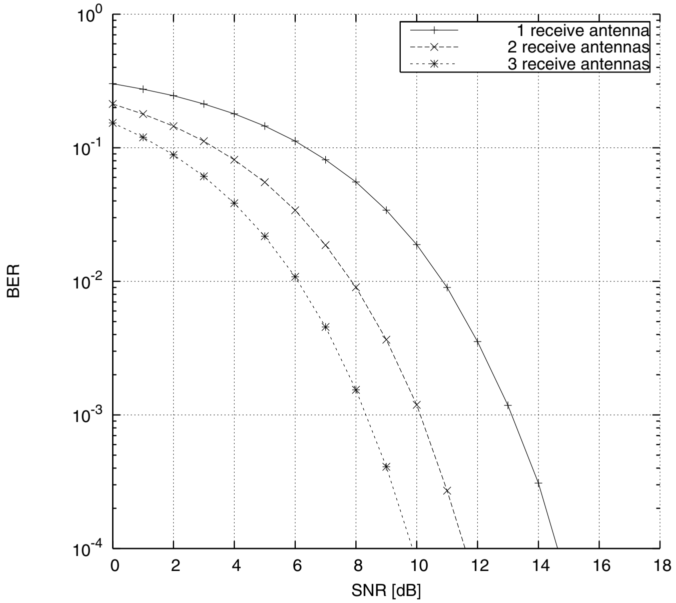


Fig. 8. Performance of G_2 equivalent system for 1, 2, and 3 receiving antennas

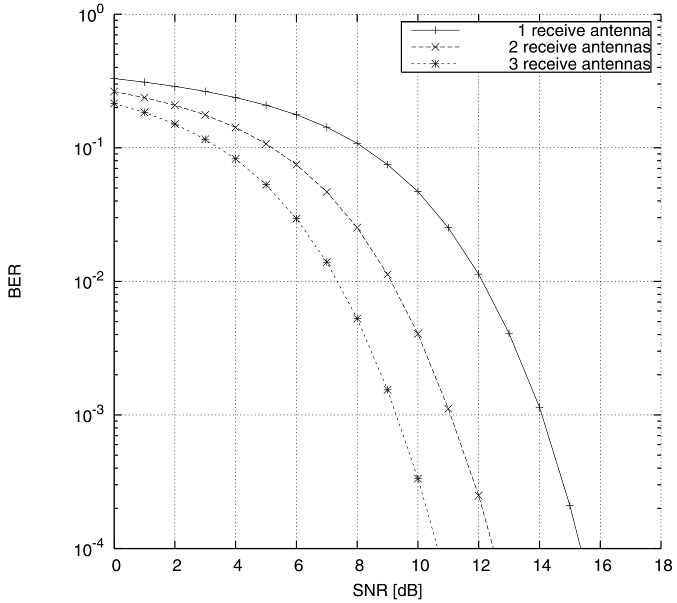


Fig. 9. Performance of space-time trellis coded equivalent system for 1, 2, and 3 receiving antennas

Additionally, two more advanced space-time block codes G_3 (5) and H_3 (6) were evaluated [17], [16]:

$$G_3 = \begin{bmatrix} x_1 & x_2 & x_3 \\ -x_2 & x_1 & -x_4 \\ -x_3 & x_4 & x_1 \\ -x_4 & -x_3 & x_2 \\ x_1^* & x_2^* & x_3^* \\ -x_2^* & x_1^* & -x_4^* \\ -x_3^* & x_4^* & x_1^* \\ -x_4^* & -x_3^* & x_2^* \end{bmatrix} \tag{5}$$

$$H_3 = \begin{bmatrix} x_1 & x_2 & \frac{x_3}{\sqrt{2}} \\ -x_2^* & x_1^* & \frac{x_3}{\sqrt{2}} \\ \frac{x_3^*}{\sqrt{2}} & \frac{x_3^*}{\sqrt{2}} & \frac{(-x_1-x_1^*+x_2-x_2^*)}{\sqrt{2}} \\ \frac{x_3^*}{\sqrt{2}} & -\frac{x_3^*}{\sqrt{2}} & \frac{(x_2+x_2^*+x_1-x_1^*)}{\sqrt{2}} \end{bmatrix} \tag{6}$$

The achieved results show that one may expect theoretical gains in terms of system performance through the application of autonomic cooperative networking to vehicular systems. Given all the assumptions are met or close to ideal, it becomes really thrilling to think what the vehicular network of the future may look like.

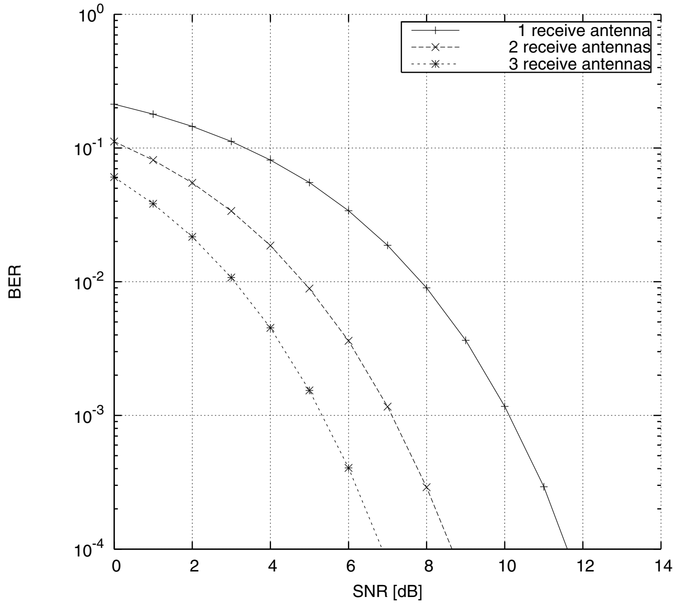


Fig. 10. Performance of G_3 equivalent system for 1, 2, and 3 receiving antennas

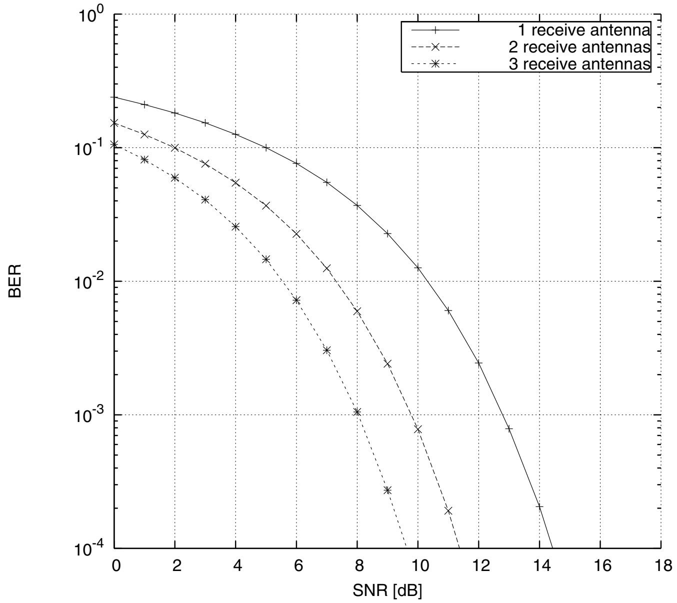


Fig. 11. Performance of H_3 equivalent system for 1, 2, and 3 receiving antennas

7 Conclusion

In this paper the concept of autonomic cooperative networking for vehicular systems has been introduced. In particular the required background on spatio-temporal processing was provided and the idea of its transition into cooperative relaying to be applied to vehicular systems was outlined. It was also discussed how important the role of network layer routines would be in context of organising vehicles into virtual antenna arrays, and then also the concept of involving the rationale behind autonomic system design was advocated for to provide durable and efficient system operation.

References

1. Alamouti, S.: A Simple Transmit Diversity Technique for Wireless Communications. IEEE Journal on Selected Areas in Communications 16(8), 1451–1458 (1998)
2. Liakopoulos, A., Zafeiropoulos, A., Polyrakis, A., Grammatikou, M., Gonzlez, J.M., Wódczak, M., Chaparadza, R.: Monitoring Issues for Autonomic Networks: The EFIPSANS Vision. In: European Workshop on Mechanisms for the Future Internet (2008)
3. Chaparadza, R., Ciavaglia, L., Wódczak, M., Chen, C.-C., Lee, B.A., Liakopoulos, A., Zafeiropoulos, A., Mancini, E., Mulligan, U., Davy, A., Quinn, K., Radier, B., Alonistioti, N., Kousaridas, A., Demestichas, P., Tsagkaris, K., Vigoureux, M., Vreck, L., Wilson, M., Ladid, L.: ETSI Industry Specification Group on Autonomic Network Engineering for the Self-managing Future Internet (ETSI ISG AFI). In: Vossen, G., Long, D.D.E., Yu, J.X. (eds.) WISE 2009. LNCS, vol. 5802, pp. 61–62. Springer, Heidelberg (2009)
4. Chaparadza, R., Papavassiliou, S., Kastrinogiannis, T., Vigoureux, M., Dotaro, E., Davy, K.A., Quinn, M., Wódczak, M., Toth, A.: Creating a viable Evolution Path towards Self-Managing Future Internet via a Standardizable Reference Model for Autonomic Network Engineering. In: Tselentis, G., Domingue, J., Galis, A., Gavras, A., Hausheer, D., Krco, S., Lotz, V., Zahariadis, T. (eds.) Towards the Future Internet - A European Research Perspective. IOS Press (May 2009) ISBN: 978-1-60750-007-0
5. Clausen, T., Jacquet, P.: Optimised Link State Routing Protocol (OLSR). RFC 3626 (October 2003)
6. Dohler, M., Gkelias, A., Aghvami, H.: 2-Hop Distributed MIMO Communication System. IEE Electronics Letters 39(18) (June 2003)
7. Dohler, M., Gkelias, A., Aghvami, H.: A resource allocation strategy for distributed MIMO multi-hop communication systems. IEEE Communications Letters 8(2), 99–101 (2004)
8. Dottling, M., Irmer, R., Kalliojarvi, K., Rouquette-Leveil, S.: System Model, Test Scenarios, and Performance Evaluation. In: Dottling, M., Mohr, W., Osseiran, A. (eds.) Radio Technologies and Concepts for IMT-Advanced. Wiley (December 2009) ISBN: 978-0-470-74763-6
9. Herhold, P., Zimmermann, E., Fettweis, G.: Cooperative multi-hop transmission in wireless networks. Computer Networks Journal 49(3), 299–324 (2005)
10. Laneman, J.N., Tse, D.N.C., Wornell, G.W.: Cooperative diversity in wireless networks: Efficient protocols and outage behavior. IEEE Transactions on Information Theory 50(12), 3062–3080 (2004)

11. Laneman, J.N., Wornell, G.W.: Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. IEEE Transactions on Information Theory 49(10), 2415–2425 (2003)
12. Li, J., Wódczak, M., Wu, X., Hsing, T.R.: Vehicular Networks and Applications - Challenges, Requirements and Service Opportunities. Academy Publisher Journal of Communications (accepted for publication, 2012)
13. Li, J., Wódczak, M., Wu, X., Hsing, T.R.: Vehicular Networks and Applications: Challenges, Requirements and Service Opportunities. In: International Conference on Computing, Networking ans Communications (ICNC), Maui, Hawai, USA, January 30 -February 2 (2012)
14. Pabst, R., Walke, B., Schultz, D.C., Herhold, P., Yanikomeroglu, H., Mukherjee, S., Viswanathan, H., Lott, M., Zirwas, W., Dohler, M., Aghvami, H., Falconer, D., Fettweis, G.: Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio. IEEE Communications Magazine 42(9), 80–89 (2004)
15. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. In: 35th Annual Hawaii International Conference on System Sciences, HICSS (January 2002)
16. Tarokh, V., Jafarkhani, H., Calderbank, A.R.: Space-time block codes from orthogonal designs. IEEE Transactions on Information Theory 45(5), 1456–1467 (1999)
17. Tarokh, V., Jafarkhani, H., Calderbank, A.R.: Space-time block coding for wireless communications: performance results. IEEE Journal on Selected Areas in Communications 17(3), 451–460 (1999)
18. Tarokh, V., Seshadri, N., Calderbank, A.R.: Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction. IEEE Transactions on Information Theory 44(2), 744–765 (1998)
19. Tarokh, V., Seshadri, N., Calderbank, A.R.: Space-Time Codes for High Data Rate Wireless Communication: Performance Criteria in the Presence of Channel Estimation Errors, Mobility, and Multiple Paths. IEEE Transactions on Communications 47(2), 199–207 (1999)
20. Wódczak, M.: On the Adaptive Approach to Antenna Selection and Space-Time Coding in Context of the Relay Based Mobile Ad-hoc Networks. In: XI National Symposium of Radio Science URSI, pp. 138–142 (April 2005)
21. Wódczak, M.: On Routing information Enhanced Algorithm for space-time coded Cooperative Transmission in wireless mobile networks. PhD thesis, Faculty of Electrical Engineering, Poznan University of Technology, Poland (September 2006)
22. Wódczak, M.: Extended REACT Routing information Enhanced Algorithm for Cooperative Transmission. IST Mobile and Wireless Communications Summit (June 2007)
23. Wódczak, M.: Autonomic Cooperative Networking. Springer, New York (2012)
24. Zimmermann, E., Herhold, P., Fettweis, G.: On the Performance of Cooperative Relaying in Wireless Networks. European Transactions on Telecommunications 16(1), 5–16 (2005)

Protocol Design for Farm Animal Monitoring Using Simulation

Shikha Sarkar, Lina Stankovic, and Ivan Andonovic

Department of Electronic and Electrical Engineering, University of Strathclyde,
204 George Street, G1 1XW, Glasgow, United Kingdom
shikha.sarkar@eee.strath.ac.uk

Abstract. This paper presents a new simulation tool for performance analysis of wireless sensor networks (WSN) deployed on farm animals. The mobility and herding patterns from real herds are fed into statistical models to give rise to network simulation that is based on accurate herd behavior. The simulation results are used in evaluation of novel protocol ideas customized to the needs of farm monitoring.

1 Introduction

Animal rearing is a profitable but challenging business. On the one hand, owing to the overlap with public health and epidemiological concerns, its practice is highly regulated, and on the other hand owing to its fungible and perishable products, these businesses always remain under tight economic competition. Any technological advance that can enhance safety and competitiveness would be readily embraced by this industry. Monitoring health and estrus of cattle in large herds is quite labor intensive and its efficiency can be much enhanced by WSN deployed on the cattle. In this paper we present a simulation model for performance analysis of large scale deployments of this application of WSN. Cattle herds have their own characteristic peculiarity in the spatial spread of nodes and their mobility patterns. This fact ensures that the commonly used spread and mobility models [1]are not directly applicable in performance modeling specific to this application. A simulation model tailored for this application must also allow for: (i) directional (vs. classical omnidirectional assumption) antenna propagation from WSN nodes mounted as cow-collars due to RF absorption by the cow's body, (ii) movement limitations in a typical farm such as water holes, fences, etc. (iii) model augmentation using herd behavior captured from satellite images and GPS tracked data. We noted that these custom features were not readily implementable on the existing network simulators (surveyed in [2]). Based on these requirements, we have developed a novel discrete event based network simulator, WSNSIM, for performance analysis of this class of wireless networks, and used it to evaluate new protocol ideas for cattle herd monitoring.

2 Related Work

This section will outline the state of the art that this work builds on. There are three branches of predecessors of this work (i) that of WSN simulators (ii) that of WSN

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 126–138, 2012.

protocols and (iii) that of farm animal behavior relevant to WSN performance. So the following subsections survey related past work on these three aspects.

2.1 Related Work on WSN Protocol Designs

The LEACH (Low Energy Adaptive Clustering Hierarchy) protocol [3] is one of the most cited protocols as this work led to a whole family of protocols that were based on LEACH in some way. This paper is often cited also because it gives mathematical formulae for power consumption in transmission and reception. These formulae have been used by many subsequent works, including WSNSIM. An improvement to LEACH is proposed by the DAC algorithm [4], which seeks to improve the spatial spread of cluster-heads by asynchronous pre-emptive cluster election. This work also introduces a metric for efficacy of a routing, expressed in terms the ratio of clustered transmission length to the transmission length of direct communication to the base station. This ratio represents the power efficiency of the network topology used for transmission. A subsequent work [5] introduced a protocol called KMMDA (K-Means-like Minimum Mean Distance Algorithm) which seeks to improve LEACH by improving the spatial clustering. The protocol presented in [6]is quite similar to KMMDA. The PEGASIS (Power-efficient GAthering in Sensor Information Systems) protocol [7] presents another improvement over the LEACH protocol. In PEGASIS, the nodes form chains from sensor nodes so that each node transmits and receives from a neighboring node, and each member of the chain backbone has a designated next-forwarder. The STEM protocol presented in [8] proposes the key idea of a low duty-cycle periodic wakeup interspersed by long sleep (a low energy non-radio-listening state) intervals. In STEM, the data channel is different from the wakeup channel. The MR-MAC protocol [9] is similar protocol to STEM, but it uses two different channels of widely different frequency bands – a low data-rate, low frequency channel for exchange of wakeup and control packets.

2.2 Related Work on Network Simulators

The simulator NS3 [10] is a pure C++ library for which the simulation scenario is implemented as a ‘main’ function. The tool GTNetS [11] also supports the same method of usage, i.e., the simulation user writes the scenario as a main function using the facilities of the simulation class library. This method of user interfacing is not very user friendly for a WSN researcher. In WSN simulator it is best if some part of the input (especially the spatial node distribution) is defined graphically. It is also very useful if it is possible to visualize the mobility of nodes and network transmissions through an animation. In order to evaluate strategies like data gathering using a mobile collector node, it is useful to support interactive simulation – one in which the user can control simulation entities as the simulator runs. Another widely used simulator NS2 [12] is the precursor of NS3, and has the same usage model as NS3. It is difficult to customize it for WSN modeling. TOSSIM [13] is a discrete event simulator for TinyOS networks. It is tightly bound to the TinyOS capabilities and does not allow simulation of platform capabilities beyond that of TinyOS motes. This fact was quite limiting as the transmit function was not parameterized by the transmission range, or the receive functionality did not seem to record the received signal strength.

NetTopo [14] is a Java-based WSN simulator that provides both simulation and visualization functions to assist the investigation of WSN algorithms. The main limitation of NetTopo is that it is not an event based simulator. So the timing characteristics and collision behavior are not modeled by it. Such simulators are good for evaluating initial ideas, but for accurate estimation of protocol performance, one has to model the protocol at event level. The ATMEU [15] simulator provides simulation along with hardware emulation at a very low level; however, it is specific to the MICA2 platform and hence did not appear suitable for general freeform protocol research. There are two WSN simulators based on the open source OMNET++ framework [16], MIXIM [17] and Castalia [18]. The extension interface of Castalia and MIXIM involves programming the low level message-handler interface of OMNET++. WSNSIM introduces a simpler programming interface based on the new lambda notation in C++ (which is a formalism derived from the 'functional programming' model) for behavior specification, which is much more compact an easier to understand than the 'message handler call-back' method of Omnet++.

2.3 Related Work on Mobility Behavior Modeling

The paper [19] reports an experiment in which sizable cattle herds were recorded using navigation equipment and a probabilistic mobility model was made from the recorded data. Our colleagues have performed a similar experiment in a Scottish farm [20] and analyzed the movement patterns in an experimental herd to determine the connectivity of the network for various radio ranges. Another work [21] by the same group is based on the same experiment described in [20] and goes further to suggest implementation of protocols for herd management. It proposes duty cycles for transmission that is necessary to let the base station pick up data messages when the cattle briefly wander in its proximity. This paper does not implement a concrete protocol per se but proposes ideas and issues for protocol design. A third paper [22] from the same group reports experimental results from evaluation of two data gathering methods vis-à-vis cattle behavior: (a) A routing based data gathering and (b) a moving collector based data gathering. Mobility has a strong influence on the design, so it would be useful to incorporate into WSNSIM modular components representing various mobility models (e.g. vehicular networks in facilities such as ports [23]).

3 The WSNSIM Simulation Model

The architecture of WSNSIM is shown in Fig. 1 in terms of its functional components. The core of WSNSIM is written in C++ and the graphic user interface (GUI) is written in a popular scripting language called Tcl/Tk [24]. The simulation scenario is captured from interactively modeled user inputs in a data format called ".sim". The modeled scenario captured in the sim file includes such items as - the geometric shape of the rearing space (fences, prohibited regions etc.), initial herd configuration, directional antenna range, protocol parameters etc. WSNSIM uses image processing techniques to extract cattle positions from satellite photographs (as shown in Fig. 2) and uses statistical methods of distribution fitting to create a probabilistic model for generation of herd scenarios.

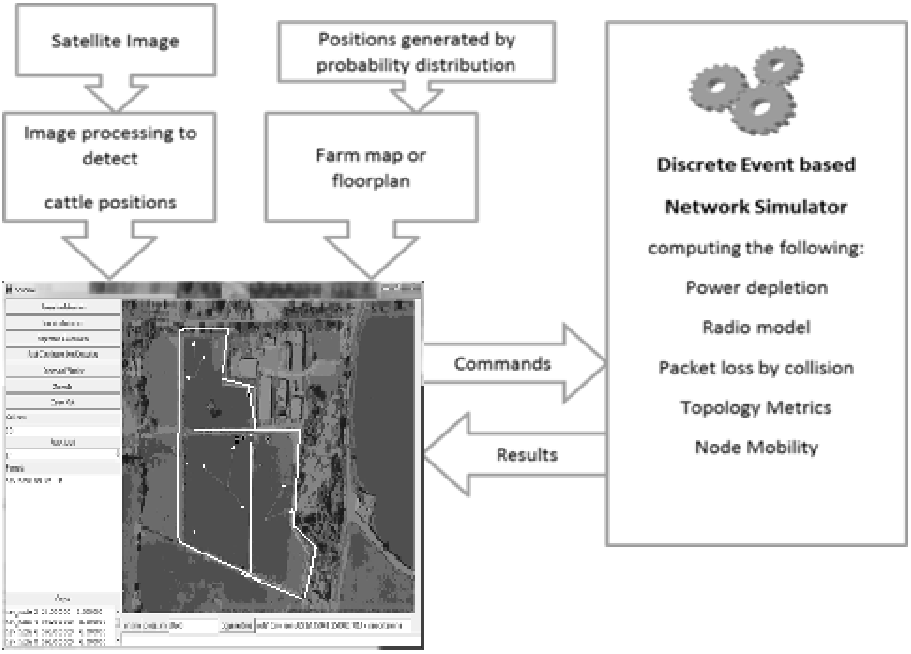
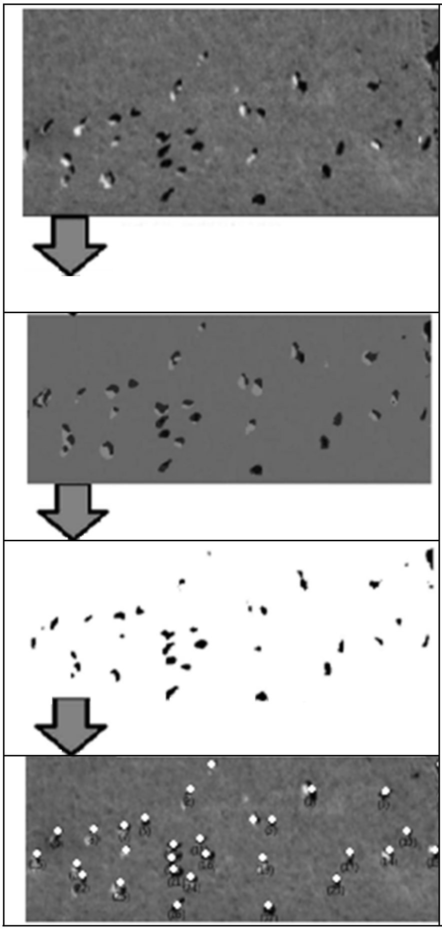


Fig. 1. The architecture of WSNSIM

WSNSIM supports definition of obstacles and fences (forbidden regions) in a two stage manner. Firstly one can define region annotations in the form of polygons. The orientation of the polygon is used to represent whether it bounds its inside or outside. The screenshot embedded in Fig. 1 shows the boundaries of a farm represented in WSNSIM. Previous investigation [25] has shown directional radio propagation properties for cow-collar transmitters. The antenna lies on one side of the cow's neck, which casts an electromagnetic shadow on the other side of the cow. Moreover, electromagnetic waves are primarily dipole radiations, so the wave propagation consists of lobes. In the presence of such directionality of transmitters, the simulator must take into account the shape of the directional range, in order to be accurate about receptions and interference. WSNSIM supports polygonal definition of antenna ranges, in addition to the default circular ranges.

A Markov chain (i.e. probabilistic state transition) based model of node mobility is superposed in WSNSIM with a discrete event simulation of network events to allow modeling of protocols. Radio models, packet loss models, and power depletion models are invoked alongside the discrete event simulation of the protocol activities.



Step 1. Initialise k clusters to represent k widely different colours that appear in such images. Each pixel is augmented with a data field called 'tag' to represent the cluster associated with the pixel (these are initially set to some invalid default that won't be confused with a cluster identifier).

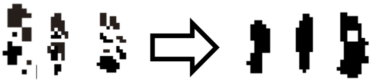
Step 2. Tag all the pixels by the cluster whose colour is closest to the colour of the respective pixel. If none of the pixels change their previously assigned tag, go to step 5.

Step 3. Re-compute the colour of each cluster as the mean (or centroid) of colours of all the pixels that has the respective cluster as its tag.

Step 4. Go to step 2

Step 5. Set pixels tagged with background-like clusters as white and those with cattle-like clusters as white.

This might produce several fragmented pixel chunks for each cow, which can be fixed using the morphological closure operator [26](as shown below)



Step 6. Finally detect the central position of each connected black region (blob).

Fig. 2. Algorithm for obtaining cattle spread data from satellite images

4 Statistical Model of Spatial Distribution

The spatial distribution of nodes has a strong influence on the protocol performance and thus an accurate model of the spatial distribution would improve the accuracy of protocol performance. The previous work in the literature assume a random or uniform distribution of nodes in a rectangular region, whereas we have formulated a statistical model of cattle distribution based on data collected from various herds captured from satellite images. A mathematical probability distribution has been fitted to the recorded distance of each cow from its four nearest neighbors in the herd. The best fit distribution was found to be the gamma distribution, which is similar to Gaussian distribution, but unlike a Gaussian variate ranges between $-\infty$ and ∞, the gamma variate ranges between 0 and ∞. Following is the density function for the gamma distribution:

$Gamma\_PDF(x) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{(\alpha-1)} e^{-\beta x}$ (Here α, β are parameters defining the distribution, and Γ in the normalizing denominator is the gamma function)

For our best fit, we had: α=2.208 (the shape parameter) and β=0.231 (the spread parameter). Fig. 3 shows a plot of the observed PDF alongside the density function of the best-fit gamma distribution.

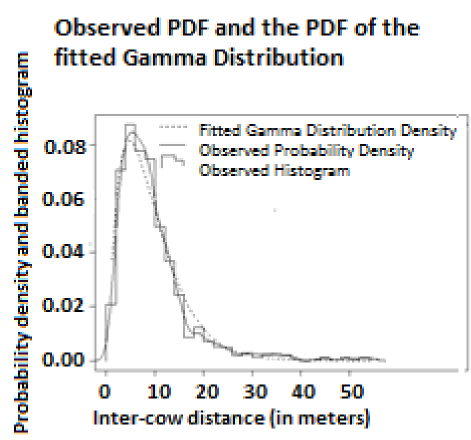
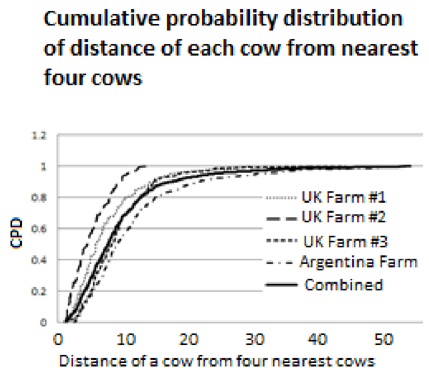
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>The fitted Gamma density function compared with the observed density function and the banded histogram.</td>
<td>Cumulative distribution of a cow's distance from four nearest cows.</td>
</tr>
</table>

Fig. 3. Probability density functions describing the composition of cattle herds

A structural recursion based growth algorithm was used for generating synthetic herds by sampling deviates from the fitted distribution. An intuitive view of that algorithm is to view the herd growth as a crystal growth scenario with atoms that can form up to four bonds with other atoms, but unlike atomic bonds, the cattle are not uniformly separated in angles. The 360° angle around a cow is divided into four 90° sectors; the herd is grown as new cows join the neighborhood of an existing cow in one of the unfilled 90° sectors at a distance sampled from the aforementioned fitted distribution. Following is an outline of the said algorithm:

```
Step 1. Start with one or more initial cows (seed cows).
Step 2. Choose a cow at random from the current herd which still has an empty neighborhood sector.
Step 3. Sample a distance from the said gamma distribution.
Step 4. Place a new cow at that distance in the unoccupied sector and mark that sector of the chosen existing cow as occupied. Reject the new placement if it goes beyond the farm boundary.
Step 5. If the total required herd size is not reached, go to step 2, else the work is done.
```

5 Statistical Model of Cattle Mobility

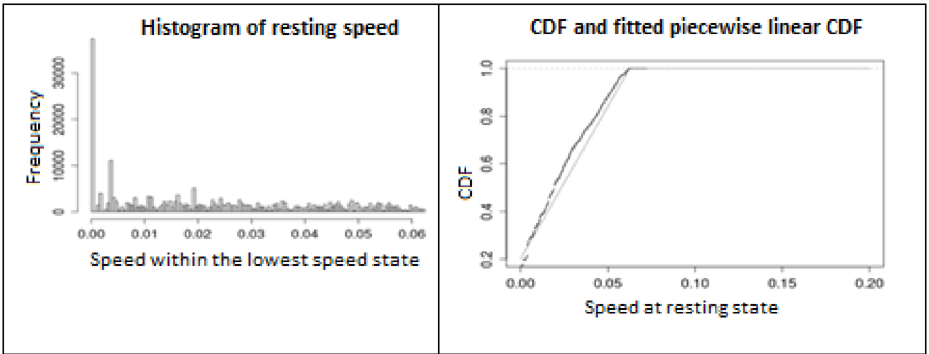
The satellite images are static snapshots of herds which tell us about the spatial spread and formations occurring in herds, but it does not tell us the patterns of mobility. In order to support the patterns of mobility, we have used timed position data recorded using GPS devices mounted on cattle collars. The movement behavior of cattle was modeled as a finite state machine with states defined in terms of speed and direction. Within each state the speed or direction follows a continuous probability distribution fitted from experimental data. The speed was partitioned into three states. The speed states were named as ‘resting’, ‘grazing’, and ‘shifting’ - and experimental speed values were snapped to each of the three cluster centers formed by k-means clustering in speed values. The three states were chosen based on subjective observation of the herd and on expert opinion. Resting is the phase in which the cow is completely immobile; grazing is the state in which the cow is very slowly mobile and in the process of feeding from the grass patches. Shifting is the state in which the cow moves from one place to another in a relative haste. It is customary to determine the number of clusters by minimizing the statistical measure called **mean index of adequacy** (MIA). Although the data for MIA (Table 1) shows k=2 to have the lowest MIA within a reasonable range of k, the number of speed states is not chosen as two as all the three states identified above are relevant for the cattle behavior, and the MIA for k=3 is quite close to that at k=2.

Table 1. Mean Index Adequacy vs. number of speed states from the herd data

Number of clusters (k)	Mean index of adequacy (MIA)
2	0.1037653
3	0.106014
4	0.1243418

Number of clusters (k)	Mean index of adequacy (MIA)
5	0.1105193
6	0.1524412
7	0.1410835

Statistical distributions were fitted to speeds within each cluster. Fig. 4 shows the speed distribution in the resting state (the lowest speed cluster). A piecewise linear equation was used to approximate the inverse function of its CDF (as shown in light gray).

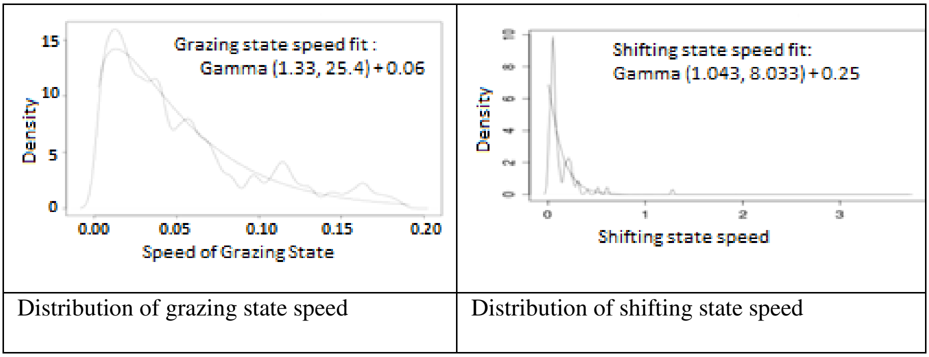


Histogram of lowest-speed cluster in speeds data

CDF of lowest speed cluster in speeds data, along with the fitted piecewise linear CDF.

Fig. 4. Speed distribution in the resting state

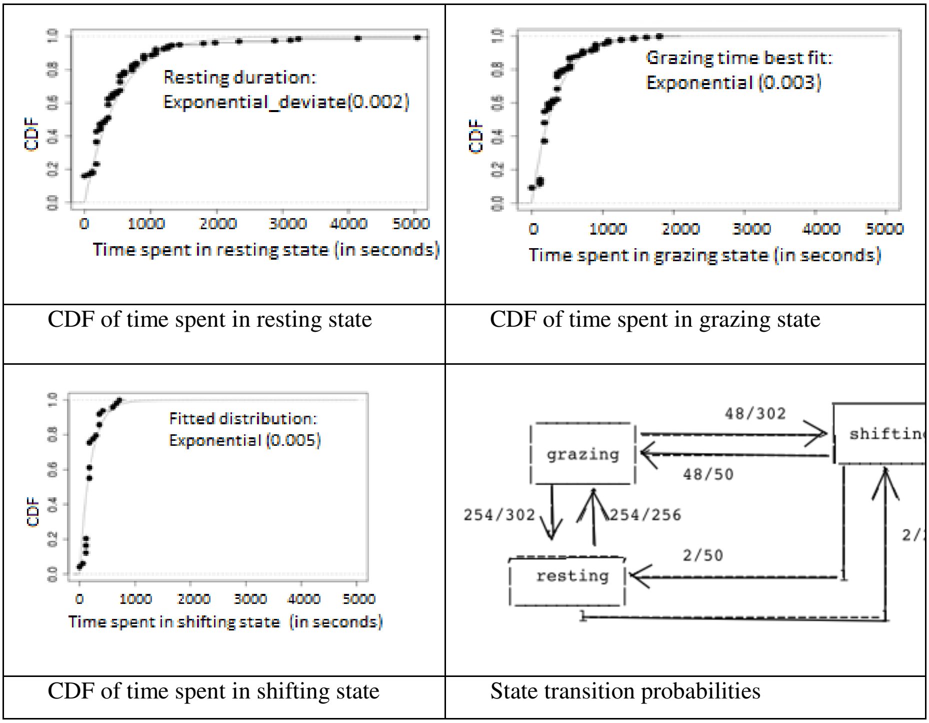
The speeds in the two other states are modeled as shifted Gamma distributions. The density plots with statistical estimates and the best-fit curve are given in Fig. 5. Another important aspect of the model is the amount of time spent in each speed state. Exponential distributions were found to be good fits for these variables. Fig. 6 shows plots of the dataset and of the best-fit exponential distributions. State transition probabilities are computed as statistical conditional probabilities derived from the transitions data (the values are shown in the bottom right panel of Fig. 6).



Distribution of grazing state speed

Distribution of shifting state speed

Fig. 5. Speed distribution in the grazing and shifting states



CDF of time spent in resting state

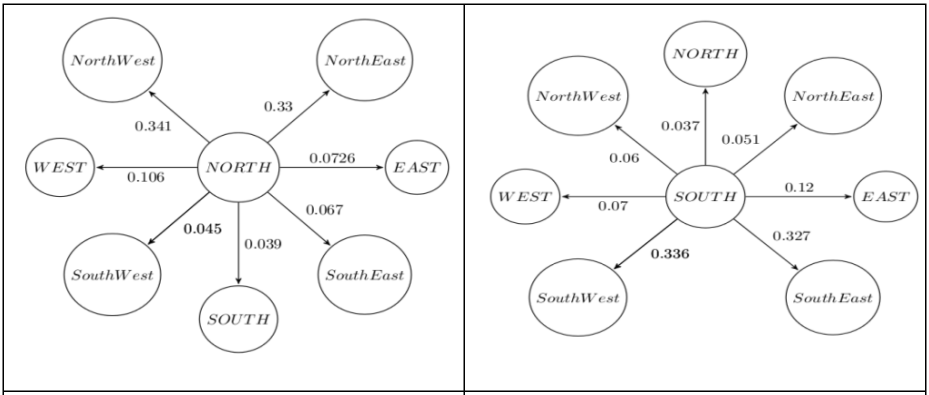
CDF of time spent in grazing state

CDF of time spent in shifting state

State transition probabilities

Fig. 6. CDFs and state transition probabilities

Heading directions were lumped into the eight directional states (viz, East, North-East (i.e. 45 degree north of east), North, North-West, West, South-West, South, and South-East) and the transition probabilities were obtained from the GPS tracked dataset. Fig. 7 shows the probabilities of direction transition, in which the arrow direction represents the direction of the state transition, with the probability of the transition noted alongside the arrow.



Transition Probabilities from Heading North

Transition Probabilities from Heading South

Fig. 7. Transition probabilities of some heading directions

The transition probabilities were calculated based on the GPS dataset used towards [20] using a simple conditional probability calculation. The probability of transition between to state S_2 given that current state is S_1 is calculated as:

$P(S_2 \mid S_1) = \frac{N(S_1,S_2)}{\sum_k N(S_1,S_k)}$; Here $N(S_i, S_j)$ stands for the number of times the GPS dataset had a transition from S_i to S_j.

6 Novel Protocols Designed Using WSNSIM

Most WSN protocols are data centric, in which the network transmits sensor readings to data-sink nodes or base stations. We have borrowed ideas from such data centric protocols as LEACH [3] and STEM [8] and evaluated a protocol that addresses the shortcomings of either taken in isolation. LEACH has the shortcoming that there is nothing in the protocol to ensure spatial spread of cluster centers. As a result, often the distributions of cluster-heads get skewed leaving a large number of nodes unreachable (and un-listened-to). Our modification of LEACH involves inference of spatial clusters through the first few communication rounds so that each node recalibrates its self-election probability according the estimated size of its local neighborhood. The self-election probability is corrected to 1/ (size of local neighborhood), so that small isolated sub-herds in space don’t get frequently excommunicated due to spatial isolation. Fig. 8 shows the results on evaluating this LEACH variant using WSNSIM. The number of excommunicated nodes reduces dramatically without compromising on the power consumption.

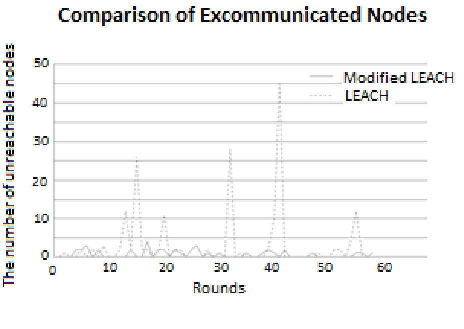
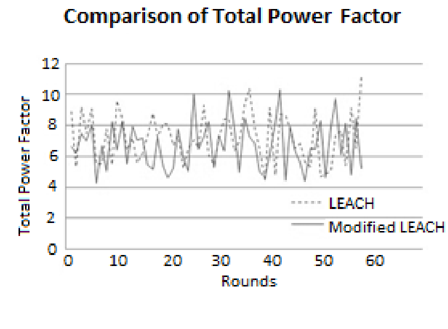
	
Comparison of the number of excommunicated nodes in LEACH and the Modified LEACH	Comparison of the total power factors in LEACH and the Modified LEACH

Fig. 8. LEACH vs. Modified LEACH

In LEACH, the nodes wake-up from low-power state to high-power state based on clocked timers. These timers work fine as long as the time-to-wakeup is not high enough to thwart synchronization due to clock skew. So LEACH rounds are closely spaced in time, but this is wasteful in the context of herd monitoring because the monitoring need not be up-to-date up to the second. It is just fine to get a snapshot of the herd condition every several minutes. So we developed a STEM-like protocol (named LEMSYP) that uses an idle duty-cycle to monitor the channel for a wakeup signal. The wakeup signal is a train of small packets broadcast with high power from a non-power-constrained central transmitter. In this protocol the nodes will normally stay in a low duty-cycle (LDC) periodic-listening idle phase, from which nodes may be awakened by a train of beacons from the base station. The train is long enough to accommodate at least one listening phase of a duty cycle, and each node's subsequent wake-up time would be synchronized according to the serial number of the received wakeup packet. Thus LEMSYP manages to synchronize just fine despite having a much longer time between data collection rounds, thereby reducing the power consumption dramatically. Fig. 9 shows the state diagram of the LEMSYP protocol.

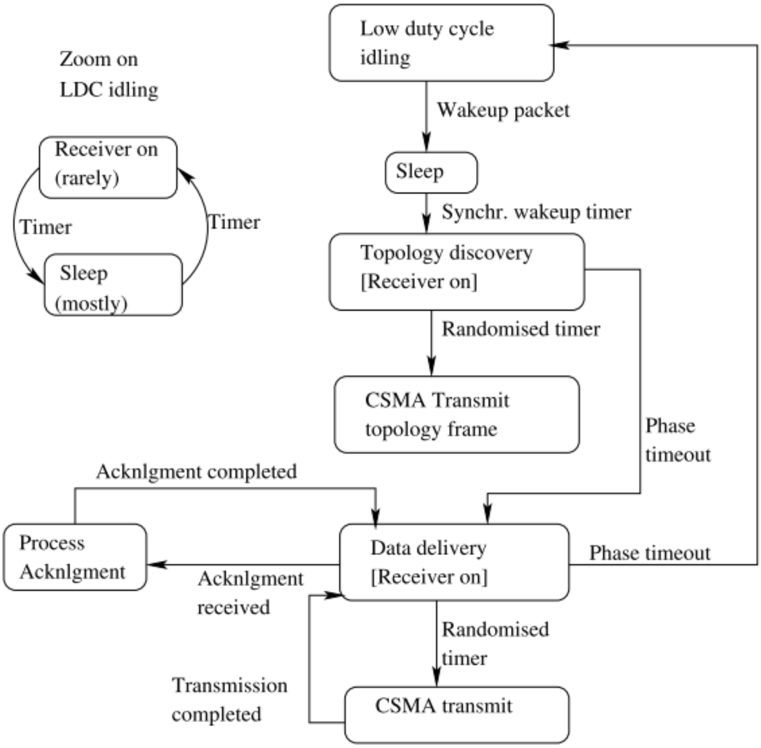


Fig. 9. State diagram of the LEMSYP protocol

The results of simulation, comparing the power performance of LEACH and LEMSYP, are shown in Fig. 10.

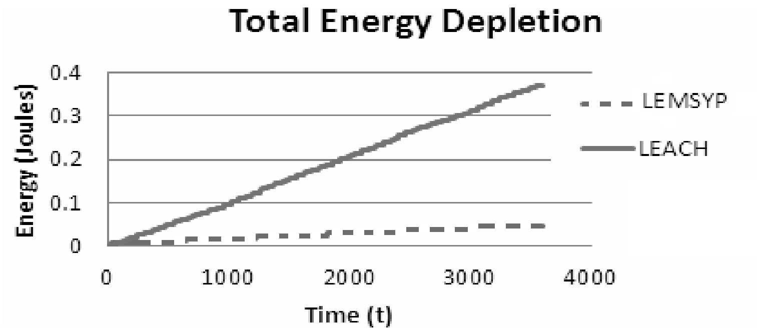


Fig. 10. Comparison of energy consumption by LEACH and LEMSYP

7 Verification and Validation

The WSNSIM simulator was verified using a set of unit-tests representing well understood and simple scenarios for which certain aspects of the results are known by analytical calculation. To facilitate this, R [27] bindings were written for the elementary units of WSNSIM functionality. The unit-tests are Tcl and R scripts that can be executed and report a failure if the understood outcome is not met. Some tests make randomized and periodic transmissions from static nodes for which the packet collision rate can be estimated without simulation. There are some tests for which the output is a stream of random variates that are fed into R as a data-vector and its distribution fitted and compared against the expected distribution parameter. For example, there is a test script that generates herd positions using the structural recursion algorithm described earlier in this paper. The generated herd positions are taken and the distances between each generated node and its four nearest neighbors recorded. This data is then fed into R to verify that the distribution agrees with the parameters obtained from satellite images.

8 Conclusion

WSNSIM is a simulation tool for design and performance analysis of wireless sensor network protocols applied to livestock monitoring and designed from real data. Simulation results from real cow distributions indicate that WSNSIM can be easily used in comparing the key performance aspects of data gathering WSN protocols. New protocols were proposed and evaluated using WSNSIM that show good improvement over the state of the art with regard to farm requirements.

References

[1] Wehrle, K., Gross, J.: Modeling and Tools for Network Simulation. Springer, Heidelberg (2010)

[2] Sundani, H., Li, H., Devabhaktuni, V.K., Alam, M., Bhattacharya, P.: Wireless Sensor Network Simulators: A Survey and Comparisons. International Journal of Computer Networks (IJCN) 2(6), 249–265 (2011)

[3] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: 33rd Annual Hawaii Int. Conf. on System Sciences, Hawaii (2000)

[4] Hebden, P., et al.: Distributed Asynchronous Clustering for Self-Organisation of Wireless Sensor Networks. In: 4th Int. Conf. on Intelligent Sensing and Information Processing, Bangalore, India (2006)

[5] Peng, W., Edwards, D.J.: K-Means Like Minimum Mean Distance Algorithm for Wireless Sensor Networks. In: 2nd International Conference on Computer Engineering and Technology, Chengdu, China (2010)

[6] Tan, L., Gong, Y., et al.: A balanced parallel clustering protocol for wireless sensor networks using k-means techniques. In: SENSORCOMM 2008, France (2008)

[7] Lindsey, S., Raghavendra, C.: PEGASIS: power efficient gathering in sensor information systems. In: IEEE Aerospace Conference, Big Sky, Montana, USA (2002)

[8] Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.: Optimizing sensor networks in the energy-latency-density design space. IEEE Transactions on Mobile Computing 1(1), 70–80 (2002)

[9] Ansari, J., Zhang, X., Mahonen, P.: Multi-radio Medium Access and Control Protocol for Wireless Sensor Networks. International Journal of Sensor Networks 8(1), 47–61 (2010)

[10] Henderson, T., Lacage, M.: Network simulations with the ns-3 simulator. In: SIGCOMM Demonstration, Seattle, USA (2008)

[11] Riley, G.F.: The Georgia Tech Network Simulator. In: ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research, Karlsruhe, Germany (2003)

[12] Fall, K., Varadhan, K.: NS2 Manual, http://www.isi.edu/nsnam/ns/doc/index.html (accessed April 14, 2012)

[13] Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: accurate and scalable simulation of entire TinyOS applications. In: Proc. 1st Int. Conf. on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles, USA (2003)

[14] Shu, L., Hauswirth, M., Chao, H.C., Chen, M., Zhang, Y.: NetTopo: A framework of simulation and visualization for wireless sensor networks. Ad Hoc Networks 9(5), 799–820 (2011)

[15] Polley, J., Blazakis, D., McGee, J., Rusk, D., Baras, J.S.: ATEMU: A Fine-grained Sensor Network Simulator. In: IEEE SECON, Santa Clara, California, USA (2004)

[16] Varga, A.: The OMNeT++ distrete event simulation system (1999), http://www.omnetpp.org/

[17] Kopke, et al.: Simulating wireless and mobile networks in omnet++: The mixim vision. In: First International OMNeT++ Developers Workshop, Marseille, France (2008)

[18] Boulis, A.: Castalia, a simulator for wireless sensor networks and body area networks (May 2009), http://castalia.npc.nicta.com.au (accessed May 14, 2009)

[19] Guoa, Y., Poultona, G., et al.: Using accelerometer, high sample rate GPS and magnetometer data to develop a cattle movement and behaviour model. Ecological Modelling 220(17), 2068–2075 (2009)

[20] Kwong, K.H., Wu, T.-T., Goh, H.G., et al.: Practical considerations for wireless sensor networks in cattle monitoring applications. Computers and Electronics in Agriculture 81, 33–44 (2012)

[21] Kwong, K.H., Wu, T.T., Goh, H.G., Sasloglou, K., Stephen, B., Glover, I., Shen, C., Du, W., Michie, C., Andonovic, I.: Implementation of herd management systems with wireless sensor networks. IET Wireless Sensor Systems 1(2), 55–65 (2011)

[22] Cao, D., Wu, T., Goh, H.G., Stephen, B., Kwong, K., Michie, C., Andonovic, I.: Exploitation of Wireless Telemetry for Livestock Condition Monitoring. In: Canadian Society for Bioengineering (CSBE/SCGAB) CIGR, Qubec City (2010)

[23] Majumder, J., Vassalos, D., Sarkar, S., Kim, H., Guarin, L., York, A., Dahlberg, T.: Simulation based planning of ferry terminal operations. In: Bertram, V. (ed.) Proceedings 6th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT 2007), Cortona (2007)

[24] Tcl developer site, http://www.tcl.tk/

[25] Stephen, B., Dwyer, C., Hyslop, J., Bell, M., Ross, D., Kwong, K., Michie, C., Andonovic, I.: Statistical Interaction Modeling of Bovine Herd Behaviors. IEEE Transactions on Systems, Man, and Cybernetics 41(6), 820–829 (2010)

[26] Tombre, K., Lamiroy, B.: Graphics recognition - from re-engineering to retrieval. In: Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, UK (2003)

[27] R - an open source statistical computing platform and environment, http://www.r-project.org/

Minimum Latency Aggregation Scheduling for Arbitrary Tree Topologies under the SINR Model

Guanyu Wang, Qiang-Sheng Hua, and Yuexuan Wang

Institute for Interdisciplinary Information Science, Tsinghua University, China
{wang-gy10,qshua,wangyuexuan}@mail.tsinghua.edu.cn

Abstract. Almost all the existing wireless data aggregation approaches need a topology construction step before scheduling. These solutions assume the availability of flexible topology controls. However, in real scenarios, lots of factors (impenetrable obstacles, barriers, etc.) limit the topology construction for wireless networks. In this paper we study a new problem called Minimum-Latency Aggregation Scheduling for Arbitrary Tree Topologies (MLAT). We first provide an NP-hardness proof for MLAT. Second, we draw an important conclusion that two frequently used greedy scheduling algorithms result in a large overhead compared with the optimal solution: the scheduling latency generated by these two greedy solutions are $\sqrt{n}$ times the optimal result, where n is the total number of links. We finally present an approximation algorithm for MLAT which works well for the tree with a small depth. All the above results are based on the SINR (Signal-to-Interference-plus-Noise Ratio) model.

1 Introduction

Data aggregation is a fundamental operation in wireless sensor networks. Given a set of sensor nodes distributed on the Euclidean plane, the data aggregation problem is to compute an aggregate function (e.g. a maximum or average function) on the data from all nodes in the wireless sensor network, and let the final aggregated value to be sent to a sink node in the fewest timeslots. To solve the data aggregation problem, also called as the $MLAS$ (Minimum-Latency Aggregation Scheduling) problem in the literature, the interference models employed will play an important role. Compared with exceedingly simplified graph based models or the protocol models used in many previous studies of data aggregation [1,7,17,19], a more realistic SINR (Signal-to-Interference-plus-Noise-Ratio) interference model [3] has been widely adopted in the community [12,11,10,4]. The SINR model is also called the physical model since it reflects the physical reality more accurately. The advantages and robustness of the SINR model are analyzed in [14]. In this paper, we employ the SINR model to study the data aggregation problem for arbitrary tree topologies.

For the MLAS problem, the best result to date under the SINR model is $O(\log n)$ given by Halldórsson et al [4]. There is a hardness result for the MLAS

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 139–152, 2012.

with uniform power assignment in [10]. The data aggregation problem for arbitrary directed acyclic networks under the SINR model is also studied in [6]. In this paper, the authors first show that this problem is NP-hard and give both heuristic and approximation algorithms. Note that the NP-hardness result for the directed acyclic networks may not mean the same hardness result for the tree topologies and the latter is the most frequently used topology for data aggregation. In addition, compared with the directed acyclic networks [6], one may get better scheduling results for the restricted tree topologies.

Also by using the SINR model, some other related wireless scheduling problems have been studied in the literature. Moscibroda et al. in 2006 [14] first initiated the connectivity scheduling problem (to construct a spanning tree over a set of sensor nodes on the plane in the fewest number of timeslots). This kind of connectivity scheduling problem has been further studied and better results have been proposed in [15,13,16]. The NP-hardness of the One-Shot scheduling problem (to pick the maximum number of links to be scheduled in the same timeslot) with uniform power (all the nodes take the same power) was proposed by Goussevskaia et al.[16]. This result was extended to the non-uniform power version later [8]. Very recently, some further hardness results have been given: Halldórsson and Wattenhofer [5] proved that One-Shot scheduling with uniform power assignment is in APX (the set of NP optimization problems that allow constant-factor approximation algorithms) and Kesselheim [9] extended the result to the power control version.

Note that, the typical solution for MLAS involves the construction of an appropriate data aggregation tree, followed by scheduling its transmission links. For example, the nearest neighbor tree is one of the widely used topologies. However, in a dynamic physical environment, it can not be guaranteed that any two nearest neighbors can communicate with each other successfully. This problem arises if there are obstacles or barriers restricting the kinds of links that can be formed between nodes that could otherwise be within communication range. Based on this observation, we study the Minimum Latency Aggregation Scheduling for Arbitrary Tree Topologies ($MLAT$). The only difference between MLAT and MLAS is that the tree topology is given in advance for the MLAT problem instead of first constructing a tree in the MLAS problem.

1.1 Formal Description of the MLAT Problem

We are given a tree consisting of nodes $V = \{v_0, v_1, v_2, \ldots, v_n\}$ with root v_0. We divide time into timeslots, defined to be the unit of time required to transmit once for any link. All the nodes are arbitrarily distributed in the Euclidean plane and can be both a sender and receiver, but only in different timeslots. The distance between any two nodes v_i, v_j is denoted by $d(v_i, v_j)$. Each edge $l_{ij} = (v_i, v_j)$ represents a communication request from a sender v_i to a receiver v_j. The length of link l_{ij} is denoted by $d_{ij} = d(v_i, v_j)$, where $d_{gj} = d(v_g, v_j)$ denotes the distance between the sender of link l_{gh} and receiver of link l_{ij}.

Formally, the SINR model is defined as follows. The signal power $P_i(j)$ received at v_j from sender v_i depends on the transmission power P_{ij} of v_i and the distance d_{ij}. The path loss radio propagation model for the reception of signals says the signal strength that v_j receives degrades at $d_{ij}^{-\alpha}$ (α denotes the path-loss exponent, and is usually a constant between 2 and 6), i.e. $P_i(j) = P_{ij}/d_{ij}^{\alpha}$. Every sender v_g (with corresponding receiver v_h) that sends concurrently with v_i causes an interference $I_g(j) = P_g(j) = P_{gh}/d_{gj}^{\alpha}$ at receiver v_j. All interferences accumulate. The total interference $I(v_j)$ experienced by receiver j is given as the sum of all interferences caused by other concurrently sending nodes, i.e. $I(v_j) = \sum_{l_{gh} \neq l_{ij}} I_g(j)$. A receiver v_j successfully receives a message from its sender v_i if and only if it obeys the *precedence* constraint (a node cannot send its data to the parent node until it has received data from *all* children nodes) and the following SINR threshold holds:

$$SINR_S(v_j) = \frac{P_i(j)}{\sum_{l_{gh} \in S \setminus l_{ij}} I_g(j) + N} \geq \beta$$

where N is ambient noise, $\beta \geq 1$ denotes the minimum SINR required for a message to be successfully received, and S is the set of concurrently transmitting links. We call the set SINR-feasible set. Denote all edges of the given tree as $E = \{l_1, l_2, \ldots, l_n\}$ (for notational simplicity, we omit the sender and receiver suffix here), we strive to find a sequence of t sets, i.e. a schedule: $S = \{L_1, L_2, \ldots, L_t\}$, $L_1 \cup L_2 \cup \cdots \cup L_t = E$ and $L_i \cap L_j = \emptyset$, $\forall i, j \in [t]$ $i \neq j$, such that:

$$S = \operatorname*{argmin}_{S' = \{L_1, L_2, \ldots, L_t\}} t$$

$$h > g \qquad \forall i, j, k \quad l_{ij} \in L_g \ and \ l_{jk} \in L_h,$$

$$SINR_{L_m}(v_j) \geq \beta, \ \forall m \ \forall l_{ij} \in L_m$$

1.2 Results

In Section 2, we will present the first NP-hardness proof for MLAT under two real conditions: (i) ambient noise exists, and (ii) all nodes have limited power ranges. We then analyze the gap between local optimal solution and global optimal solution for the MLAT Problem. Section 3 provides the evidence of that most local greedy approaches of existing aggregation scheduling algorithms perform poorly compared with the optimal result. The timeslots (scheduling latency) needed by the local optimal methods could be $\sqrt{n}$ times larger than the global optimal solution, where n is the total number of wireless links. We then present an approximation algorithm for MLAT in Section 4, which adopts an existing strategy of iteratively maximizing concurrently transmitting links. We derive the exact approximation ratio bounded by $O(\min\{d \cdot \log n, n/d\})$, where d is the depth of the given tree. Even though the greedy approaches have been proved to perform poorly, this analysis shows that it still has guaranteed efficiency for the tree with a small depth.

2 NP-Hardness Proof for MLAT

In this section, we will show the following NP-hardness result.

Theorem 1. *MLAT with given power range and nonzero background noise is NP-hard.*

Proof. Let $P_{v_i} \in [P_{min}, P_{max}]$ be the transmission power assigned to every sender v_i, and let $N > 0$.

We will give a polynomial time reduction that expands on methods used in [16,8] from the Partition Problem to the decision version of MLAT, when one must decide whether there exists a schedule of a specified length for a given aggregation tree.

Partition Problem: Do there exist sets $\mathcal{I}_1$, $\mathcal{I}_2 \subset \mathcal{I}$ where $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ is a set of integers s.t.

$$\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}, \mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset,$$

$$\sum_{i_j \in \mathcal{I}_1} i_j = \sum_{i_j \in \mathcal{I}_2} i_j = \frac{1}{2} \sum_{i_j \in \mathcal{I}} i_j = \frac{1}{2}\sigma.$$

This problem was proved to be NP-complete by Karp [2]. We construct a many-to-one reduction from an arbitrary Partition Problem instance to an instance of MLAT. We will argue that the instance of MLAT can be scheduled in $T \leq n+3$ timeslots if and only if the reduced Partition Problem instance can be solved.

Lemma 1. *The Partition Problem can be reduced to MLAT in polynomial time.*

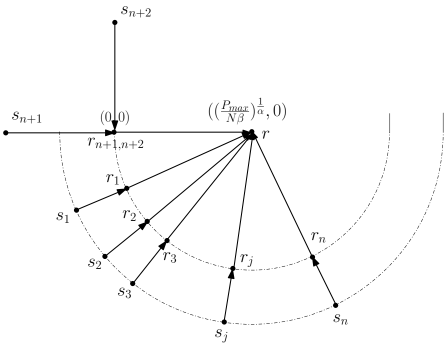


Fig. 1. Example of constructed instance of MLAT from Partition Problem

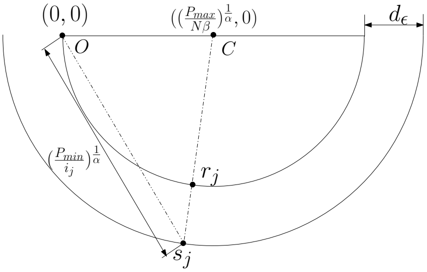

$(0,0)$
$((\frac{P_{max}}{N\beta})^{\frac{1}{\alpha}}, 0)$
d_ϵ
O
C
$(\frac{P_{min}}{i_j})^{\frac{1}{\alpha}}$
r_j
s_j

Fig. 2. The semicircle in the plane

Without loss of generality, we assume all elements in the Partition Problem instance $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ to be distinct and positive. Next, we construct an instance of MLAT with $2n+3$ links $L = \{l_1, l_2, \ldots, l_{2n+3}\}$ (cf. Fig. 1). We define the sender and receiver of link l_i as s_i and r_i, respectively.

To begin, we must scale all the integers in $\mathcal{I}$ by the same factor k such that $k \cdot i_{min} \geq \frac{N\beta P_{min}}{2^\alpha P_{max}}$, where the i_{min} and i_{max} (used in the definition of d_ϵ below) are the smallest and largest integers in $\mathcal{I}$, respectively. In the following we regard the set $\mathcal{I}$ to be properly scaled.

We assign every node a position in the Euclidean plane. First, we fix a semicircle of radius $R_1 = (P_{max}/(N\beta))^{1/\alpha}$ to the plane centered at some point C, followed by another semicircle with radius $R_2 = R_1 + d_\epsilon$ also centered at C, d_ϵ a small constant. Let I_{min} be defined as:

$$I_{min} = \min_{i_g, i_h \in \mathcal{I}, i_g \neq i_h} |(\frac{P_{min}}{i_h})^{\frac{1}{\alpha}} - (\frac{P_{min}}{i_g})^{\frac{1}{\alpha}}|.$$

We will show that for any small $\epsilon > 0$,

$$d_\epsilon = \min\{\frac{I_{min}}{(\frac{(1+\epsilon)n\beta P_{max}}{\epsilon P_{min}})^{\frac{1}{\alpha}} + 1}, (\frac{P_{min}}{N\beta(1+\epsilon)})^{\frac{1}{\alpha}}, (\frac{P_{min}}{i_{max}})^{\frac{1}{\alpha}}\}$$

is sufficiently small for our reduction.

For each integer $i_j \in \mathcal{I}$, we place sender s_j on the larger semicircle such that its distance from the leftmost point of the smaller semicircle (origin O) is $(P_{min}/i_j)^{1/\alpha}$ (cf. Fig.2). Because of our scaling of $\mathcal{I}$ and choice of d_ϵ, such a point will always exist.

$$d(s_j, O) = (\frac{P_{min}}{i_j})^{\frac{1}{\alpha}} \quad \forall 1 \leq j \leq n$$

Next, we designate the position for every receiver r_j, $1 \leq j \leq n$ to be the intersecting point on the smaller semicircle of the line which passes through both s_j and C. Note that the distance between any pair s_j, r_j, $1 \leq j \leq n$ is always d_ϵ.

Finally we place four nodes s_{n+1}, s_{n+2}, $r_{n+1,n+2}$ and r. Note that $r_{n+1,n+2}$ is the receiver corresponding to senders s_{n+1} and s_{n+2}. Node r is the parent node for all receivers $r_1, r_2, \ldots, r_n, r_{n+1,n+2}$ in the tree (cf. Fig. 1).

$$pos(s_{n+1}) = (-(\frac{P_{max}}{\beta(N + \frac{\sigma}{2})})^{\frac{1}{\alpha}}, 0), \quad pos(r_{n+1,n+2}) = (0, 0)$$

$$pos(s_{n+2}) = (0, (\frac{P_{max}}{\beta(N + \frac{\sigma}{2})})^{\frac{1}{\alpha}}), \quad pos(r) = ((\frac{P_{max}}{N\beta})^{1/\alpha}, 0)$$

Next $n + 3$ links of the tree are constructed as follows:

$$l_{n+1} = (s_{n+1}, r_{n+1,n+2}), \quad l_{n+2} = (s_{n+2}, r_{n+1,n+2})$$

$$l_{n+3} = (r_{n+1,n+2}, r), \quad l_{n+(i+3)} = (r_i, r) \;\; 1 \leq i \leq n$$

We then prove four properties of this tree:

1. l_{n+1}, l_{n+2} must transmit in different timeslots.
2. All l_i, $1 \leq i \leq n$, and one of l_{n+1}, l_{n+2} can transmit concurrently (i.e. in the same timeslot).
3. l_{n+1}, l_{n+2} can transmit successfully if and only if the total interference from other senders is not greater than $\sigma/2$.
4. l_j, $n+3 \leq j \leq 2n+3$, can only transmit alone.

The first property arises directly from the observation that l_{n+1} and l_{n+2} have a common receiver $r_{n+1,n+2}$. If they transmit in the same timeslot, there is no possibility that both of their SINR is greater than β.

The second property can be derived from the following lemma:

Lemma 2. *Every transmission $l_i \in L' = \{l_1, l_2, \ldots, l_n\}$ is successful using transmission power P_{min}, no matter how many other links $l_j \in L'$ along with either l_{n+1} or l_{n+2} transmit concurrently, even if all transmitting links, except for l_i, use power P_{max}.*

Proof. For links in L', the worst case scenario is that all senders s_i, $1 \leq i \leq n$, and either s_{n+1} or s_{n+2} transmit concurrently with P_{max}. Recall that we have chosen a very small d_ϵ (i.e. the distance between s_i and r_i, $1 \leq i \leq n$). It is easy to see that $d(s_{n+1}, r_{n+1,n+2}) = d(s_{n+2}, r_{n+1,n+2}) \geq d_\epsilon$. We can bound the distance between r_i and s_j, $1 \leq i \leq n, 1 \leq j \leq n+2, i \neq j$.

$$d(r_i, s_j) \geq d(s_j, s_i) - d(s_i, r_i) \geq |d(s_j, O) - d(s_i, O)| - d_\epsilon \tag{1}$$

$$\geq I_{min} - d_\epsilon \geq \left(\left(\frac{(1+\epsilon)n\beta P_{max}}{\epsilon P_{min}}\right)^{\frac{1}{\alpha}} + 1 - 1\right)d_\epsilon \tag{2}$$

$$= \left(\frac{(1+\epsilon)n\beta P_{max}}{\epsilon P_{min}}\right)^{\frac{1}{\alpha}} d_\epsilon \tag{3}$$

The first inequality follows from two triangle inequalities (cf. Fig.3). The second inequality follows from the definition of I_{min} and d_ϵ. Thus, we can derive an SINR lower bound for all receivers r_i, $1 \leq i \leq n$:

$$SINR(r_i) = \frac{\frac{P_{s_i}}{d(s_i, r_i)^\alpha}}{N + \sum_{l_j \in L'/l_i \cup \{l_{n+1} \text{ or } l_{n+2}\}} \frac{P_{s_j}}{d(r_i, s_j)^\alpha}}$$

$$\geq \frac{\frac{P_{min}}{d_\epsilon^\alpha}}{N + \frac{nP_{max}}{d(s_j, r_i)^\alpha}} = \frac{\frac{P_{min}}{d_\epsilon^\alpha}}{N + \frac{\epsilon P_{min}}{(1+\epsilon)d_\epsilon^\alpha \beta}}.$$

On the other hand, according to the value of d_ϵ, we know $d_\epsilon \leq (P_{min}/(N\beta(1+\epsilon)))^{1/\alpha}$. So we obtain $N \leq \frac{P_{min}}{(1+\epsilon)d_\epsilon^\alpha \beta}$.

Combining these, we get

$$SINR(r_i) \geq \frac{\frac{P_{min}}{d_\epsilon^\alpha}}{\left(\frac{P_{min}}{(1+\epsilon)d_\epsilon^\alpha \beta}\right) + \frac{\epsilon P_{min}}{(1+\epsilon)d_\epsilon^\alpha \beta}} = \beta.$$

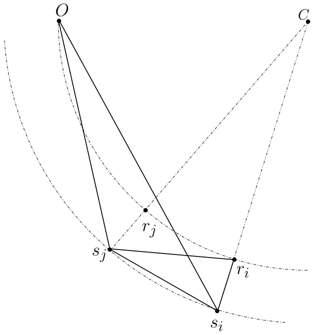


Fig. 3. The distance between r_i and s_j

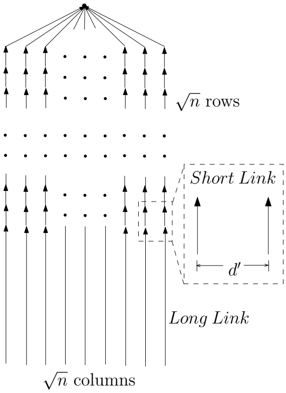


Fig. 4. Counterexample for the Leaf-First approach

The third property can be derived from the total interference suffered by $r_{n+1,n+2}$ from s_k, $1 \leq k \leq n$. When these senders transmit with minimum power P_{min},

$$I_{r_{n+1,n+2}}(s_k) = \frac{P_{min}}{((\frac{P_{min}}{i_k})^{\frac{1}{\alpha}})^{\alpha}} = i_k.$$

However, even if s_{n+1} uses transmission power P_{max}, we have:

$$P_{s_{n+1}}(r_{n+1,n+2}) = \frac{P_{max}}{((\frac{P_{max}}{\beta(N+\frac{\sigma}{2})})^{\frac{1}{\alpha}})^{\alpha}} = \beta(N + \frac{\sigma}{2}).$$

If we want s_{n+1} to transmit successfully, the following inequality must hold:

$$\frac{P_{s_{n+1}}(r_{n+1,n+2})}{N+I} = \frac{\beta(N+\frac{\sigma}{2})}{N+I} \geq \beta.$$

It is easy to see that if s_{n+1} transmits using a smaller power or if other senders transmit using larger powers, then the SINR balance will be destroyed. The same analysis also holds for s_{n+2}. Thus the following lemma can be derived from these three properties:

Lemma 3. *There exists a 2-slot schedule for all links in $L'' = \{l_1, l_2, \ldots, l_{n+2}\}$ if and only if there is a solution to instance $\mathcal{I}$ of the Partition Problem.*

Proof. By the second property, we only need to consider l_{n+1}, l_{n+2}.

If $\{\mathcal{I}_1, \mathcal{I}_2\}$ is a solution to $\mathcal{I}$, then $\sum_{i_j \in \mathcal{I}_1} i_j = \sum_{i_k \in \mathcal{I}_2} i_k = \sigma/2$. This means we can let l_{n+1} and all l_j, $\forall i_j \in \mathcal{I}_1$ transmit concurrently in the first timeslot, and l_{n+2}, l_k, $\forall i_k \in \mathcal{I}_2$ transmit in the second timeslot. The correctness of this schedule is guaranteed by the third property.

From the first property, if there is a 2-slot schedule for L'', l_{n+1}, l_{n+2} must transmit in different timeslots. Without loss of generality, we assume l_{n+1} transmits in the first timeslot, l_{n+2} in the second. Let L_1 and L_2 be the sets of links transmitting in first and second timeslot, respectively. According to the third property, $\sum_{l_j \in L_1} i_j \leq \sigma/2$, and $\sum_{l_k \in L_2} i_k \leq \sigma/2$. However, we already know $\sum_{l_j \in L_1 \cup L_2} i_j = \sigma$. So the following equation holds:

$$\sum_{l_j \in L_1} i_j = \sum_{l_k \in L_2} i_k = \sigma/2$$

which means we have a solution for the Partition Problem instance $\mathcal{I}$.

The fourth property follows naturally. Since the lengths of all l_j, $n+3 \leq j \leq 2n+3$ are $(P_{max}/(N\beta))^{1/\alpha}$, i.e., the radius of the smaller semicircle, receivers $r_j \in \{r_1, r_2, \ldots, r_n, r_{n+1,n+2}\}$ become senders with transmission power P_{max}. Then $P_{r_j}(r) = (P_{max})/((P_{max}/N\beta)^{1/\alpha})^\alpha = N\beta$, which means any other additional interference will make $SINR_{r_j}(r)$ fall below the threshold β, i.e., l_j, $n+3 \leq j \leq 2n+3$, can only transmit alone.

Combining all four properties, we conclude that the constructed instance of MLAT can be scheduled in $T \leq n+3$ timeslots if and only if the reduced Partition Problem instance can be solved. Therefore, if we have a polynomial time algorithm A for MLAT, then we may also solve the Partition Problem using A as a subroutine in polynomial time.

3 Gap between the Local and Global Optimization

In this section, we will show that two greedy approaches (layer-first and leaf-first) result in very poor schedules: the scheduling latencies generated by greedy solutions could be $\sqrt{n}$ times the optimal result, where n is the total number of links. Note that most existing data aggregation scheduling algorithms use the greedy ideas after the topology construction step, even the best $O(\log n)$ result for the MLAS problem [4]. This may give some hint that using an appropriate topology construction algorithm could help reducing the scheduling latency for the data aggregation problem.

3.1 Leaf-First Method

Assume we have a black box which can find the maximum size set of concurrently transmitting links, from all the given links. This black box is used for *greedily select* operation in this section. We want to show that even we can find the local optimal concurrent transmissions using this black box, it still leads to a very poor performance compared with the global optimal solution in the worst case.

Definition 1. *For any given data aggregation tree defined in Section 1.1, in any round, greedily select the leaves of the tree (i.e. choose the maximum number of links that can transmit simultaneously) at the beginning of that round to transmit, without violating the SINR threshold. This approach is called Leaf-First.*

Theorem 2. *Given a data aggregation tree with n links, assume the output of the Leaf-First approach is the schedule S_{leaf} and the minimum-latency schedule of this tree is S_{opt}. In the worst case, $|S_{leaf}| = \Omega(\sqrt{n})|S_{opt}|$.*

Proof. Construct the data aggregation tree with $\sqrt{n}$ layers and every layer consists of $\sqrt{n}$ links, as shown in Fig.4. Except for the links in the top layer, this tree has only two types of links: *long link* and *short link*. The long links only appear in the deepest layer and their lengths are all $d_{long} = (\frac{P_{max}}{N\beta})^{1/\alpha}$. It is easy to show that every long link can only transmit alone since $\frac{P_{max}/d_{long}^{\alpha}}{N} = \beta$, i.e. any additional interference from other senders let the transmission of a long link fail. The remaining links are short links which is very short compared with distance (denoted as d' in the figure) between any two short links such that all the short links in the same layer can transmit concurrently. Simply set their length as $d_{short} = (\frac{P_{min}}{\beta(\sqrt{n}P_{max}/d'^{\alpha}+N)})^{\frac{1}{\alpha}}$, then the required property above for short link holds.

After the construction, we apply the Leaf-First approach on this data aggregation tee. Its performance is shown in Fig.5. Obviously, the Leaf-First approach needs $\Theta(n)$ timeslots in total to finish the aggregation. However, a much better aggregation should be finishing all the long links one by one and then short links layer by layer, which results in a $2\sqrt{n} - 2$ time-slot schedule, and all the links in the top layer still needs extra $\sqrt{n}$ timeslots. Therefore, $3\sqrt{n} - 2$ timeslots is enough to finish the aggregation. So we get $|S_{leaf}| = \Omega(\sqrt{n})|S_{opt}|$.

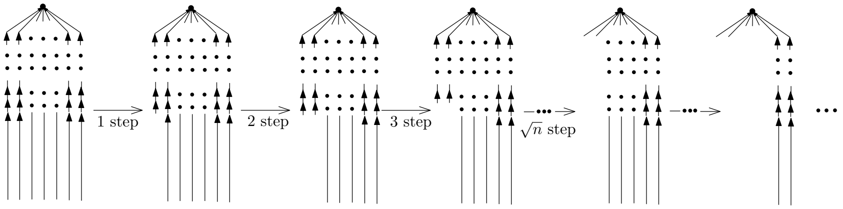


Fig. 5. Performance of the Leaf-First approach on counterexample

3.2 Layer-First Approach

For simplicity, we assume $P_{max} = P_{min} = P$ and $\beta = 1$ in this part, i.e., adopting the special case –uniform power assignment– in the following analysis. In addition, the ambient noise is so small compared with P that it can be ignored. We give the formal definition of the Layer-First approach like above.

Definition 2. *For any given data aggregation tree defined in Section 1.1, in any round, greedily select the wireless links belong to the deeper layers (i.e. transmit the deepest links as many as possible, then the second deepest ones, and so on) at the beginning of that round to transmit, without violating the SINR threshold. This approach is called Layer-First.*

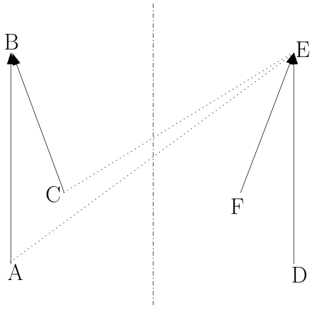


Fig. 6. Example of counter-block links

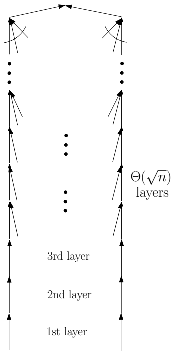


Fig. 7. Counterexample for the Layer-First approach

Theorem 3. *Given a data aggregation tree with n links, assume the output of the Layer-First approach is the schedule S_{layer} and the minimum-latency schedule of this tree is S_{opt}. In the worst case, $|S_{layer}| = \Omega(\sqrt{n})|S_{opt}|$.*

Before we start the proof, we need to define a special pair of links called *counter-block links.*

Definition 3. *Two links are called counter-block links if and only if they are axially symmetric (as l_{AB}, l_{DE} and l_{CB}, l_{FE} shown in Fig.6) and can transmit concurrently without any additional interference (i.e. the existence of any other transmission makes their SINRs lower than the threshold).*

Proof. As shown in Fig.6, we just need to choose appropriate $||AB||$ and $||AE||$ such that $\frac{P/||AB||^\alpha}{P/||AE||^\alpha} = \beta$ then l_{AB} and l_{DE} are counter-block links. It is easy to show that if $||CB||$ is a little shorter than $||AB||$, we can also set $||CE||$ to make sure that l_{CB} and l_{FE} are also counter-block links. Obviously, this method allows us to have infinitely many pairs of counter-block links with two common receivers. Using this technique, we construct the data aggregation tree which is shown in Fig.7, whose ith layer has $\lceil i/3 \rceil$ pairs of counter-block links (except for the top layer). Assume this tree has n links in total and m layers, then $3(m-1)m/2 = (n-2)/2$, i.e., the number of layers $m = \Theta(\sqrt{n})$.

Still, we need to apply the Layer-First approach on this data aggregation tree. For every layer, the maximum number of concurrently transmitting links is two due to the fact that there are only two receivers. One pair of counter-block links can be selected to transmit any timeslot by the Layer-First approach, so the whole aggregation needs $(n-2)/2 + 2$ timeslots. Next, we will show a more efficient scheduling which needs just $m = \Theta(\sqrt{n})$ timeslots.

Except for the top layer, we separate the tree into left part and right part. For the left (or the right) part, in the ith timeslot, one link in the $(3k+i-1)$th layer, $k = 1, 2, \cdots$, can transmit. In other words, for one single side, links are chosen at intervals of three layers in one timeslot (no vertical links can be chosen except for the deepest layer). So after two timeslots (one for each side), the 1st layer of the tree finishes transmission. At the same time, the 4th layer leaves one vertical link for each side, which means after eight timeslots, all the deepest four layer are removed from the tree after transmission, and so on. Finally, this scheduling can finish aggregation of this tree with $2m$ timeslots. We show $|S_{leaf}| = \Omega(\sqrt{n})|S_{opt}|$ by plugging in the definition of m. Next lemma explains that this schedule satisfies the SINR constraint.

Lemma 4. *For a set of links in one side of the tree constructed above, it is SINR-feasible if any two links in this set are at least three layers far away from each other.*

Proof. Assume this SINR-feasible set is L. For any link l_0 in L, there are at most two links (denoted as l_3 from higher layer and l_{-3} from lower layer) which are three layers far away from it. The distance between the sender of l_3 (or l_{-3}) and the receiver of l_0 is at least $2||l_0||$ (or $4||l_0||$). Similarly, there are two links l_6, l_{-6} which are six layers far away from l_0. Corresponding distances between senders and l_0's receiver are at least $5||l_0||$ and $7||l_0||$, and so on. Since the total number of links in one side of the tree is bounded by $n/2$, the SINR of l_0 can be easily derived:

$$SINR(l_0) \geq \frac{\frac{P}{||l_0||^\alpha}}{\frac{P}{2||l_0||)^\alpha} + \frac{P}{4||l_0||)^\alpha} + \cdots + \frac{P}{(2+3(\frac{n}{2}-1))||l_0||)^\alpha} + \frac{P}{(4+3(\frac{n}{2}-1))||l_0||)^\alpha}}$$

$$> \frac{\frac{P}{||l_0||^\alpha}}{2\sum_{k=1}^{n/2} \frac{P}{(k||l_0||)^\alpha}} > \frac{1}{2\sum_{k=1}^{\infty} \frac{1}{k^\alpha}} > \frac{1}{2\frac{\alpha}{\alpha-1}} > 1$$

where the second-to-last inequality follows the Riemann's zeta-function and the last one based on the fact that $\alpha \in [2, 6]$.

4 Approximation Algorithm for MLAT

In this section, we describe a greedy algorithm that solves MLAT, using existing techniques for the Wireless Capacity Maximization Problem (the same as the One-Shot Scheduling problem) [9,18] ([9] needs much larger P_{max}). Our algorithm is performed in a layer-by-layer style. Note that even though the greedy approaches have been proved to perform poorly without appropriate topology control, we show that we can still acccomplish an acceptable approximation ratio when data aggregation trees have small but reasonable depths.

The basic assumption for this section:

- The maximum power P_{max} for all senders is large enough that every link can transmit successfully in some case: $P_{max}/(Nd_{max}^\alpha) \geq \beta$, where d_{max} is the length of the longest edge of the aggregation tree.

– All the edges in the aggregation tree have lengths greater than 1. Any instance can be transformed into this case by scaling.

According to the analysis in [9,18], the correctness of the SINR-feasible sets generated in line 18 of Algorithm 1 is guaranteed. The approximation ratio for maximizing concurrently transmitting links is constant, which leads to an approximation algorithm for minimum latency scheduling with an approximation ratio bounded by $O(\log n)$. Note that the algorithm in [9] needs a very large maximum transmitting power.

Algorithm 1 labels all the edges of the given tree by first using a depth first search approach and then finds the depth of this tree in lines 3-7. Simply, the currently deepest edges of the tree are selected as scheduling candidates in lines 11-15. Then the existing scheduling algorithm for maximizing concurrent transmissions is used to select an approximated maximum SINR-feasbile link set. It repeats this process until all the links have been scheduled.

Algorithm 1. The Layer-by-Layer Algorithm for MLAT

```
Input: An arbitrary aggregation tree $T = \{V, E\}$ and $N$, $\alpha$, $\beta$, $P_{min}$, $P_{max}$;
Output: A schedule $S$ in which every edge can transmit successfully under SINR;
 1: $S := \emptyset$, $depth := 0$, $t := 1$;
 2: Use Depth First Search(DFS) to label every edge $e$ with its layer, to be stored in
    $layer(e)$
 3: for every edge $e$ in $T$ do
 4:    if $depth < layer(e)$ then
 5:       $depth := layer(e)$;
 6:    end if
 7: end for
 8: $L := E$;
 9: while $|L| > 0$ do
10:    $L' := \emptyset$;
11:    for every edge $l_i$ in $L$ do
12:       if $layer(l_i) = depth$ then
13:          $L' := L' \cup l_i$;
14:       end if
15:    end for
16:    $L := L \setminus L'$;
17:    while $|L'| > 0$ do
18:       Given $N$, $\alpha$, $\beta$, $P_{min}$, $P_{max}$, use the constant-approximation algorithm for
          maximizing concurrent transmissions (please refer to [9] or [18]), to compute
          an approximate maximum SINR-feasible link set $L''$ in $L'$;
19:       $S_t := L''$; $S := S \cup \{S_t\}$; $L' := L' \setminus S_t$; $t := t + 1$;
20:    end while
21:    $depth := depth - 1$;
22: end while
23: Return $S$;
```

In Algorithm 1, we divide the schedule generated by the algorithm into sub-schedules: $S = \{S_1, S_2, \ldots, S_d\}$. d is the depth of the input aggregation tree, and S_i is the sub-schedule in which only links in the ith layer can appear: $S_1 \cup S_2 \cup \cdots \cup S_d = S_{alg}$, and $S_i \cap S_j = \emptyset$, $\forall i, j \in [d], i \neq j$.

Define $T_{opt} = |S_{opt}|$, where S_{opt} is the optimal solution of MLAT. If we only schedule the links in the ith layer, the minimum number of timeslots T_i needed must not exceed T_{opt}. This is because if T_i is greater than T_{opt}, we can find a subschedule $S_{i,opt}$ in S_{opt} which schedules all links in the ith layer, i.e., $|S_{i,opt}| \leq T_{opt} < T_i$, which contradicts the fact that T_i is the minimum number of timeslots required to schedule these links. Thus, we have $|S_{1,opt}| + |S_{2,opt}| + \cdots + |S_{d,opt}| \leq d\dot{T}_{opt}$. We already know that every sub-schedule from our algorithm adheres to $|S_i| \leq \log n |S_{i,opt}|$, which implies that $T_{alg} \leq d \log n \cdot T_{opt}$.

On the other hand, it is obvious that T_{alg}, T_{opt}, n, d are all greater than 0 and $T_{opt} \geq d$, $T_{alg} \leq n$. Therefore, we derive another bound: $T_{alg} \leq (n/d)T_{opt}$.

From these, we bound the approximation ratio by $O(\min\{d \cdot \log n, n/d\})$. This gives an approximation ratio of $O(\log^2 n)$ when d is either very small ($d \leq O(\log n)$), or very large ($d \geq \Omega(n/\log^2 n)$). Fortunately, common aggregation trees often fall within these depth ranges. For example, applying our algorithm on a nearest neighbor tree (which has depth $O(\log n)$) leads to an $O(\log^2 n)$ approximation ratio.

5 Conclusion

In this paper, based on the fact that lots of factors (impenetrable obstacles, barriers, etc.) limit the topology construction for wireless networks in real scenarios, we introduce the Minimum Latency Aggregation Scheduling for Arbitrary Tree Topologies (MLAT) problem. We give the first NP-hardness proof for MLAT. In addition, we prove that the scheduling latencies generated by the two frequently used greedy algorithms could be $\sqrt{n}$ times the optimal result, where n is the total number of links. Given the fact that the MLAS problem could be solved in $O(\log n)$ timeslots [4] using the greedy scheduling approaches and topology control, the gaps we find for the MLAT problem show that giving another freedom of topology control could help reduce the aggregation latency. Finally, we propose an approximation algorithm for MLAT with approximation ratio bounded by $O(\min\{d \cdot \log n, n/d\})$ which is acceptable when the data aggregation trees have small depths. One of our future work is to give an approximation algorithm with a much better approximation ratio. Second, based on the preliminary observation that a clever mixture of different greedy approaches could lead to a better performance, we hope to devise a new heuristic algorithm combining both layer-first and leaf-first approaches. Another interesting extension to our work would be to design a distributed solution for MLAT.

Acknowledgment. The authors thank the anonymous reviewers for their helpful comments. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61103186, 61073174, 61033001, 61061130540, the Hi-Tech research and Development Program of China Grant 2006AA10Z216, and Hong Kong RGC-GRF grants 714009E and 714311.

References

1. Chen, X., Hu, X., Zhu, J.: Minimum Data Aggregation Time Problem in Wireless Sensor Networks. In: Jia, X., Wu, J., He, Y. (eds.) MSN 2005. LNCS, vol. 3794, pp. 133–142. Springer, Heidelberg (2005)
2. Fanghänel, A., Keßelheim, T., Vöcking, B.: Improved Algorithms for Latency Minimization in Wireless Networks. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 447–458. Springer, Heidelberg (2009)
3. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Transactions on Information Theory 46(2), 388–404 (2000)
4. Halldórsson, M.M., Mitra, P.: Wireless connectivity and capacity. In: SODA (2012)
5. Halldórsson, M.M., Wattenhofer, R.: Wireless Communication Is in APX. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 525–536. Springer, Heidelberg (2009)
6. Hua, Q.-S., Wang, Y., Yu, D., Tan, H.: Minimum latency link scheduling for arbitrary directed acyclic networks under precedence and sinr constraints. Journal of Interconnection Networks 12(1-2), 85–107 (2011)
7. Huang, S.C.H., Wan, P.J., Vu, C.T., Li, Y., Yao, F.: Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In: INFOCOM (2007)
8. Katz, B., Völker, M., Wagner, D.: Energy efficient scheduling with power control for wireless networks. In: WiOpt (2010)
9. Kesselheim, T.: A constant-factor approximation for wireless capacity maximization with power control in the sinr model. In: SODA (2011)
10. Lam, N.X., An, M.K., Huynh, D.T., Nguyen, T.N.: Minimum latency data aggregation in the physical interference model. In: MSWiM (2011)
11. Li, H., Hua, Q.-S., Wu, C., Lau, F.C.M.: Minimum-latency aggregation scheduling in wireless sensor networks under physical interference model. In: MSWiM (2010)
12. Li, X.Y., Xu, X.H., Wang, S.G., Tang, S.J., Dai, G.J., Zhao, J.Z., Qi, Y.: Efficient data aggregation in multi-hop wireless sensor networks under physical interference model. In: MASS (2009)
13. Moscibroda, T.: The worst-case capacity of wireless sensor networks. In: IPSN (2007)
14. Moscibroda, T., Wattenhofer, R.: The complexity of connectivity in wireless networks. In: INFOCOM (2006)
15. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology control meets sinr: the scheduling complexity of arbitrary topologies. In: MOBIHOC (2006)
16. Anne, Y., Goussevskaia, O.O., Wattenhofer, R.: Complexity in geometric sinr. In: MOBIHOC (2007)
17. Wan, P.-J., Huang, S.C.-H., Wang, L.X., Wan, Z.Y., Jia, X.H.: Minimum-latency aggregation scheduling in multihop wireless networks. In: MOBIHOC (2009)
18. Wan, P.-J., Ma, C., Tang, S., Xu, B.: Maximizing Capacity with Power Control Under Physical Interference Model in Simplex Mode. In: Cheng, Y., Do Eun, Y., Qin, Z., Song, M., Xing, K. (eds.) WASA 2011. LNCS, vol. 6843, pp. 84–95. Springer, Heidelberg (2011)
19. Yu, B., Li, J., Li, Y.: Distributed data aggregation scheduling in wireless sensor networks. In: INFOCOM (2009)

An Optimized In-Network Aggregation Scheme for Data Collection in Periodic Sensor Networks

Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej

FEMTO-ST Laboratory, DISC Departement University of Franche-Comté
Rue Engel-Gros, 90016 Belfort, France
firstname.lastname@univ-fcomte.fr

Abstract. In-network data aggregation is considered an effective technique for conserving energy communication in wireless sensor networks. It consists in eliminating the inherent redundancy in raw data collected from the sensor nodes. Prior works on data aggregation protocols have focused on the measurement data redundancy. In this paper, our goal in addition of reducing measures redundancy is to identify near duplicate nodes that generate similar data sets. We consider a tree based bi-level periodic data aggregation approach implemented on the source node and on the aggregator levels. We investigate the problem of finding all pairs of nodes generating similar data sets such that similarity between each pair of sets is above a threshold t. We propose a new frequency filtering approach and several optimizations using sets similarity functions to solve this problem. To evaluate the performance of the proposed filtering method, experiments on real sensor data have been conducted. The obtained results show that our approach offers significant data reduction by eliminating in network redundancy and out-performs existing filtering techniques.

1 Introduction

Data collection from sensor networks can be made on demand or by data streaming. The first category is done by bi-directional dialogs between the sensor nodes and the base station. A request for data is sent from the end user via the sink to the sensor nodes which, in return, send back the data to the user via multi hop communications. On the other side, in data streaming, data flows primarily from the sensor node to the sink. In this category we distinguish the periodic sampling and the event driven data models. In this paper we are interested in ”periodic sampling” data model in sensor networks, where the acquisition of sensor data from a number of remote sensor nodes are forwarded to the gateway on a periodic basis. This data model is appropriate for applications where certain conditions or processes need to be monitored constantly, such as the temperature in a conditioned space or pressure in a process pipeline. There are couple of important design considerations associated with the periodic sampling data model. The most critical design issue is the phase relation among multiple sensor nodes. If two neighbor nodes operate with identical or similar sampling rates, redundant packets from the two nodes are likely to happen repeatedly. It is essential for sensor networks to be able to detect and clean redundant transfered data from the nodes to the sink. In-network data aggregation has been proven as an effective technique for

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 153–166, 2012.

eliminating redundancy and forwarding only the extracted information from the raw data. Furthermore, by doing so data aggregation can often reduce the communication cost and extend the whole network lifetime.

In this paper we present a hierarchical multilevel data aggregation scheme aiming to optimize the volume of data transmitted thus saving energy consumption and reducing bandwidth on the network level. A first level in-sensor process is done by the nodes themselves. Instead of sending each sensor node's raw data to a base station, the data is cleaned periodically by the sensor node itself before sending it to an aggregator node for a second level of aggregation. At this level, we are interested in exploring a new part of the filtering aggregation problem, by focusing on identifying the similarity between data sets generated by neighboring nodes and sent to the same aggregator. Our objective is to identify similarities between near sensor nodes, and integrate their captured data into one record while preserving information integrity.

In this paper, we provide a new prefix filtering method to study the sets similarity in sensor networks. We propose frequency filtering optimization techniques, which exploits the ordering of measurements according to their frequencies. A frequency of a measure is defined by the number of occurrences of this measure in the set defined at the first aggregation level. Furthermore, we provide a new optimization method for early termination of sets similarity computing. To evaluate our approach we conducted extensive experimental study using real data measurements. The obtained results compared to the existing algorithms show the effectiveness of our method which significantly reduces the number of duplicate data.

The rest of the paper is organized as follows, Section 2 gives an overview on related works reported on data aggregation in sensor networks. Section 3 describes our periodic data aggregation scheme. The local aggregation level is presented in section 4. Review on similarity functions and our proposed frequency filtering techniques are presented in Section 5. Experimental results are given in Section 6. Section 7 concludes the paper with some directions to a future work.

2 Previous Data Aggregation Work

Data aggregation in wireless sensor networks has been well studied in recent years [1] [2] [3]. It means computing and transmitting partially aggregated data to the end user rather than transmitting raw data in networks to reduce the energy consumption [4]. There are vast amount of extant works on in-network data aggregation in the literature.

Some of the methods reported recently are query based methods [5] [6]. A query is generated at the sink and then broadcasted through the network. Some nodes just process the query, while others propagate it, receive partial results, aggregate results, and send them back to the sink. Various algorithmic techniques have been proposed to allow efficient aggregation without increasing the message size [7].

Some works, such as [8] [9] [10], use the clustering methods for aggregating data packets in each cluster separately. Among these methods, the LEACH protocol [11] [12]. In [9], the authors propose a self-organizing method for aggregating data based on the architecture CODA (Cluster-based self-Organizing Data Aggregation), based on the Kohonen Self-Organizing Map to aggregate sensor data in cluster. In a first step before deployment, the nodes are trained to have the ability to classify the sensor data. Thus, it increases the quality of data and reduces data traffic as well as energy-conserving. An adaptive data aggregation (ADA) scheme for clustered sensor networks has been proposed in [10]. In this scheme, a time based as well as spatial aggregation degrees are introduced. They are controlled by the reporting frequency at sensor nodes and by the aggregation ratio at cluster heads (CHs) respectively. The function of the ADA scheme is mainly performed at the sink node, with a little function at CHs and sensor nodes.

In a tree based network as our presented work, sensor nodes are organized into a tree where data aggregation is performed at aggregators along the tree to arrive to the sink. Tree based data aggregation approaches are suitable for in-network data aggregation. The authors in [13] [14], have proposed Tree on DAG (ToD) for data aggregation, a semistructured approach that uses Dynamic Forwarding on an implicitly constructed structure composed of multiple shortest path trees to support network scalability. The key principle behind ToD was that adjacent nodes in a graph will have low stretch in one of these trees in ToD, thus resulting in early aggregation of packets.

In our previous work [3], we have shown that existing prefix filtering methods are very complex and not suitable for sensor networks and we proposed a heuristic based on the frequency ordering. In this paper, we propose two optimization techniques based on frequency filtering extention which can be integrated with our previous prefix method [3] to find similar data sets efficiently. Furthermore we provide a new and faster technique for sets similarity computation.

3 Periodic Data Aggregation

Due to resource restricted sensor nodes, it is important to minimize the amount of data transmission among sensor networks so that the average network lifetime and the overall bandwidth utilization are improved. To reduce the amount of sending data, an aggregation approach can be applied along the path from sensors to the sink. Sensor nodes collect information from the region of interest and send it to aggregators. Each aggregator then condenses the data prior to sending it on.

Our data aggregation method works in two phases, the first one at the nodes level, which we call local aggregation and the second at the aggregators level. At each period p each node sends its aggregated data set to its proper aggregator which subsequently aggregates all data sets coming from different sensor nodes and sends them to the sink.

4 Local Aggregation

In periodic sensor networks, we consider that each sensor node i at each slot s takes a new measurement y_{is}. Then node i forms a new set of captured measurements M_i with period p, and sends it to the aggregator. It is likely that a sensor node takes the same (or very similar) measurements several times especially when s is too short. In this phase of aggregation, we are interested in identifying locally duplicate data measurements in order to reduce the size of the set M_i. Therefore, to identify the similarity between two measures, we provide the two following definitions:

Definition 1 **($link$ function).** *We define the $link$ function between two measurements as:*

$$link(y_{is_1}, y_{is_2}) = \begin{cases} 1 & \text{if} \quad \|y_{is_1} - y_{is_2}\| \leq \delta, \\ 0 & \text{otherwise.} \end{cases}$$

where δ is a threshold determined by the application. Furthermore, two measures are similar if and only if their $link$ function is equal to 1.

Definition 2 **(Measure's frequency).** *The frequency of a measurement y_{is} is defined as the number of the subsequent occurrence of the same or similar (according to the $link$ function) measurements in the same set. It is represented by $f(y_{is})$.*

Using the notations defined above the local aggregation algorithm is done as follows [3]. For each new sensed measurement (at each slot), a sensor node i searches for the similar measure already captured. If a similar measurement is found, it deletes the new one while incrementing the corresponding frequency by 1, else it adds the new measure to the set and initialize its frequency to 1. At the end of the period p, each node i will possess a local aggregated set M_i and send it to its aggregator.

5 Duplicate Data Sets Aggregation

At this level of aggregation, each aggregator has received k sets of measurements and their frequencies. The idea here is to identify all pairs of sets whose similarities are above a given threshold t. For this reason we use a similarity function which measures the degree of similarity between the two sets and returns a value in $[0, 1]$. A higher similarity value indicates that the sets are more similar. Thus we can treat pairs of sets with high similarity value as duplicates and reduce the size of the final data set that will be sent to the sink.

5.1 Similarity Functions

A variety of similarity functions have been used in the literature such as overlap threshold, Jaccard similarity and Cosine similarity [15–17]. We denote $|M_i|$ as the number of elements (measures) in the set M_i. The following functions can be used to measure the similarity between two sets of measurements M_i and M_j :

Overlap similarity: $O(M_i, M_j) = |M_i \cap M_j|$
Jaccard similarity: $J(M_i, M_j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|}$
Cosine similarity: $C(M_i, M_j) = \frac{|M_i \cap M_j|}{\sqrt{|M_i| \times |M_j|}}$
Dice similarity: $D(M_i, M_j) = \frac{2 \times |M_i \cap M_j|}{|M_i| + |M_j|}$

All these functions are commutative and can be transformed to the Overlap similarity easily. For instance, we can present the Jaccard similarity function as follows:

$$J(M_i, M_j) = \frac{O(M_i, M_j)}{|M_i| + |M_j| - O(M_i, M_j)}$$

In our approach, we will focus on the Jaccard similarity. It is one of the most widely accepted function because it can support many other similarity functions [16]. In our application, two given sets M_i and M_j are considered similar if and only if:

$$J(M_i, M_j) \geq t$$

where t is a threshold given by the application itself. This equation can be transformed as:

$$J(M_i, M_j) \geq t \Leftrightarrow O(M_i, M_j) \geq \alpha \tag{1}$$

where, $\alpha = \frac{t}{1+t}.(|M_i| + |M_j|)$.

In order to study the similarity functions for data aggregation in sensor networks, we define a new function for overlapping "$\cap_s$" between two sets of measurements as follows:

Definition 3 (Overlap function). *Consider two sets of measurements M_1 and M_2, then we define:*

$M_1 \cap_s M_2 = \{(y_1, y_2) \in M1 \times M2$ *such that* $link(y_1, y_2) = 1\}$*; and* $O_s(M_1, M_2) = |M_1 \cap_s M_2|$.

To evaluate the similarity between two sets we obtain:

$$J(M_i, M_j) \geq t \Leftrightarrow |M_i \cap_s M_j| \geq \alpha = \frac{t}{1+t}.(|M_i| + |M_j|) \tag{2}$$

5.2 Sets Similarity Computation

In this section we provide techniques for computing the similarity between the received sets. A naïve solution to find all similar sets is to enumerate and compare every pair of sets. This method is obviously prohibitively expensive for large data sets (such the case of sensor networks), as total number of comparison is $O(n^2)$.

To reduce the number of comparisons between sets a prefix filtering method has been proposed. Several approaches for traditional similarity join between sets are based on the prefix filtering principle [15] [17] [3]. This method is based on the intuition that if all sets of measures are sorted by a global ordering, some fragments of them must share several common tokens with each other in order to meet the threshold similarity. An inverted index maps a given measurement m to a list of identifiers of sets that contain m_i such that $link(m_i, m) = 1$. After inverted indices for all measures in the set are built, we can scan each one, probe the indices using every measure in the set M, and obtain a set of candidates; merging these candidates together gives us their actual overlap with the current set M; final results can be extracted by removing sets whose overlap with M is less than $\lceil \frac{t}{1+t}.(|M_i| + |M_j|) \rceil$(Equation 1).

This intuition is formalized by the following $Lemma$ inspired from [17]:

Lemma 1. *Consider two sets of sensor measures M_i and M_j, such that their elements are ordered by a global defined ordering. Let the p-$prefix$ be the first p elements of M_i. If $|M_i \cap_s M_j| \geq \alpha$, then the $(|M_i| - \alpha + 1)$-$prefix$ of M_i and the $(|M_j| - \alpha + 1)$-$prefix$ of M_j must share at least one element.*

Proof. Lemma 1 can be proven similarly to the lemma of page 6 in [17].

To ensure the prefix filtering based approach does not miss any similarity set result, as shown in Lemma 1 we need a prefix of length $|M_i| - \lceil t.|M_i| \rceil + 1$ for every set M_i [3]. The algorithm for finding similarity sets based on prefix filtering technique is given in Algorithm 1. It takes as input a collection of datasets coming from different sensor nodes already sorted according to a defined ordering. It scans sequentially each set M_i, selects the candidates that intersects with its prefix. Afterwards, M_i and all its candidates will be verified against the jaccard similarity threshold to finally return the set of correct similar measurements sets.

Algorithm 1. Prefix-filtering based algorithm.

```
Require: Set of measures' sets $M = \{M_1, M_2 ... M_n\}$, and a threshold $t$.
Ensure: All pairs of sets $(M_i, M_j)$, such that $J(M_i, M_j) \geq t$.
$S \leftarrow \emptyset$
$I_i \leftarrow \emptyset$ ($1 \leq i \leq$ total number of measures)
for each set $M_i \in M$ do
    $p \leftarrow |M_i| - \lceil t \times |M_i| \rceil + 1$
    $X \leftarrow$ empty map from set id to int
    for $k \leftarrow 1$ to $p$ do
        $w \leftarrow M_i[k]$
        if ($I_{w_s}$ exists such that $link(w, w_s) = 1$) then
            for each Measurement $(M_j[l]), f(M_j[l]) \in I_{w_s}$ do
                $X[M_j] \leftarrow X[M_j] + 1$
            end for
            $I_{w_s} \leftarrow I_{w_s} \cup \{M_i\}$
        else
            create $I_w$
            $I_w \leftarrow I_w \cup \{M_i\}$
        end if
    end for
    for each $M_j$ such that $X[M_j] > 0$ do
        if $O_s(M_i, M_j) \geq \alpha$ then
            $(S \leftarrow \{(M_i, M_j)\})$
        end if
    end for
end for
return $S$
```

Prefix filtering algorithm helps prune out unfeasible sets of measures, however, in practice the number of non-similar sets surviving after this technique is still quadratic growth [18]. Following the prefix filtering, many optimization methods [18] [19] were proposed to prune out further the unfeasible non-similar sets. A trade-off of these prefix filtering optimizations is that usually require more computational efforts which is unsuitable by heavy resources sensor networks. In our approach, we provide some optimizations for prefix filtering techniques based on measures frequency while taking into account this trade-off.

5.3 Frequency Filtering Approach

In this section, we present our frequency filtering method based on prefix extension. We begin by introducing some definitions and notations which will be the basis of what follows. In periodic sensor networks, two data sets are similar if their measurements overlap with each other, and especially the ones having *higher frequencies values*.

Definition 4 (Ordering $\mathcal{O}$). *We define an ordering $\mathcal{O}$ which arranges the measurements of a given set by the decreasing order of their frequencies.*

For two similar measures m_i and m_j such that $link(m_i, m_j) = 1$, we denote f_{min} $(m_i, m_j) = Min(f(m_i), f(m_j))$ the minimum value of the frequency of these measures.

Definition 5 ($f_s(M_i, M_j)$). *Consider two sets of measures M_i and M_j, we define* $f_s(M_i, M_j) = \sum_{k=1}^{O_s(M_i,M_j)} (f_{min}((m_i, m_j) \in M_i \cap_s M_j))$.

In this paper, we consider that all sensor nodes operate with the same sampling rate, and every node captures τ measures with each period p. Thus we can deduce that for every received set M_i from node i we have: $\sum_{k=1}^{|M_i|}(f(m_k \in M_i) = \tau$.

Using the Jaccard similarity function, two sets M_i and M_j are similar if and only if: $O_s(M_i, M_j) \geq \alpha$ where $\alpha = \frac{t}{1+t}.(|M_i|+|M_j|)$ (Equation (2)). Supposing that the sets were sent to the aggregators without applying the first aggregation phase and without computing measures frequencies, thus we can observe that:

$$|M_i| = |M_j| = \tau \text{ and } f_s(M_i, M_j) = O_s(M_i, M_j). \tag{3}$$

Hence, from Equation (2) and Equation (3) we can deduce that:

$$M_i \text{ and } M_j \text{ are similar iff: } f_s(M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t}. \tag{4}$$

Frequency Filter Principle. Lemma 1 states that the prefixes of two sets of measures must share at least one measure in order to satisfy the prefix filtering condition (PFC). Nevertheless, in sensor networks this condition is easily satisfied. In this section, we will present an extension of the prefix filtering technique making the PFC condition more difficult to be satisfied.

Lemma 2. *Assume that all the measures in the sets M_i and M_j are ordered according to the global ordering $\mathcal{O}$. Let the p-$prefix$ be the first p elements of M_i. If $f_s(M_i, M_j) \geq \frac{2\times t\times\tau}{1+t}$, then $f_s(p\text{-}M_i, p\text{-}M_j) \geq \sum_{k=1}^{|p\text{-}M_i|}(f(m_k \in p\text{-}M_i)) - \frac{1-t}{1+t} \times \tau$.*

Proof. We denote by p-M_i the prefix of the set M_i and r-M_i the set of reminder measures where $M_i = \{p\text{-}M_i + r\text{-}M_i\}$. We have:

$$\begin{aligned} f_s(M_i, M_j) &= f_s(p\text{-}M_i, M_j) + f_s(r\text{-}M_i, M_j) \\ &= f_s(p\text{-}M_i, p\text{-}M_j) + f_s(p\text{-}M_i, r\text{-}M_j) + \\ &\quad f_s(r\text{-}M_i, M_j) \\ &\cong f_s(p\text{-}M_i, p\text{-}M_j) + f_s(r\text{-}M_i, M_j) \\ &\leq f_s(p\text{-}M_i, p\text{-}M_j) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) \end{aligned}$$

In the second line we can omit the term $f_s(p\text{-}M_i, r\text{-}M_j)$ because we have assumed that it is negligible compared to the other terms in the equation. Indeed, if the two sets are similar then the measures having highest frequencies must be in the prefix set and not in the reminder, which means that the overlapping between the $p\text{-}M_i$ and $r\text{-}M_j$ is almost empty. From the above equations and equation (4)(similarity condition) we can deduce:

$$\frac{2 \times t \times \tau}{1+t} \leq f_s(p\text{-}M_i, p\text{-}M_j) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) \tag{5}$$

From the following equation:

$$\sum_{k=1}^{|p\text{-}M_i|} (f(m_k \in p\text{-}M_i)) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) = \tau \tag{6}$$

We obtain:

$$f_s(p\text{-}M_i, p\text{-}M_j) \geq \sum_{k=1}^{|p\text{-}M_i|} (f(m_k \in p\text{-}M_i)) - \frac{1-t}{1+t} \times \tau \tag{7}$$

The lemma is proved.

Algorithm 2 describes our method to find similar sets of measures based on the frequency filtering approach. It is a hybrid solution, where we integrate our frequency condition presented in Lemma 2 to the prefix filtering approach presented in Algorithm 1.

Jaccard Similarity Computation. Although filtering approaches reduce the number of comparisons between the received sets of measures, the number of candidate sets surviving after this phase is still non negligible. Furthermore, the computation of the jaccard similarity between two candidates sets can be very complex, especially when it comes to sensor networks where measures' sets can have ten hundreds or thousands elements. Therefore, to continue filtering out further candidate sets we propose a new frequency filtering constraint in the verification phase. In doing so, we can also reduce the overhead of the jaccard similarity computation.

Algorithm 2. Frequency-filtering based algorithm.

```
Require: Set of measures' sets $M = \{M_1, M_2 ... M_n\}, t, \tau$.
Ensure: All pairs of sets $(M_i, M_j)$, such that $J(M_i, M_j) \geq t$.
  Replace line 5 in Algorithm 1 with
    – $Fs \leftarrow$ empty map from set id to int
    – $sumFreq \leftarrow 0$
    – for $k \leftarrow 1$ to $p$ do
        $sumFreq \leftarrow sumFreq + f(m_k \in p\text{-}M_i)$
    – end for
  Replace line 10 in Algorithm 1 with
    – $Fs[M_j] \leftarrow Fs[M_j] + f_{min}(M_i[k], M_j[l])$
  Replace line 18 in Algorithm 1 with
    – for each $M_j$ such that $Fs[M_j] > sumFreq - \frac{1-t}{1+t} \times \tau$ do
```

Assume that we want to compute the similarity between two sets M_i and M_j. Then, these sets are similar if they satisfy the overlap condition $f_s(M_i, M_j) \geq \frac{2\times t\times\tau}{1+t}$. We also assume that a measure $m \in M_i$ divides M_i into two partitions: one partition containing all the measures having frequencies higher than $f(m)$ including m denoted by $h\text{-}M_i$ and the second $l\text{-}M_i$ containing all the measures having frequencies less than $f(m)$. Similarly, we assume that any measure in M_j divides it in two partitions $h\text{-}M_j$ and $l\text{-}M_j$. The idea of dividing the sets is to find a measure where at this position a similarity upper bound is estimated and checked against the similarity threshold. As soon as the check is failed we can stop the overlap computing early. This hypothesis is formalized by the following lemma:

Lemma 3. *Assume that $|M_i| < |M_j|$ and all measures in M_i are ordered according to the global ordering $\mathcal{O}$. M_i and M_j are similar $\Rightarrow$ for any $m \in M_i$ dividing M_i into $h\text{-}M_i$ and $l\text{-}M_i$ we have: $f_s(h\text{-}M_i, M_j) \geq \frac{2\times t\times\tau}{1+t} - \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i))$.*

Proof. M_i and M_j are similar

$$\Rightarrow f_s(M_i, M_j) \geq \frac{2\times t\times\tau}{1+t} \tag{8}$$

$$\Rightarrow f_s(h\text{-}M_i, M_j) + f_s(l\text{-}M_i, M_j) \geq \frac{2\times t\times\tau}{1+t} \tag{9}$$

$$\Rightarrow f_s(h\text{-}M_i, M_j) \geq \frac{2\times t\times\tau}{1+t} - f_s(l\text{-}M_i, M_j) \tag{10}$$

Then we have:

$$f_s(l\text{-}M_i, M_j) \leq min(\sum_{k=1}^{|l\text{-}M_i|}(f(m_k)), \sum_{k=1}^{|M_j|}(f(m_k))) \tag{11}$$

$$\leq min(\sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)), \tau) \tag{12}$$

$$\leq \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)) \tag{13}$$

From equations (10) and (13) we can deduce that:

$$f_s(h\text{-}M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} - \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)).$$

The lemma is proved.

The algorithm of overlap computation is given in Algorithm 3

Algorithm 3. Overlap Computation.

```
Require: Two sets of measures $M_i$ and $M_j$, $t$, $\tau$.
Ensure: $O_s(M_i, M_j)$.
$O_s \leftarrow 0$
Consider $|M_i| < |M_j|$
$sumFreqH \leftarrow 0$
$sumFreql \leftarrow \tau$
$M_j \leftarrow sort(M_j, |M_j|)$ $M_j$ is sorted in increasing order of the measures
for $k \leftarrow 0$ to $|M_i|$ do
    $sumFreql \leftarrow sumFreql - f(M_i[k])$
    Search similar of $M_i[k]$ in $M_j$
    find $M_j[l]/link(M_i[k], M_j[l]) = 1$
    $sumFreqH \leftarrow sumFreqH + f_{min}(M_i[k], M_j[l])$
    if $sumFreqH \geq \frac{2 \times t \times \tau}{1+t} - sumFreql$ then
        $O_s \leftarrow O_s + 1$
    else
        Return $-\infty$
    end if
end for
Return $O_s$
```

In this algorithm, we used two kinds of measures ordering depending on the sets sizes. The first one according to the global ordering $\mathcal{O}$ (M_i in the above algorithm) and the second is sorted in increasing order of the measures to accelerate a measure search[1].

[1] In our experiments we used the binary search.

6 Experimental Results

To evaluate our approach, we conducted multiple series of simulations using the discrete event simulator OMNET++ [20]. The objective of these simulations is to confirm that our prefix frequency filtering (PFF) technique can successfully achieve desirable results for data aggregation in periodic sensor networks. Therefore, In our simulations we used real readings collected from 45 sensor nodes deployed in the Intel Berkeley Research Lab [21]. Every 31 seconds, sensors with weather boards were collecting humidity, temperature, light and voltage values. For the sake of simplicity, in this paper we are interested in one field of sensor measurements: the temperature[2]. We performed several runs of the algorithms (an average of 15 runs). In each experimental run, we generated a network of 46 nodes corresponding to those was deployed in the Intel Berkeley Lab. Each node then reads periodically real measures saved in a file while applying the first aggregation algorithm. At the end of this step, each node sends its set of measures/frequencies to an aggregator node which in his turn applies prefix and filtering algorithms to theses sets. Furthermore, we compare our approach to the ToD protocol proposed in [13] [14]. As our real data sensor network consists of 46 nodes, we use ToD in a one dimensional Network as explained in [14] and we only divide the network into two F-cluster.

We evaluated the performance of the protocols using the following parameters: **a)** the number of sensor measurements taken by all nodes during a period τ, and **b)** the threshold of the Jaccard similarity function t. The threshold δ is fixed to 0.07. The aggregation function used for the ToD protocol is the same used in our approach (PFF) based on the link function (cf section 4). We employ four metrics in our simulations:

- The number of candidate sets generated after applying the prefix filtering approach [3], the frequency filtering algorithms with optimizations (PFF) and the final result (the real number of duplicate sets);
- Percentage of received measures: It represents how effective a protocol is in aggregating data. It is the number of measures received by the sink over the number of measures taken by all nodes.
- Data accuracy: represents the measures loss rate. It is a evaluate of measures taken by the source nodes and did not received at the base station (sink). It is defined also as the aggregation error.
- Overall energy dissipation: is the total energy dissipation of the entire network. To evaluate the energy consumption of our approach we used the same radio model as discussed in [21].

6.1 Prefix Frequency Filtering Optimizations

In this section we compared the number of candidates (number of comparisons) generated respectively by our frequency filtering technique (PFF), the prefix filtering algorithm and the results obtained after applying the Jaccard similarity function. We

[2] The others are done by the same manner.

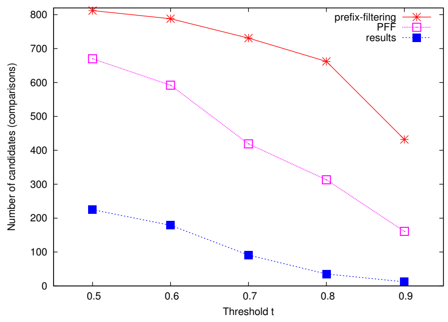


Fig. 1. Sets comparison

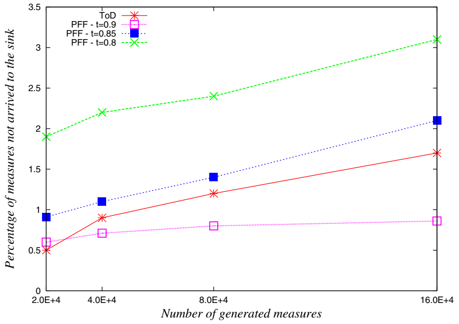


Fig. 3. Data accuracy

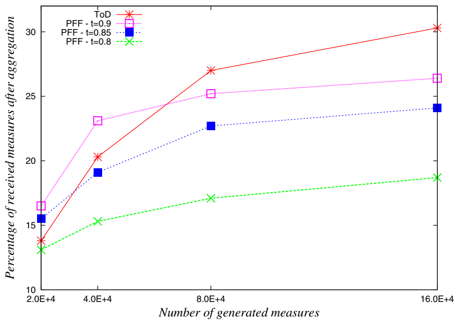


Fig. 2. Received measures

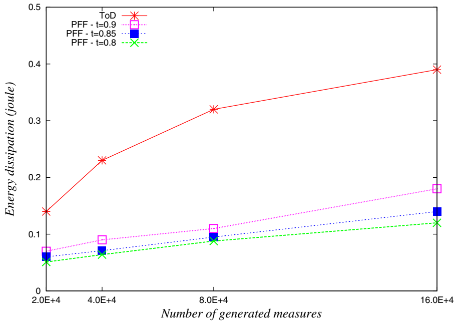


Fig. 4. Total energy dissipation

fixed the number of the total measuremtns taken by all the nodes during a period to $\tau = 8.E + 04$. The obtained result is shown in figure 1. We notice that, when the similarity threshold increases from 0.7 to 0.9, the number of comparisons of the frequency filtering and the prefix filtering becomes closer. We can also see that our frequency filtering technique (PFF) outperforms the prefix filtering methods in all cases. Moreover, the number of candidates generated by all the algorithms is far bigger than the results number. This is to prove that under this circumstance, applying early termination algorithm is very effective (Algorithm 3).

6.2 Percentage of Received Measures and Data Accuracy

Figure 2 shows the percentage of received measures over the total number taken by all nodes for the temperature field. These experiments permit to show how well aggregation protocols do aggregation and reduce redundant measures. PFF performs better than ToD in terms of data aggregation because of it is ability to compare sets of data instead of single packets. In other words, PFF reduces the number of redundant data traveling into the networks better than TOD especially when the number of readings increase (the case of periodic networks). We also notice that, the percentage of received packets remains almost unchangeable while increasing the sensor readings.

Figure 3 depicts the resulsts of the aggregation error. This metric is an important performance index, and the high measures loss rate will impact the use of the data greatly. The obtained results show that the two protocols have good performance regarding the aggregation error. As expected, when we increase the threshold t of the similarity function we reduce the measures loss rate. For instance, we can notice that PFF outperforms ToD in terms of data accuracy for $t = 0.9$.

6.3 Overall Energy Dissipation

The overall energy dissipation is the total energy consumption of the entire network. Figure 4 shows the results for total energy consumption obtained while varying the total number of sensor readings. The figure shows that the overall energy dissipation for different protocols increases as the number of readings increases. We notice that ToD consumes not too much, but does not scale well as the number of readings increases. For all the values of the threshold t tested, PFF always outperforms the ToD protocol in total energy dissipation. This is because, the packet-packet comparison used in ToD instead of data sets in PFF generates more transmissions in the network, furthermore, the packet construction in ToD contains additional information required for the aggregation which is not the case in PFF.

7 Conclusion and Future Work

In this paper we proposed a tree based bi-level model for data aggregation in periodic sensor networks: Local aggregation and Frequency filtering aggregation. In the first one we provided an aggregator for simple captured measurements based on a link similarity function while in the second level our objective is to detect and aggregate multiple data sets generated by different neighboring nodes. We proposed a new frequency filtering approach and several optimizations using sets similarity functions to find similar data sets. It was shown through simulations on real data measurements that our method reduces drastically the redundant sensor measures and outperforms the existing prefix filtering approaches.

We have two major directions for our future work. The first direction seeks to adapt our proposed method to take into account reactive periodic sensor networks, where sensor nodes operate with different sampling rate. In periodic applications the dynamics of the monitored condition or process can slow down or speed up; and to save more energy the sensor node can adapt its sampling rates to the changing dynamics of the condition or process. The second direction is to develop a new suffix frequency filter algorithm beside the frequency filtering approach proposed in this paper. Our goal is to use additional filtering method that prunes erroneous candidates that survive after applying the prefix and frequency filtering technique.

References

1. Yu, B., Li, J., Li, Y.: Distributed data aggregation scheduling in wireless sensor networks. In: IEEE INFOCOM 2009 (2009)

2. Zheng, Y., Chen, K., Qiu, W.: Building representative-based data aggregation tree in wireless sensor networks. Mathematical Problems in Engineering, 11 pages (2010)
3. Bahi, J., Makhoul, A., Medlej, M.: Data aggregation for periodic sensor networks using sets similarity functions. In: IWCMC 2011, 7th IEEE Int. Wireless Communications and Mobile Computing Conference, pp. 559–564 (July 2011)
4. Sharaf, M.A., Beaver, J., Labrinidis, A., Chrysanthis, P.K.: Tina: A scheme for temporal coherency-aware in-network aggregation. In: 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, pp. 69–76 (2003)
5. Xu, Y., Lee, W.-C., Xu, J., Mitchell, G.: Processing window queries in wireless sensor networks. In: 22nd Int. Conf. on Data Engineering, ICDE, p. 70 (2006)
6. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: A tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper. Syst. Rev. 36(SI), 131–146 (2002)
7. Cormode, G., Garofalakis, M., Muthukrishnan, S., Rastogi, R.: Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In: 2005 ACM SIGMOD International Conference on Management of Data, pp. 25–36 (2005)
8. Cormode, G., Garofalakis, M., Muthukrishnan, S., Rastogi, R.: Prolonging the lifetime of wireless sensor networks via unequal clustering. In: Proceedings of the 5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (2005)
9. Lee, S., Chung, T.: Data Aggregation for Wireless Sensor Networks Using Self-organizing Map. In: Kim, T.G. (ed.) AIS 2004. LNCS (LNAI), vol. 3397, pp. 508–517. Springer, Heidelberg (2005)
10. Chen, H., Mineno, H., Mizuno, T.: Adaptive data aggregation scheme in clustered wireless sensor networks. Computer Communications 31(15), 3579–3585 (2009)
11. Shah, R.C., Rabaey, J.M.: Energy aware routing for low energy ad hoc sensor networks. In: IEEE Wireless Communications and Networking Conf. WCNC, pp. 350–355 (2002)
12. Younis, O., Fahmy, S.: An experimental study of routing and data aggregation in sensor networks. In: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 8 pages (2005)
13. Prakash, G.L., Thejaswini, M., Manjula, S.H., Venugopal, K.R., Patnaik, L.M.: Tree-on-dag for data aggregation in sensor networks. World Academy of Science, Engineering and Technology 37 (2009)
14. Fan, K.-W., Liu, S., Sinha, P.: Dynamic forwarding over tree-on-dag for scalable data aggregation in sensor networks. IEEE Trans. on Mobile Computing 7(10), 1271–1284 (2008)
15. Bayardo, R.J., Ma, Y., Srikant, R.: Scaling up all pairs similarity search. In: 16th International Conference on World Wide Web, WWW 2007, pp. 131–140 (2007)
16. Sarawag, S., Kirpal, A.: Efficient exact set-similarity joins. In: 32nd international Conference on Very large Data Bases, VLDB 2006, pp. 918–929 (2006)
17. Chaudhuri, S., Ganti, V., Kaushik, R.: A primitive operator for similarity joins in data cleaning. In: 22nd International Conference on Data Engineering (ICDE 2006), p. 5 (2006)
18. Xiao, C., Wang, W., Lin, X., Yu, J.X.: Efficient similarity joins for near duplicate detection. In: Proceeding of the 17th International Conference on World Wide Web, pp. 131–140. ACM (2008)
19. Xiao, C., Wang, W., Lin, X., Shang, H.: Top-k set similarity joins. In: Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 916–927 (2009)
20. OMNeT++, http://www.omnetpp.org/
21. Madden, S.: http://db.csail.mit.edu/labdata/labdata.html

Impulsive Interference Avoidance in Dense Wireless Sensor Networks

Nicholas M. Boers[1], Ioanis Nikolaidis[2], and Pawel Gburzynski[3]

[1] Department of Computer Science, Grant MacEwan University,
10700 104 Ave. NW, Edmonton, Alberta T5J 4S2, Canada
boersn@macewan.ca

[2] Department of Computing Science, University of Alberta, 2-21 Athabasca Hall,
Edmonton, Alberta T6G 2E8, Canada
nikolaidis@ualberta.ca

[3] Olsonet Communications Corporation, 51 Wycliffe Street, Ottawa,
Ontario K2G 5L9, Canada
pawel@olsonet.com

Abstract. Wireless sensor networks (WSNs) are subject to interference from other users of the radio-frequency (RF) medium. If the WSN nodes can recognize the interference pattern, they can benefit from steering their transmissions around it. This possibility has stirred some interest among researchers involved in cognitive radios, where special hardware has been postulated to circumvent non-random interference. Our goal is to explore ways of enhancing medium access control (MAC) schemes operating within the framework of traditional off-the-shelf RF modules applicable in low-cost WSN motes, such that they can detect interference patterns in the neighbourhood and creatively respond to them, mitigating their negative impact on the packet reception rate. In this paper, and based on previous work on the post-deployment characterization of a channel aimed at identifying “spiky” interference patterns, we describe (a) a way to incorporate interference models into an existing WSN emulator and (b) the subsequent evaluation of a proof-of-concept MAC technique for circumventing the interference. We found that an interference-aware MAC can improve the packet delivery rates in these environments at the cost of increased, but acceptable, latency.

Keywords: classification, interference, sampling, wireless sensor networks, channel modelling, medium access control.

1 Introduction

WSN nodes must be particularly resilient to interference because the ISM bands are heavily used, particularly in dense urban environments [1]. ISM sources are quite varied, including cordless telephones/headphones, wireless local area networks (WLANs), and microwave ovens. Most existing studies are based either on over-simplistic environmental models assuming Gaussian background noise, or on the assumption that interference arises from peer devices (members of the

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 167–180, 2012.

same networked wireless system). The two types of disturbance have received considerable attention in research under the umbrellas of channel modelling and MAC protocol design, respectively. The third type of disturbance, namely external interference from a different wireless system (possibly even from a system whose purpose is not data communication per se) has been much overlooked. Based on our empirical findings, external interference is already a major source of communication problems in WSN systems, especially those deployed in densely populated urban areas.

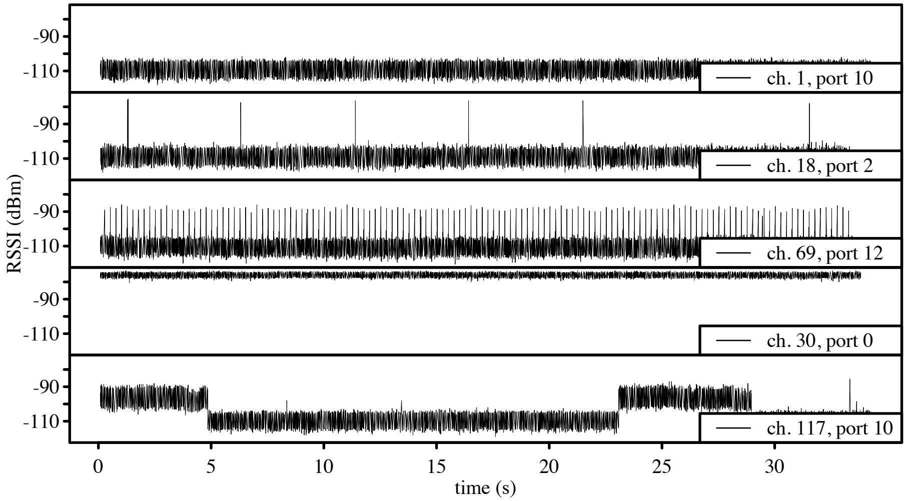


Fig. 1. The different primary interference classes identified in our RSSI traces. The middle pattern, representing frequent impulses of short duration, is the focus of this work.

Specifically, external interference became blatantly obvious to us in our 2008 deployment of the Smart Condo – a network to passively monitor an independent living environment [2,3]. As soon as its simple transceivers (RF Monolithics TR8100) began their operation (at 916.5 MHz), we noticed significant packet losses even over short distances and with the obvious lack of interference from peers. Those losses disappeared when the same set of motes was moved to another environment (several blocks away) for an in-lab study of their poor performance. Having thus confirmed that the environment itself was the culprit, we returned to it with another WSN comprised of 16, more flexible, motes to assess the character of the external interference. We specifically wanted that assessment to be carried out by a WSN, as opposed to some specialized and sophisticated spectrum analyzing equipment, because we wanted to determine how WSNs could analyze and respond to interference problems on their own.

The nodes of the new WSN were equipped with the Texas Instruments CC1100, capable of collecting digitized samples of the received signal strength indicator

(RSSI) at high rates over varying channels. Using that network, we took an extensive collection of RSSI traces sampled at 5000 Hz on 256 channels ranging from 904 to 954 MHz [4]. After plotting those traces as time series data, we immediately identified a number of recurrent interference patterns, including the one that caused our original alarming packet losses (Figure 1, middle). Reflecting back on that negative experience, although the TR8100 transmitted at reasonably powerful levels (10 dBm), it used a very simple encoding scheme (on-off keying) that is particularly susceptible to interference [5,6]. The analysis and the characterization and classification of interference patterns using WSN–suitable low-complexity techniques is described in two earlier publications [4,7]. Here, we present just a summary of the main results and describe how we were able to (a) integrate interference sources in a high–quality simulation testbed and (b) evaluate a simple MAC protocol that takes advantage of particular interference patterns.

Specifically, we explore the avoidance of impulsive (spiky) interference in dense wireless sensor networks (Figure 1, middle). We first review work related to the general characterization of channels (Section 2), and we then summarize our previously developed techniques capable of identifying this particular pattern. In Section 3, we describe the extension of an existing simulator with this characterization. After modelling the interference, we incorporate the classifier and a proof-of-concept MAC into a WSN application (Section 4) and present the results from simulating it (Section 5). Finally, in Section 6, we present some concluding remarks.

2 Related Work

When exploring interference, some researchers have focused on the interaction of specific protocols, e.g., IEEE 802.11b (WLAN), 802.15.1 (Bluetooth), and 802.15.4 (ZigBee) [8]. Similarly, others have concentrated their efforts on specific expected interferers, e.g., Chandra [9] used a spectrum analyzer in a 3-story building to explore the noise generated by electronic equipment in a workshop, a photocopier, elevator, and fluorescent tubes. In this section, we describe the small body of work that addresses interference more generally.

Using sensor platforms, Srinivasan, Dutta, Tavakoli, and Levis [10] studied packet delivery performance. With nodes synchronized, they encountered strong, spatially-correlated impulses (up to -35 dBm or higher) in their traces. Given the high correlation, they concluded that the spikes originated externally to the nodes.

Researchers working on closest-fit pattern matching (CPM) sampled noise in (a) WLAN-enabled buildings, (b) WLAN-enabled outdoor areas, (c) outdoor quiet areas, and (d) controlled areas [11,12]. They sampled channels both overlapping and non-overlapping an IEEE 802.11b network and observed three characteristics: (a) spikes sometimes as strong as 40 dB above the noise floor, (b) many of the spikes were periodic, and (c) over time, the noise patterns changed. In their work, they offered little description of the patterns beyond what we summarize here. Instead of focusing on specific patterns, they developed a modelling approach that initially replays the recorded trace and then estimates future points based on computed probabilities.

More recently, Srinivasan, Dutta, Tavakoli, and Levis [8] expanded on much of their previous work. With six synchronized nodes, they sampled RSSI values at 128 Hz and explored the correlation in the noise traces. They observed 802.11b interference as high at 45 dB above the noise floor, and in their figures, this interference appears as periodic impulses at roughly 36 Hz.

In our recent work, we explored measurements from a grid of sixteen nodes in an indoor urban environment [4]. Within the 80 m^2 space, we deployed the grid with 1.83 m spacing and elevated each node 28 cm off of the floor. We connected all of the nodes to a single computer using USB and then proceeded to simultaneously measure each node's RSSI value sampled at 5000 Hz. To the best of our knowledge, this sampling rate has been unmatched so far in a WSN framework. Over a period of roughly 2.5 hours, we scanned the 256 available channels ranging from 904 to 954 MHz.

Upon inspecting our high resolution traces, we identified the five recurrent patterns that we show in Figure 1. Specifically, the patterns are: (1) quiet, (2) sparse (random) impulses, (3) frequent (strongly periodic) impulses, (4) high level interference, and (5) shifting-mean interference. It would be highly presumptuous to claim that any interference patterns that we observed in a particular environment and on a particular day should be immediately generalized into blanket rules applicable to all wireless systems. However, the very fact that we clearly saw a small number of simple and easily discernible patterns and that some of those patterns have been uncovered before strengthens our confidence that the set of patterns we observed can be considered representative.

In this paper, we are interested in exploiting the pattern of periodic impulse "spikes." From the original 4096 traces (16 nodes × 256 channels), we randomly sampled 1024 traces and carefully hand-classified them for the presence of frequent periodic impulses. We encountered the pattern in 154 of the traces, and some of these traces contained other patterns as well. Since each full trace consists of 175 000 points, and because we are interested in a small set of samples (to conserve the amount of energy spent to sampling the medium), we *subsampled* the traces using even and Poisson subsampling techniques. For each trace of subsamples, we record whether the periodogram indicates the presence of frequent periodic impulses. Furthermore, in the periodogram calculation, we simplified the computation of the (co)sine by approximating it with values from a lookup table which contained quantized approximations of the (co)sine function. As reported in [7], we found that the automated classification yielded the same results as the hand classification (i.e., our ground truth) in the vast majority of cases, and hence the classifier was deemed adequate for our purposes. We also found that the performance of the classifier did not improve significantly past the point where 4000 samples were used.

3 Simulation

One need arising in the study of networks under interference patterns observed in traces collected from real networks is that, in order to produce repeatable simulation experiments, it is essential to model the interference sources (diverse as they might be) in a manner that is both general and easily implementable on a simulator. An ideal approach, advocated in this paper, would be to support a form of "scripted" interference source behaviours. Since our platform of choice for application development and for emulation and simulation is PicOS [13], we crafted simulated interference patterns in the idiom of PicOS threads. PicOS is conceptually derived from the SMURPH/SIDE simulator. Rather recently, PicOS gained wireless channel support [14] and the ability to emulate PicOS applications at the level of their API (application programming interface) using a component named VUE^2 [15] which leverages SIDE to provide an accurate simulation environment on which pre-deployment evaluation of protocols and systems can be conducted. Given this close relationship between our chosen OS and a mature simulator, our decision to use SIDE was quite natural.

We extended SIDE by adding the ability to define scripted external impulsive interference. The extension consists of (a) a user-specified configuration, (b) a new "node" type within the simulator, and (c) threads running on those nodes to produce the specified interference. Additional tags and attributes added to an existing XML (extensible markup language) configuration file provide the user-specified interference configuration. A new `interferers` attribute to the network tag indicates the number of interferers in the environment, e.g.,

```
<network nodes="40" interferers="3">
```

The user can use the new `<interferers>` tag to identify an interferer-specific block within the configuration akin to the existing `<nodes>` tag. Within this new section, the user can define the parameters for each interferer, e.g.,

```
<interferer number="0" type="impulsive">
  <location type="random">170.0 170.0</location>
  <pattern>
    R 0.245 s          ; random delay
    P                  ; start periodic portion
    O 0.0 dBm 3 dB     ; on at 0.0 dBm with 3 dB sd
    T 0.005 s          ; delay
    F                  ; off
    T 0.245 s          ; delay then implicit jump to P
  </pattern>
</interferer>
```

The attribute and value `type="random"` for the location causes SIDE to generate a new location every time the simulator starts (assuming a new random number generator seed). It uses the specified coordinates to bound the random values. Internally, each interferer becomes an object within the simulation, not unlike what already occurs for nodes. For these new objects, the user can create a library of processes, each capable of producing a certain class of interference.

For this work, we implemented an `Impulsive` process to simulate user-specified impulsive interference.

The body of the `<pattern>` tag essentially provides the *interference behaviour script* for the process `Impulsive` to follow. For an impulsive interferer, SIDE supports following commands:

`R`: delay for a random duration between 0 and the double argument (in seconds),
`T`: delay for the specified duration (in seconds),
`O`: generate interference at the specified power level (in dBm) with the specified standard deviation (in dB),
`F`: stop the generation of interference, and
`P`: mark the start of the periodic portion of the pattern.

Essentially, the `Interferer` process interprets (in a fetch-decode-execute style) the command sequence provided in the specification block. Once the end of the list of commands is reached, an implicit jump occurs to the command immediately following the `P` command.

We placed a number of synchronized impulsive interferers in a virtual environment, and using our earlier sampling application [4], collected a number of virtual traces (e.g., Figure 2). With very little tweaking, we were able to make the simulated traces match the substance of the real traces. Upon close inspection, there are slight differences, e.g., the simulated traces lack some random non-periodic components, and with a little more work, we could include these in our model as well. That said, the existing detail suffices for the classification and medium access control techniques that we implement next.

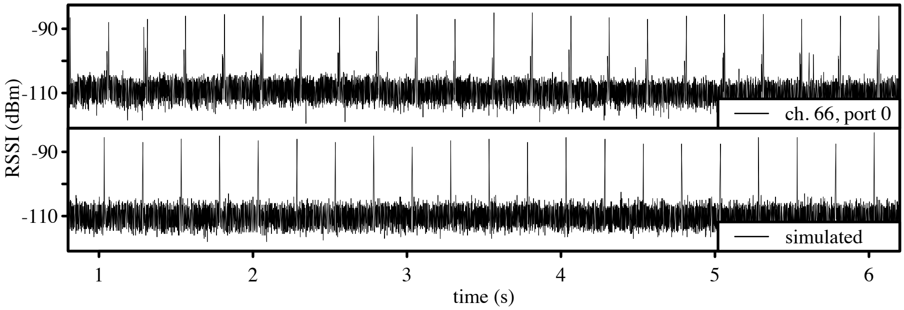


Fig. 2. An actual trace (top) plotted with a simulated trace (bottom). We used the same application to collect both traces.

4 Exploiting Interference Classification in a MAC Protocol

4.1 On-Line Classifier

To classify channels with regularly-spaced short-duration impulsive interference, we implement the approximate least-squares spectral analysis (LSSA) technique described in [7]. In the transceiver's transmit state machine, we introduce three new states to accommodate the classifier:

`CLS_INIT` Initializes the variables required for classification and immediately advances to `CLS_MEANEST`.

`CLS_MEANEST` A visit to this state represents the measurement of a single RSSI sample to compute the mean RSSI estimate. It remains in this state for 200 iterations prior to transitioning to `CLS_SAMPLE` – a number of iterations that proved reasonable in our early tests. The delay between each iteration is uniformly randomly distributed between 0 and 7 ms.

`CLS_SAMPLE` A visit to this state represents obtaining a single RSSI sample for calculating the LSSA. It remains in this state for 5000 iterations which span approximately 17.5 s and then transitions to the (regular) MAC state. The delay between each iteration is uniformly randomly distributed between 0 and 7 ms. We used more iterations than the minimal 4000 identified earlier simply as a precaution.

The complete classification process lasts just over 18 s during which we prevent nodes from communicating. If the application wishes to transmit packets during the process, they are simply queued until the classifier completes. It is implied that in a real deployment, the classification task is to be executed occasionally to assess the new levels and periods of any periodic impulse interference.

4.2 Pattern-Aware Medium Access Control (PA-MAC)

The output from the classifier indicates the presence of periodic short-duration impulsive interference at any of the tested frequencies. Our proof-of-concept pattern-aware MAC (PA-MAC) then uses this output in its attempt to steer transmissions around the impulses. In fact, the protocol makes a virtue out of interference, because the periodic interference becomes well-defined time points around which to anchor transmissions (with some back-off of course as we will later see). Stretching definitions a bit, the interference becomes a means for implicit synchronization of the MAC transmissions across nodes.

In our approach, we make observations about the interference at a transmitting node and assume that they also hold for the receiving node, i.e., we assume a significant amount of correlation in the interference between nodes. Particularly in small dense deployments, we have found this assumption to hold, e.g., we observed significant correlation in the traces collected in the Smart Condo. Moreover, other researchers have observed significant correlations in packet losses [16]. Even in larger environments, this assumption may hold given either a particularly strong interferer or a collection of correlated interferers.

To implement PA-MAC, we introduce one further state to the transceiver's state machine: `MAC_SEARCH` with the intention to use it as a way to track the impulse instants. Initially after the classification, and then again regularly after each transmission window, the process will enter this state to sample the channel to track the next impulse. Successfully finding an impulse causes the transceiver thread to (a) set a timer to mark the end of the next transmission window, i.e., the expected arrival of the next impulse and (b) delay for the expected duration of the currently identified impulse and then transition to the thread's primary state (XM_LOOP). Once in the main loop, the driver will retrieve outgoing packets as they become available and transmit them until the expiration of the first timer. At that point, the process reenters MAC_SEARCH where it attempts to track the next impulse.

For the sake of comparison, a simple baseline MAC protocol is *listen before transmitting* (LBT), which, other than sensing prior to attempting transmission, resolves contention using a random back-off. Given multiple transmitting nodes, the random back-off leaves little opportunity for nodes to synchronize. With PA-MAC, however, our regular tracking of the interference introduces a new opportunity for nodes to synchronize, which could ultimately cause a number of nodes to transmit at the same time. We eliminated this point of contention by introducing an additional random back-off.

5 Results

We evaluated the pattern-aware MAC within the interference-generating simulator, using the built-in shadowing channel model, and we tweaked the simulator's parameters to represent our physical hardware. We include results for both single- and multi-hop random topologies. For a given configuration, we average the measurements from 100 different topologies, each with its own traffic pattern, and plot the results with 95% confidence intervals.

5.1 Single-Hop

The single-hop configurations consist of 19 source nodes, one destination node, and two interferers, and the simulator places them all randomly within an 18 m × 18 m field. Since the model neither includes obstructions nor considers radio irregularity [17], these dimensions guarantee that the destination is within the transmission range of every source node. Each node transmits at a rate of 10 kbps and a transmission power of -20 dBm. The interferers introduce 5 ms pulses of impulsive interference at a period of 4 Hz and -30 dBm.

We first evaluated the effect of varying the packet length (Figure 3) on the packet reception rates (PRRs) and latency. Note that the destination node does not acknowledge received packets and nodes make no attempt to retransmit lost packets. We measure latency from the application perspective: the time that elapses between receiving the packet from the application (at the transmitter) and the application receiving the packet (at the receiver). Note that the effect of varying the packet length should be seen relative to the frequency of impulsive interference. Alternatively, we could have kept the packet length the same and changed the interference's period. We chose the former approach. In these tests, nodes generated new packets according to an exponential distribution with mean 50 s to reduce (if not practically eliminate) the effect of congestion. For each run

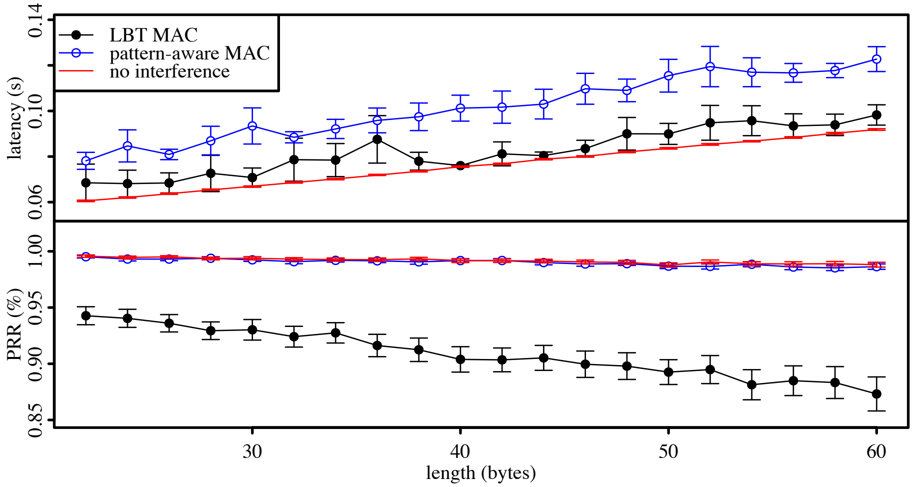


Fig. 3. In a dense single-hop network, the effect of the packet length on the packet reception ratio and the latency

of the simulation, we generate 500 s of input and allow the simulator to run for 600 s (in case of delayed packets).

When increasing the length, the PRR decreases for all configurations and the latency increases (as expected). In terms of PRRs, PA-MAC performs similarly to the quiet configuration because it successfully steers the transmissions around the interference. To obtain these PRRs, it ends up delaying transmissions that may collide with the interference, and the latency graph reflects this behaviour. The traditional LBT MAC's PRRs suffer at a greater rate than the other two configurations as more packets are lost to collisions than simply the non-zero bit error rate. The traditional LBT MAC shows higher latency than what is achieved by the quiet channel, demonstrating that the LBT MAC yields also occasionally to interference because it senses the medium as being busy.

We also evaluated the effect of varying the packet generation rate (Figure 4) on the PRRs and latency. In these experiments, we set the packet length to its maximum (60 bytes) in order to accentuate the variable's effect.

Under high congestion (mean packet inter-arrival time at each node $\mu <$ 2 s), the packet reception rates drop significantly for all methods in this dense network, and the LBT MAC and PA-MAC perform very similarly. Since all nodes are within range of each other, all transmissions will generate interference, but that interference may not be sufficiently strong for a node to recognize the medium as busy. The PRRs are lower for both of the interference configurations because the MACs will sometimes yield to the interference, leaving less of a window for data transmission. At lower levels of congestion, the PA-MAC tends towards the performance of the quiet configuration.

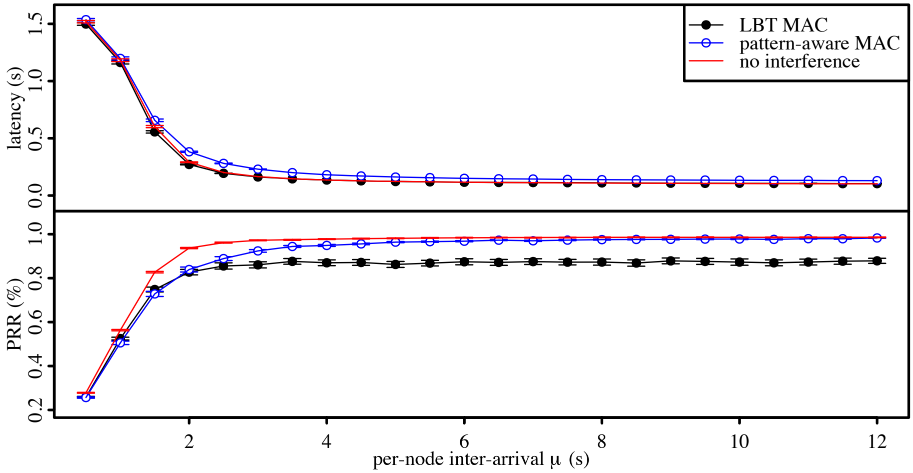


Fig. 4. In a dense single-hop network, the effect of the mean latency between per-transmitter packet introductions on the packet reception ratio and the latency

5.2 Multi-Hop

The multi-hop configurations consist of 39 source nodes, one destination node, and three interferers, and the simulator places them all randomly within a 170 m × 170 m field. As with the single-hop scenario, nodes transmit at a rate of 10 kbps and with transmission power -20 dBm. In this case, the three interferers produce a similar interference pattern to the single-hop case, but transmit at 0 dBm rather than -30 dBm. Given the larger field, we made this change to ensure the visibility of interferers across the network.

The transmitting nodes use the tiny ad hoc routing protocol, TARP [18], to deliver packets to the destination. TARP is a light-weight on-demand routing protocol that quickly converges to the shortest path in static networks. Because it lacks explicit control packets (minimal control information is present in the packet header) it does not inflate the overall traffic needed to support it. Although the application only demands one-way communication, the destination sends short 14-byte replies to each source node for the benefit of the routing protocol. Note that communication continues to be unacknowledged, and nodes make no attempt to retransmit lost packets.

Given random node locations, we need to take precautions to ensure that each source node has a path to the destination node. Immediately after generating a random layout, the simulator will search for a path from every source to the single destination while ensuring that each hop is less than the maximum transmission range. If the procedure finds a disconnected node, the simulator will generate a completely new node placement until a path exists between every pair of nodes, i.e., until the communication graph is connected.

Like in the single-hop case, we first evaluate the effect on varying the packet length on the PRRs and latency (Figure 5). To reduce congestion in the

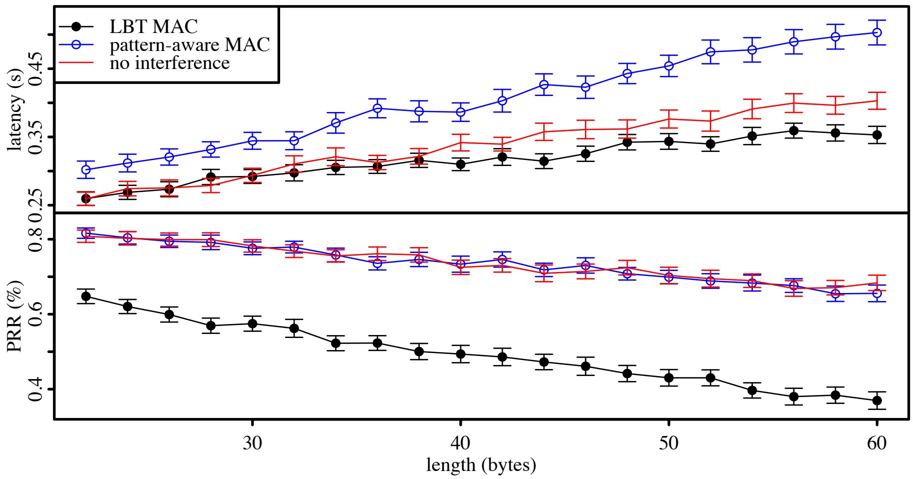


Fig. 5. In a connected multi-hop network, the effect of the packet length on the packet reception ratio and the latency

multi-hop environment given the high initial number of retransmissions, we lower the packet generation rate to follow an exponential distribution with mean of 200 s. Given the lower packet generation rate, we generate 2000 s of input and allow the simulator to run for 2100 s.

As with the single-hop case, we notice decreasing PRRs and increasing latencies as the length increases, and PA-MAC again follows the PRR of the quiet configuration. However, unlike the single-hop case, we notice that the quiet configuration no longer provides the baseline for delay. To explore this phenomenon, we investigate the hop lengths compared to packet lengths (Figure 6).

Since the network is static, we would expect little change in the expected number of hops as the packet length increases. However, we notice that the expected number of hops decreases for the LBT MAC as the packet length

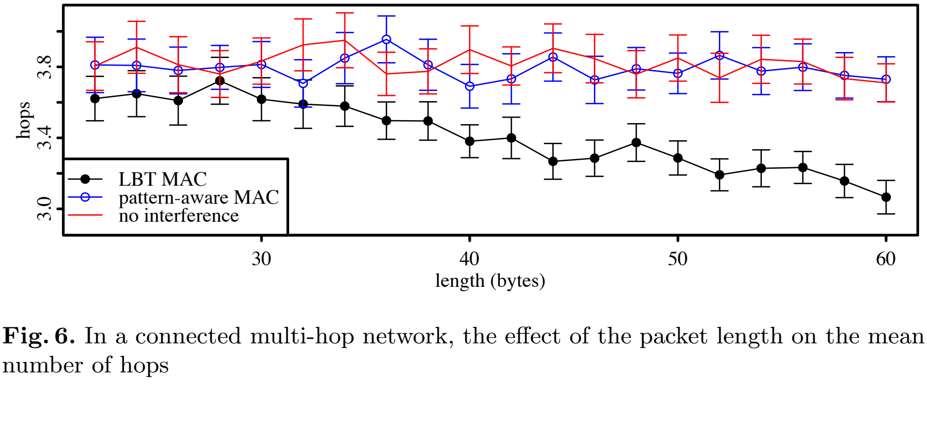


Fig. 6. In a connected multi-hop network, the effect of the packet length on the mean number of hops

increases. The significant number of packet losses cause this behaviour: packets are more likely to be lost on the long paths, and these lost packets will not factor into the latency calculations.

Our final graph shows the effect of varying the packet generation rate on the PRRs and latency (Figure 7). In these experiments, we set the packet length to its maximum (60 bytes) in order to accentuate the variable's effect.

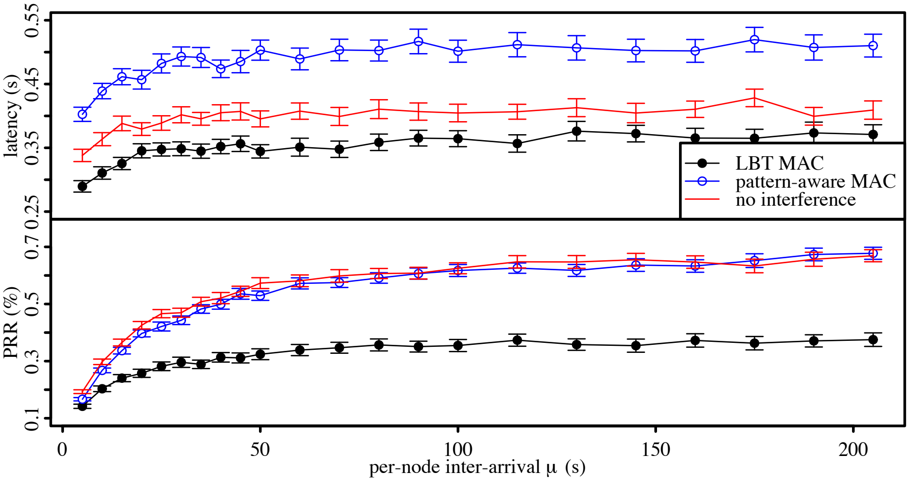


Fig. 7. In a connected multi-hop network, the effect of the mean latency between per-transmitter packet introductions on the packet reception ratio and the latency

Here, the PRR rate follows a similar trend to the single-hop case just at significantly lower levels. Unlike with the single-hop case, the latency curve again increases as we slow the rate of packet generation. As with the packet lengths, less congestion results in an increased number of the long paths succeeding which subsequently increase the latency.

In summary, the results demonstrate the benefits of using interference in a constructive manner. The benefits are evident even if used to augment a trivial MAC protocol, such as a rudimentary LBT. Naturally, more elaborate schemes can be devised. Suffice is to say that the impulse interference is the basis of synchronization around which a self-organizing TDMA-like MAC protocol could eventually be constructed.

6 Conclusion

In this paper, based on our previous work on simplification of the Lomb periodogram for the post-deployment identification of frequent impulsive interference, we extend an existing simulator with a flexible interface for the production of impulsive interference. Subsequently, we incorporated the impulse classifier and a proof-of-concept pattern-aware MAC (PA-MAC) into the simulator and simulated a variety of different configurations. We found that PA-MAC could improve the packet reception rates in both single- and multi-hop environments at the cost of increased latency.

In terms of future work, we plan to explore protocols that would allow nodes to come to a consensus about the channel classification. An immediate result from this would be the weakening of our correlation assumption. Such a protocol would allow nodes to join the network without pausing communication while the evaluation occurs. Moreover, it may make sense to delegate the task of channel sampling solely to a subset of nodes (possibly the ones that have better energy reserves) while providing a way of disseminating the information about interference patterns to the rest of the network. Generally, the problem of optimal collaborative identification of interference patterns and selective dissemination of knowledge (not all nodes need to receive the same information) appears as an interesting topic for a further study.

References

1. Do, J., Akos, D., Enge, P.: L and S bands spectrum survey in the San Francisco Bay area. In: PLANS 2004: Position Location and Navigation Symposium, pp. 566–572 (2004)
2. Boers, N.M., Chodos, D., Huang, J., Stroulia, E., Gburzynski, P., Nikolaidis, I.: The Smart Condo: Visualizing independent living environments in a virtual world. In: PervasiveHealth 2009: Proceedings from the 3rd International Conference on Pervasive Computing Technologies for Healthcare, London, UK (April 2009)
3. Stroulia, E., Chodos, D., Boers, N.M., Huang, J., Gburzynski, P., Nikolaidis, I.: Software engineering for health education and care delivery systems: The Smart Condo project. In: SEHC 2009: Proceedings from the 31st International Conference on Software Engineering, Vancouver, Canada (2009)
4. Boers, N.M., Nikolaidis, I., Gburzynski, P.: Patterns in the RSSI traces from an indoor urban environment. In: CAMAD 2010: IEEE 14th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Coconut Creek, FL, December 3-4 (2010)
5. Vieira, M., Coelho Jr., C.N., da Silva Junior, D.C., da Mata, J.: Survey on wireless sensor network devices. In: ETFA 2003: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, pp. 537–544 (September 2003)
6. Oetting, J.: A comparison of modulation techniques for digital radio. IEEE Transactions on Communications 27(12), 1752–1762 (1979)
7. Boers, N.M., Nikolaidis, I., Gburzynski, P.: Sampling and classifying interference patterns in a wireless sensor network. ACM Transactions on Sensor Networks (to appear)
8. Srinivasan, K., Dutta, P., Tavakoli, A., Levis, P.: An empirical study of low-power wireless. ACM Transactions on Sensor Networks 6(2), 1–49 (2010)
9. Chandra, A.: Measurements of radio impulsive noise from various sources in an indoor environment at 900 MHz and 1800 MHz. In: 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 639–643 (September 2002)

10. Srinivasan, K., Dutta, P., Tavakoli, A., Levis, P.: Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In: SenSys 2006: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 419–420. ACM, New York (2006)
11. Lee, H., Cerpa, A., Levis, P.: Improving wireless simulation through noise modeling. In: IPSN 2007: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, pp. 21–30. ACM, New York (2007)
12. Rusak, T., Levis, P.: Physically-based models of low-power wireless links using signal power simulation. Computer Networks 54(4), 658–673 (2010)
13. Akhmetshina, E., Gburzynski, P., Vizeacoumar, F.: PicOS: A tiny operating system for extremely small embedded platforms. In: Arabnia, H.R., Yang, L.T. (eds.) Embedded Systems and Applications, pp. 116–122. CSREA Press (2003)
14. Gburzynski, P., Nikolaidis, I.: Wireless network simulation extensions in SMURPH/SIDE. In: WSC 2006: Proceedings of the 2006 Winter Simulation Conference, Monterey, California (December 2006)
15. Boers, N.M., Gburzynski, P., Nikolaidis, I., Olesinski, W.: Developing wireless sensor network applications in a virtual environment. Telecommunication Systems 45(2), 165–176 (2010)
16. Srinivasan, K., Jain, M., Choi, J.I., Azim, T., Kim, E.S., Levis, P., Krishnamachari, B.: The κ-factor: Inferring protocol performance using inter-link reception correlation. In: MobiCom 2010: Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, pp. 317–328. ACM, USA (2010)
17. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.A.: Impact of radio irregularity on wireless sensor networks. In: MobiSys 2004: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 125–138. ACM, USA (2004)
18. Olesinski, W., Rahman, A., Gburzynski, P.: TARP: A tiny ad-hoc routing protocol for wireless networks. In: ATNAC 2003: Proceedings of Australian Telecommunications Networks and Applications Conference, Melbourne, Australia, December 8-10 (2003)

Resilient Secure Localization and Detection of Colluding Attackers in WSNs

Wei Shi[1], Meng Yao[2], and Jean-Pierre Corriveau[2]

[1] University of Ontario Institute of Technology, Oshawa, L1H 7K4, Canada
wei.shi@uoit.ca
[2] Carleton University, Ottawa K1S5B6, Canada
mengyao86@gmail.com, jeanpier@scs.carleton.ca

Abstract. There exists extensive work in wireless sensor networks (WSNs) on security measures that guarantee the correctness of the estimation of the position of a node despite attacks from adversaries. But very little work has investigated how colluding attackers can modify the behavior of known attacks or even create new ones. In this paper, we first present an attack model that allows three types of colluding attackers to threaten the secure localization process and/or the attacker detection process. We then describe a decentralized algorithm that is used to determine the position of a location-unknown sensor U despite the presence of colluding attackers (that can alter any type of information being exchanged in a WSN in order to form an attack jointly). Most importantly, the proposed algorithm allows U to detect such colluding attackers in its sensing range. Our simulation results show that in both a uniformly deployed WSN environment and in a randomly deployed one, our Super Cross Check algorithm can achieve a high success rate for both secure localization and detection of colluding attackers.

Keywords: Secure Localization, Colluding Attacker Detection, Wireless Sensor Networks.

1 Introduction

1.1 Background

Wireless sensor networks (WSNs) have enabled a new form of communication between tiny embedded devices equipped with sensing capabilities. Such sensors act as the nodes of an ad-hoc network in which communication relies on a distributed collaborative exchange of information. Applications for WSNs range from environmental and health monitoring, to home networking and tracking systems (for objects, animals, humans, and vehicles). And in many WSNs applications, the positions of unknown (or equivalently, location-unknown) nodes play a critical role. Moreover, many fundamental techniques in WSNs (such as geographical routing, geographic key distribution, and location-based authentication) require determining the positions of unknown nodes. When a WSN is deployed in an unattended and/or hostile environment, it is vulnerable to

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 181–192, 2012.

threats and risks. Many attacks (such as wormhole, sinkhole and sybil ones) make the estimated positions incorrect. Such incorrect positions may have severe consequences in many applications. For example, a battlefield surveillance system incorrectly reporting enemy movement or wrongly identifying an ally as an enemy; a patient monitoring system sending the wrong location of a patient in critical condition; a forest fire monitoring system incorrectly reporting the location of a fire; a nuclear reactor monitoring system locating erroneously a malfunction. Thus, a secure localization scheme, that is, one that guarantees the accuracy of computed locations, is absolutely required.

Usually, there are two steps in a localization process: information acquisition and position calculation. Most adversaries attack the first step of a localization process. An adversary can either a) corrupt normal nodes into sending false localization information, or b) pretend to be a legitimate node in order to forge, alter or replay communication data. Such attacks will lead to inaccurate localization calculations (regardless of whether it is a centralized authority node that calculates the location of a location-unknown node, or such a node calculates its own location locally). Consequently, security measures have been extensively studied in order to make estimated positions correct despite attacks from an adversary. But several questions remain, in particular: a) What happens when several adversaries collude? and b) Can the attack model change and, if so, what kind of damage can it inflict on the localization process?

1.2 Related Work

Meadows et al. [1] analyze existing techniques for collusion prevention, and show how these techniques are inadequate for addressing the issue of collusion in sensor networks. Wang et al. [2] propose a novel localization algorithm called TMCA. This is a distributed algorithm based on the cooperation of non-beacon neighbor nodes. It is robust against some known attacks such as the wormhole, sybil and replay attacks. Even when there are more malicious anchor nodes than benign anchor nodes in a WSN, TMCA can still generate adequate localization results. The algorithm calculates an unknown node S's location using a distance bounding technique when S receives the coordinates (x_i, y_i) and distance d_i from a reference beacon. The Maximum Likelihood Estimate technique is used to receive a reasonably precise location of S. However, despite the algorithm being called "Tolerant Majority Colluding Attacks", there is no evidence of collaboration (e.g., exchange of messages) between attackers to form an attack jointly.

In [3], Garcia-Alfaro et al. introduce algorithms that enable the unknown nodes to determine their positions in the presence of neighbor sensors that may lie about their locations. In algorithm *Majority-Three Neighbor Signals*, all the neighbor anchor nodes of a sensor advertise their locations. For every three anchor nodes, the unknown node uses trilateration [4] to calculate a position. Then, a majority decision rule is used to obtain the final position of the unknown node. All triplets that compute a location different from this final one have their nodes considered to be liars. In [5,6], an Evil Ring (ER) attack is introduced. An evil ring attacker who lies about its position can successfully fool all the sensors that use trilateration to obtain or verify their locations. Algorithm *Cross Check* is presented to detect such attackers. The evil ring is an attack on the location determination algorithms of Garcia-Alfaro et al. When inquired, an attacker returns a fake location sitting on a circle centered at the victims location and with radius equal to the attacker-victim separation distance. The calculation of the distance between the victim and the attacker is not affected. A location-unknown node correctly determines its location. The attack, however, misleads such as node in getting and using wrong locations for its malicious neighbors. An evil ring attacker who lies about its position can successfully fool all the sensors that use trilateration to obtain or verify their locations. Algorithm *Cross Check* is presented to detect such attackers. Its main steps are:

- Request locations: Location-unknown node U sends using broadcast a location request to all the other nodes in the neighborhood.
- Calculate location: Using every possible three neighbor combination and their distances, node U calculates a location (x, y) according to the majority decision rule.
- Build a *cross-check* list: All neighbors in triplets in agreement with the majority are added to the *cross-check* list. The accepted location and *cross-check* list are sent using broadcast to neighbors.
- Liar detection: Node U waits until it receives two *cross-check* lists from two different neighbors. Every node present with identical location in all three *cross-check* lists is added to the neighbor table. Otherwise, it is added to the list of liars.

Most importantly, however, both [3] and [5,6] assume a dense network in which no sensor collusion exists.

In “Collaborative Collusion” [7], the CCAM model is proposed. In that model all malicious nodes can collaborate with each other to alter the location information they receive and/or jointly forward it. The authors present a solution to detect such malicious nodes. However their algorithm TSFD cannot detect the ER attackers introduced in [5,6]. Also, their proposed solution rests on the existence of a trusted base station that periodically broadcasts trusted grids to all nodes. In fact, attacker detection is performed only by this base station. Such calculation at a central node has several drawbacks. First, in order to forward the location information to a central node, a route to the latter must be known. This implies the use of a non-location-based routing protocol, which entails an additional communication cost. Second, because of the large volume of traffic to and from the central node, the battery lifetime of the nodes around the central node will be seriously impacted. Third, centralization hinders the robustness of the system: if the routes to the central node are broken, the nodes will not be able to communicate their location information to the central node and vice versa. In summary, a centralized implementation will not only reduce a network’s lifetime, but it will also increase its complexity and compromise its robustness. On the other hand, if location estimation takes place at each node, in a distributed manner, such problems can be alleviated [8].

1.3 Our Contribution

Very little work has been done on investigating how colluding attackers can change the behavior of known attacks or even create new attacks. Depending on the nature of each secure localization algorithm, sensors could collaboratively elude the location-unknown sensors by: 1) jointly announcing false information or 2) forming an illegal position (physical) pattern (e.g., more than two sensors are collinear). Either one of these two scenarios could lead to an erroneous calculation of a position. Furthermore, beyond location information, the colluding attackers can also alter other information or signals in the WSN under attack. For example, reputation lists are used in [2, 5, 6, 9, 10] in order to support different secure locational algorithms. In [2,5,6], the consistency of reputation lists is checked in order to detect malicious sensor nodes. All existing solutions assume that such reputation lists are not corrupted and that there are no colluding attackers that can jointly compromise this consistency verification procedure. In contrast, in this paper, we present a decentralized algorithm that is used to determine the position of a location-unknown sensor U when there are colluding attackers that can alter all types of information being exchanged in the WSN in order to form attacks jointly. Most importantly, the proposed algorithm allows U to detect such colluding attackers in its sensing range.

2 Colluding Attackers Detection Algorithm: Super Cross Check (SCC)

2.1 Attack Model and Assumptions

Let $\mathcal{K}$ be a set of location-known sensor nodes. Let $\mathcal{A}$ be a set of Anchor nodes, which know their own positions beforehand by either using GPS or being manually configuration [11, 12]. And let $\mathcal{S}$ be a set of regular sensor nodes, where $\mathcal{S} \subset \mathcal{K}$ and $\mathcal{A} \subset \mathcal{K}$. $\mathcal{U}$ denotes a set of location-unknown sensor nodes. Each location-unknown sensor $U \in \mathcal{U}$ can measure accurately the distance between itself and any other node in its sensing range. All sensor nodes are deployed on a $2-$dimensional plane $\mathcal{G}$. In $\mathcal{K}$, there are sensors (hereafter called *liars*), including Evil Ring attackers [5, 6], that can lie about their locations and any other information being exchanged with neighboring nodes in a liar's sensing range. Such lying behavior may be the result of malicious attacks or an unintentional act due to a sensor's physical malfunctioning. There are upper bounds on the number of tolerable liars, otherwise the algorithms in [3,5,6] fail. As a function of the liar number, Table 1 lists the minimum number of neighbors required to determine a location. We call a *liar* $C_i \in \mathcal{C}, \mathcal{C} \subset \mathcal{K}$ a *Colluding Attacker* if C_i and one or more other *liar(s)* jointly form an attack. A *Colluding Attacker* can be either a regular node or an Anchor node.

In this paper, we present a solution that, despite the presence of colluding attackers, allows a location-unknown sensor U to obtain its position and detect such attackers within its sensing range. In order to detect colluding attackers, we must know what kind of threatening behavior (affecting the accuracy of the secure localization process) sensor collusion can bring into the WSN. Clearly, the less information being exchanged between sensor nodes, the less chance adversaries have to attack successfully. In our solution the only messages being exchanged between sensor nodes are the coordinates of each sensor and its *Cross Check Lists* (hereafter *CC* lists). In this paper, we do not focus on a colluder's ability to attack the system by corrupting periodic 'Hello' messages (used to check if neighbors are alive and to exchange routing information).

We organize colluding attackers into three categories. Attacks from attackers in all three categories involve, in part, sending out an altered *CC* list. Beyond having this common behavior, further categorization depends on how an attacker lies about its location during the localization process:

1. a liar lies about its location by using a randomly generated fake location;
2. a liar lies about its location by giving out a fake location: in cooperation with two other attackers, it returns the location of a location-unknown sensor $U \in \mathcal{U}$.
3. a liar lies about its location by giving out a fake location that sits on a circle centered at the victim's location and of a radius corresponding to the distance between the attacker and victim. The attack succeeds and is undetectable by any existing *liar* detection algorithm, except for the one presented in [5,6]. Namely, only algorithm *Cross Check* can detect such an *Evil Ring* attack. But this type of colluding attackers, like in the other categories, also send out colluded *CC* lists, which algorithm *Cross Check* cannot address.

In the rest of this paper, we will focus on explaining how to deal with the third category of liars. This is because an algorithm that can detect such liars (of our third category) can *de facto* handle the first two categories of liars. These first two categories of attackers must still be identified because they use different attacking behaviors to jeopardize the localization process, and such differences must be simulated.

Also, we will compare our proposed solution only with the algorithm Cross Check presented in [5,6] since only that algorithm can detect non-colluding liars similar to those of our third category.

Table 1. Minimum number of location-aware neighbors required to determine a correct location, as a function of the number of liars [3]

Number of Liars	Min Number of Neighbors
1	7
2	11
3	16
4	21
5	26
10	31
15	74
20	98

We postulate that in order for a location-unknown node U to systematically detect a third category colluding attacker, the number of non-colluding attackers in the intersection of the sensing circles of U and A should be at least two more than the number of colluding attackers in U's sensing range.

2.2 SCC Algorithm

There are two steps in this algorithm:

- obtain the position of a location-known sensor $k \in \mathcal{K}$
- detect the colluding attackers.

Step one: calculate the position and create a CC list for each U. In this first step (Lines 1-9 in Algorithm 1 below), we use the trilateration technique to calculate the location of each U. We then used a majority decision rule, as used in [5,6], to obtain the final position of the unknown nodes.

Step 2: exchange CC lists and detect the colluding attackers. In this second step (Lines 10-29 in Algorithm 1), upon receiving a request for its CC list, each $k \in \mathcal{K}$ sends out its CC list. If it does not have a CC list, it will construct one the same way a location-unknown sensor U does. Naturally, if k is a colluding attacker, it will send out a CC list that does not reflect its real calculation results. For example, it may purposely delete a few sensors that should have been in the list, in order to give the illusion that these deleted sensors could be colluding attackers. This attack could succeed if there were several such colluding attackers lying about the same fact. This second step of the algorithm detects the three categories of colluding attackers by using a voting technique: U requests a CC list from its neighbor nodes. Upon receiving a CC list L_i, U gives a positive credit to a node $k \in K$ if this node (that is, its coordinates) is in both U's CC list and L_i, a negative credit to k otherwise. Once U receives all the CC lists, it will compute the number of positive and negative credits of each neighboring node. If a neighboring node k received two more positive credits than negative credits, U can conclude that k is not a colluding attacker and U can use k to construct its routing table. Otherwise k is identified as a colluding attacker.

The pseudo code for this algorithm is shown in Algorithm 1.

3 Simulation and Evaluation

In this section, simulation results are presented and analyzed. The simulations are performed using Omnet 4++. We conduct tests under the following two scenarios: 1) uniformly deployed Anchor nodes and regular sensor nodes and 2) randomly deployed sensor nodes. We evaluate the performance of our proposed algorithm by measuring the success rate for the detection of colluding attackers and by comparing our results with those of the Cross Check algorithm presented in [6]. We explain the details of these two scenarios separately.

Algorithm 1. Super Cross Check

```
repeat
    Request neighbors' locations.
    for all triplets of neighbors ($V_1$; $V_2$; $V_3$) do Compute the intersection point of the three circles centered at $V_1$; $V_2$; $V_3$ with radius $d_1$, $d_2$, $d_3$.
    end for
until there is a consensus on $(x, y)$ determined by the majority of triplets.
Accept $(x, y)$ as the location of $U$.
for all triplets of neighbors ($V_1$; $V_2$; $V_3$) in agreement with the majority do
    Add the locations $V_1$; $V_2$; $V_3$ to $U$'s $CC$ list.
end for
Broadcast location of $U$ and its $CC$ list.
request $CC$ lists from all the neighbors that are in $U$'s $CC$ list
for each neighbor $i$ on $U$'s $CC$ list (referred to as $I$) do
    Select all sensors in the intersection of the two sensing circles of $U$ and $i$
    for all the selected sensors (referred to as $c \in C$) do
        compare the $CC$ list from $c$ with the one from $U$
        if $i$ is in both $CC$ lists then
            give a positive credit to $i$
        else if $i$ is not in the $CC$ list received from $i$ then
            give a negative credit for $i$
        end if
    end for
end for
for each $i \in I$ do
    if $i$ received 2 more positive credits than negative credits then
        this node is not a colluding attacker and it can be used to construct $U$'s routing table
    else
        this node is a colluding attacker
    end if
end for
```

3.1 Uniform Sensor Deployment

In this section, we consider the situation in which all sensors are uniformly deployed in a WSN. In Figure 1, solid black dots represent Anchor nodes, which have $\mathcal{R}_a = 2d$ with $(d > 0, d = \overline{AB})$ as their sensing range; and grey dots represent the regular location-known sensors, which have $\mathcal{R}_r = \sqrt{2}d$ with $(d > 0)$ as their sensing range. Any of these Anchor nodes and regular sensor nodes could be colluding attackers. Additionally, each black and white dot represents a location-unknown sensor node that also has $\mathcal{R}_r = \sqrt{2}d$ with $(d > 0)$ as its sensing range. A key observation is that, in the worst case, all the colluding attackers in the WSN under attack belong to our third category.

As mentioned earlier, in order for a location-unknown node U to systematically detect a third category colluding attacker A (or B) in Figure 1, the number of non-colluding attackers in the intersection of the sensing circles of U and A should be at least two more than the number of colluding attackers in U's

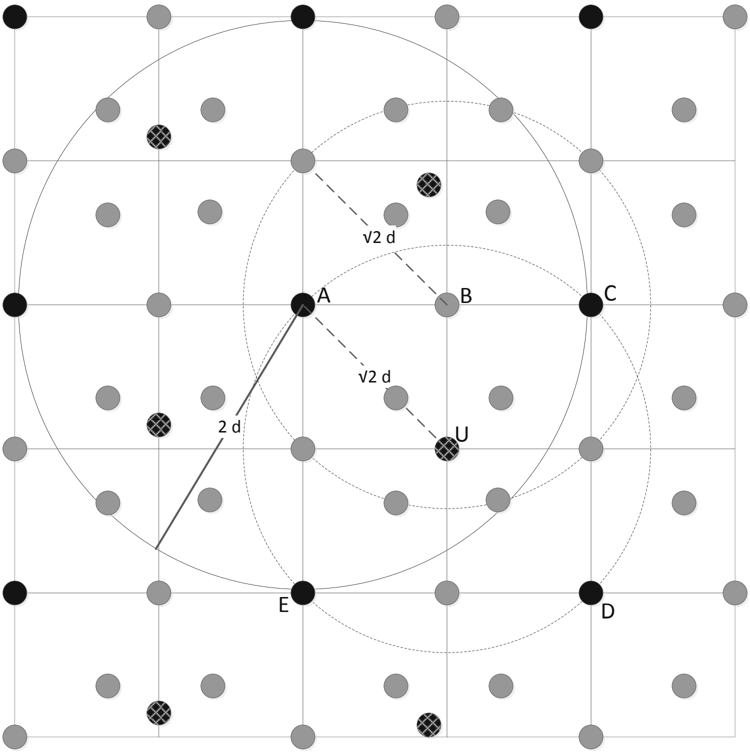


Fig. 1. Uniform deployment

sensing range. In our proposed uniform deployment WSN, this assumption leads to approximately 33.3% of third category colluding attackers in the total number of location-known sensor nodes. We focus here on the success rate of detecting third category colluding attackers after executing algorithm Super Cross Check and then compare these results against the ones of algorithm Cross Check. Our simulation results are presented in Figure 2. The solid line shows that in the proposed uniformly deployed WSN, the success rate of detecting colluding attackers (which will all be third category colluding attackers in the worst case) after executing algorithm Super Cross Check is 100% consistently, when the percentage of colluding attackers does not exceed 33%. We also observe that in order to keep a 100% colluding attacker detection success rate, the percentage of third category colluding attackers among all the colluding attackers should be inversely proportional to the percentage of colluding attackers among all location-known sensor nodes. In other words, beyond 33% of third category colluding attackers, having a higher percentage of colluding attackers from the two first categories among all the location-known sensor nodes does not affect the colluding attacker detection success rate. The dashed line in the figure corresponds to the performance of algorithm Cross Check under the same setup. The results of algorithm Cross Check show that in each cell of the grid (e.g. cell $ACDE$ in Figure 1), as long as

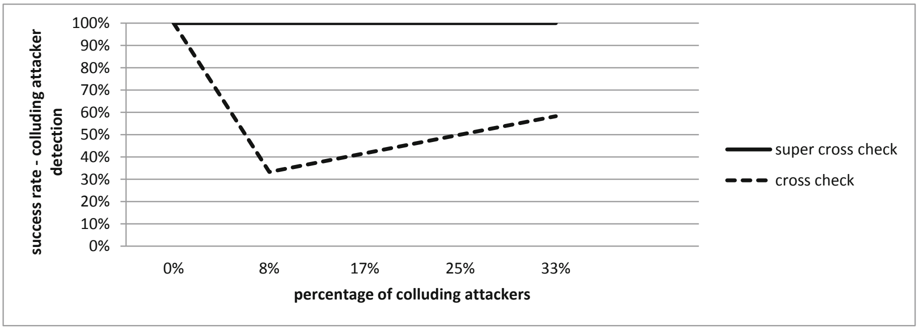


Fig. 2. Comparison of colluding attacker detection success rate between algorithms Super Cross Check and Cross Check in a uniformly deployed WSN

there are more than 1 colluding attackers, the success rate of detecting colluding attackers will drop drastically (to as low as 30%). Interestingly, we observe that when the percentage of colluding attackers among all location-known sensor nodes is between 8% to 33%, the success rate increases. However, after careful analysis, we conclude this increase does not correlate to the ability of algorithm Cross Check to detect colluding attackers. Instead, this rate increase stems from the fact that the algorithm will wrongly label some nodes as colluding attackers. Thus, the more genuine colluding attackers present in the WSN, the more nodes labeled arbitrarily by algorithm Cross Check as colluding attackers will end up being real colluding attackers, thus increasing the detection rate.

3.2 Random Sensor Deployment

In this section, we consider the situation in which all sensors are randomly deployed in a WSN. We compare the performance of algorithm Super Cross Check and algorithm Cross Check with respect to the following two aspects: the success rate of localizing location-unknown sensors and the success rate of detecting colluding attackers.

Figure 3 shows the success rate of detecting colluding attackers using algorithm Super Cross Check in a randomly deployed WSN in which there are 80 non-colluding attackers and 10 location-unknown sensors. We can see from this figure that the success rate drops to 90% when the number of colluding attackers is around 50, which entails that 1 of the 10 location-unknown sensors under test did not receive its correct location. This is because in this random deployment setup, the number of colluding attackers exceeds the number of non-colluding attackers. Otherwise, algorithm Super Cross Check's success rate for secure localization of location-unknown sensor nodes in a randomly deployed WSN is 100%.

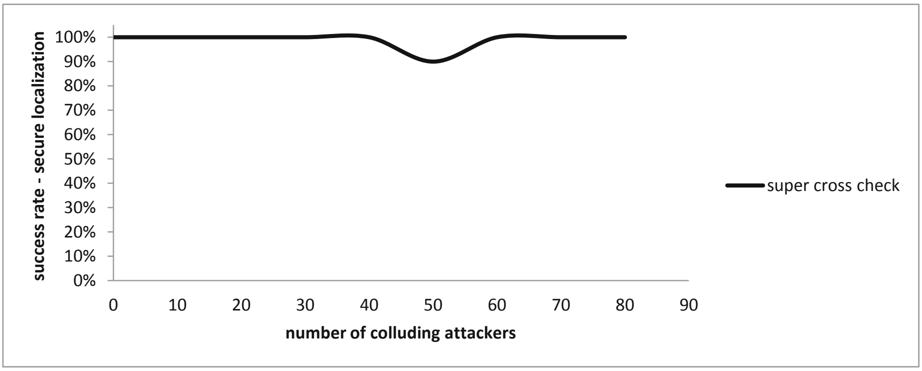


Fig. 3. Secure localization success rate of algorithm Super Cross Check in a randomly deployed WSN

Figure 4 illustrates the success rate of detecting colluding attackers using algorithm Super Cross Check, when there are 80 non-attacking nodes (location-known sensors) and 10 location-unknown nodes in the WSN. We let the ratio of the three categories of colluding attackers be 3 : 4 : 3 and the total number of colluding attackers increase from 0 to 80. The results show that algorithm Super Cross Check's success rate at detecting colluding attackers is never lower than 93%, while the success rate at detecting colluding attackers of algorithm Cross Check varies between 22% to 42%.

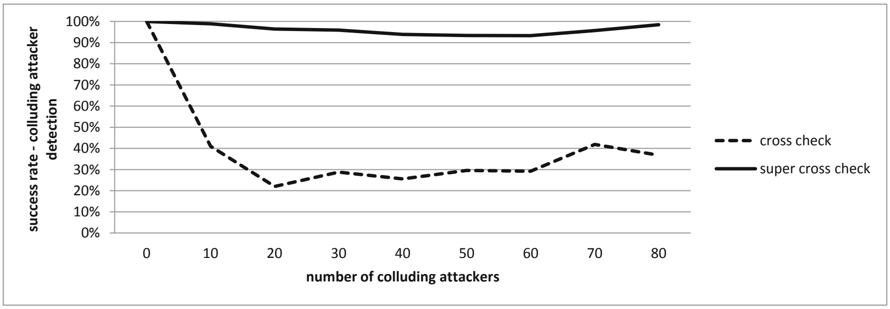


Fig. 4. Comparison of colluding attacker detection success rate between algorithms Super Cross Check and Cross Check in a randomly deployed WSN

4 Conclusions

When a WSN is deployed in unattended and/or hostile environments, it is vulnerable to threats and risks. Many attacks make the estimated positions incorrect.

Such incorrect positions may lead to severe consequences in many applications. Thus, security measures have been studied extensively in order to make the estimated positions correct despite the attacks from an adversary. But very little work has been done on investigating how colluding attackers can change the behavior of known attacks or even create new attacks. In this paper, we present an attack model that allows three types of colluding attackers to attack the secure localization process and/or the attacker detection process. We then present a decentralized algorithm that is used to determine the position of a location-unknown sensor U when there are colluding attackers that can alter all types of information being exchanged in the WSN in order to form attacks jointly. Most importantly, the proposed algorithm allows U to detect such colluding attackers in its sensing range. The simulation results show that in both uniformly deployed and randomly deployed WSN environments, our algorithm Super Cross Check achieves a significantly high success rate for both secure localization and colluding attackers detection.

References

1. Meadows, C., Poovendran, R., Pavlovic, D., Chang, L., Syverson, P.: Distance bounding protocols: Authentication logic analysis and collusion attacks. In: Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks. Springer (2007)
2. Wang, X., Qian, L., Jian, H.: Tolerant majority colluding attacks for secure localization in wireless sensor networks. In: 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–5 (2009)
3. Garcia-Alfaro, J., Barbeau, M., Kranakis, E.: Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes. In: Annals of Telecommunications, pp. 1–18. Springer (2011), doi: 10.1007/s12243-010-0221-z
4. Niculescu, D., Nath, B.: Ad hoc positioning system (APS). In: The 2001 IEEE Global Telecommunications Conference of the IEEE Communications Society, pp. 2926–2931 (2001)
5. Shi, W., Barbeau, M., Garcia-Alfaro, J., Corriveau, J.-P.: Detection of the Evil Ring Attack in Wireless Sensor Networks Using Cross Verification. In: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2010), Montreal, Canada, 8 pages (June 2010)
6. Shi, W., Barbeau, M., Garcia-Alfaro, J., Corriveau, J.-P.: Handling the Evil Ring Attack on Localization and Routing in Wireless Sensor Networks. Journal of Ad Hoc & Sensor Wireless Networks (to appear, 2012)
7. Jiang, J., Han, G., Shu, L., Chao, H., Nishio, S.: A novel secure localization scheme against collaborative collusion in wireless sensor networks. In: 7th International Wireless Communications & Mobile Computing Conference, pp. 308–313 (July 2011)
8. Savvides, A., Han, C., Strivastava, M.B.: Dynamic fine-grained localization in Ad-Hoc networks of sensors. In: 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), pp. 166–179. ACM, New York (2001)
9. Liu, D.G., Ning, P., Du, W.: Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In: 25th Int. Conf. on Distributed Computing Systems (ICDCS), pp. 609–691. IEEE Computer Society Press, Washington (2005)

10. Srinivasan, A., Wu, J., Teitelbaum, J.: Distributed reputation-based secure localization in sensor networks. Journal of Autonomic and Trusted Computing (2007)
11. Du, Q., Qian, Z., Jiang, H., Wang, S.: Localization of Anchor Nodes for Wireless Sensor Networks. In: New Technologies, Mobility and Security (NTMS 2008), pp. 1–5 (2008)
12. Tian, S., Zhang, X., Wang, X., Sun, P., Zhang, H.: A Selective Anchor Node Localization Algorithm for Wireless Sensor Networks. In: International Conference on Convergence Information Technology, pp. 358–362 (2007)

Low Cost Data Gathering Using Mobile Hybrid Sensor Networks

Dan Tao[1], Shaojie Tang[2], and Huadong Ma[3]

[1] School of Electronic and Information Engineering,
Beijing Jiaotong University, Beijing, China
[2] Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
[3] Beijing Key Laboratory of Intelligent Telecomm. Software and Multimedia,
Beijing University of Posts and Telecomm., Beijing, China

Abstract. In this work we study energy efficient hybrid sensor network design using mobile sinks, motivated by the practical GreenObs system application. In our model, the movement of mobile sinks is constrained to be on some predefined road-segments. Two different network structures are investigated: the one-hop structure in which each static sensor can be reached by the mobile sink at some stage of the movement, and the multi-hop structure where some sensors need the relay by other sensors to reach the sink. The challenge is to find a movement schedule of mobile sink that will minimize the energy cost while meet other constraints. In this work, we first show that the problem is NP-hard and then design an efficient movement scheme and theoretically prove that the total cost is within a constant factor of the optimum. We further present a scheduling solution using integer program for multi-hop structure, which is near optimal and can be computed in polynomial time. Finally, we conduct extensive study of our method in a real wireless sensor network deployment composed of hundreds of static sensors. Our experiments validate the theoretical findings of our method.

Keywords: mobile hybrid sensor networks, data gathering, mobile sink, group steiner tree, flow network.

1 Introduction

The emergence of large-scale wireless sensor networks revolutionizes information gathering and processing paradigm in a variety of application scenarios. One of the most fundamental challenges in such data gathering is energy efficiency. In conventional flat topology of sensor networks, data gathering paradigm is relay routing, where sensing data are routed to a remote static sink via some relay sensors with the objective of minimizing overall energy expenditure or balancing energy cost among multiple sensors. In this case, sensor nodes around the sink inevitably drain their energies ahead of others because of heavier data forwarding, and thus the arisen coverage/commucation holes will lead to a disconnected and dysfunctional network. Considering the weakness mentioned above, intelligent, mobile sink has been introduced into the conventional static sensor networks.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 193–206, 2012.

Mobile sink helps to prolong the lifetime of network by alleviating the data forwarding burden, and thus balances the energy consumption of network [1]. In this paper, we address low cost data gathering with mobile hybrid sensor networks. A network consists of static sensors which have "sensing" ability and mobile sink which has "moving" ability.

In general, the energy consumption of a hybrid network mainly comes from two parts: the principal one is on data forwarding among static sensors, and the secondary one is consumed on motion by mobile sink. For a static sensor, the energy expenditure of sensing information is relatively stable as it mainly depends on the sampling [2]. In contrast, its main energy expenditure is caused by radio transmission. Intuitively, if a static sensor only transmits its own sensing data to the sink without relaying the data from other static sensors, it will have the minimum energy consumption. Motivated by this intuition, we first investigate a simple one-hop structure in which each static sensor can be reached by the mobile sink at some stage of the movement. Particularly, each static sensor in the hybrid network is capable of controlling its discrete transmitter power levels to the lowest value to reach certain road-segment which a mobile sink walks along with the lowest energy cost, which is different from the assumption that each static sensor has uniform transmission ability in [3]. In this scenario, we aim to finding a motion path with the minimum length for mobile sink to visit each static sensor to minimize the motion energy consumption of the hybrid network.

As a matter of fact, it is hard to guarantee that all the static sensors sparsely deployed can exchange data with mobile sink via one-hop transmission even if they can flexibly control transmitter power levels. Static sensors that are too far away from the road-segments need the relay by other static sensors to reach the sink. In this case, the total energy consumption of a hybrid sensor network mainly depends on the part from energy-limited static sensors. From the perspective of minimizing data transmission energy cost, we aim to finding a scheduling algorithm to gather and transmit the data from all the static sensors so that the system lifetime can be maximized.

Recently, a series of research has been conducted and a wide variety data gathering schemes have been proposed in the context of mobile hybrid sensor networks. The majority of research on mobile data gathering is based on the assumption that the trajectory of mobile sink is arbitrary in a large monitoring field [2] [4] [5]. In reality, there likely exist natural obstacles, such as stones, bushes and lakes and man-made structures. To avoid these obstacles, the mobile sink may have to move within some constrained regions. In our application [6][1], sensors are statically deployed in a large-scale forest, and the forest patrol guards carrying with mobile sink node need to collect physical environment information (*e.g.* temperature, humidity and optical intensity) from all static sensors one by one. Mobile sink node can be equipped with a powerful transceiver, battery and large memory. As a forest patrol guard moves in close proximity to static

[1] GreenOrbs is one of a wide variety of surveillance and dissemination applications that span large geographic areas. http://greenorbs.org/

sensors, data is transferred to the mobile sink, and then mobile sink returns and uploads data to the base station for further processing. Instead of walking arbitrary, forest patrol guards are given a set of deterministic roads. One of the key challenges in such data gathering is to conserve sensors' energies, so as to maximize their lifetime.

Taking the constrained moving paths of mobile sink into account, we study energy-efficient hybrid sensor network design using mobile sink with the objective to minimize the energy cost of a network. The main contributions of this paper are summarized as follows: Firstly, we study the energy cost performance for data gathering in a one-hop structure. We also show that the problem is NP-hard and then design a movement scheduling of mobile sink that will minimize the energy cost and theoretically prove that the total tour length is within a constant factor of the optimum. Secondly, we present an integer programming to formulate the mobile data gathering in multi-hop structure so that the lifetime of static network can be maximized with energy constraint on each static sensor. Thirdly, we conduct extensive study of our solutions in practical GreenOrbs system composed of hundreds of static sensors, and a series of experiments verify the theoretical findings of our solutions.

The remainder of this paper is organized as follows. Section 2 presents system model and problem formulation. Section 3 and Section 4 respectively discuss energy-efficient single-hop and multi-hop data gathering schemes. Simulations and numerical results are given in Section 5. Section 6 reviews related work, and we conclude this paper in Section 7.

2 Preliminaries and Problem Formulation

2.1 Network Model

In this paper, we consider a network of n static sensors $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ and one mobile sink m deployed over a two-dimensional monitoring area. The locations of the static sensors are known a priori. Assume that there exists a road map M, which indicates the feasible paths for mobile sink to move along, as the gray strips illustrated in Fig.1. The monitoring field can be partitioned into several subregions by walking roads in the map.

A mobile sink starts from *starting point*, traverses the monitoring area along road map, and gathers data from static sensors if and only if they are within each other's transmission range, and then returns. We define the certain point(s) that mobile sink can stop for data gathering from static sensors as *anchor point(s)*. When a mobile sink arrives at an anchor point, some static sensors in the neighborhood that receive the gathering request message from the mobile sink can upload data packets through short-range communication.

2.2 Radio Model

Assume that static sensor and mobile sink are based on a disk radio model, and the radio transceiver in each static sensor can be tuned to m discrete power

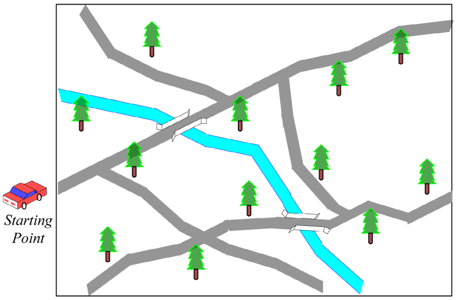


Fig. 1. Road map

levels [7] (see Fig.2), *i.e.* $\{p_1, p_2, \cdots, p_m\}$, with each power level corresponding to a unique transmission range $\{r_1, r_2, \cdots, r_m\}$. The maximum power level and transmission range are denoted by p_m and r_m respectively. In the stage of data gathering, the radios have power control and can expend the minimum required power to reach the intended recipients (*e.g.* restricted paths). The powerful transceiver of mobile sink ensure that its transmission range R is always larger than the maximum transmission range r_m of static sensors. Given the distance $d_{i,j}$ between node (static sensors or mobile sink) i and j, we can convert $d_{i,j}$ into the minimal available transmission range which is greater than $d_{i,j}$ by function $F(d_{i,j})$. We further assume that a static sensor can communicate with others within its transmission range with the same energy spent.

2.3 Energy Consumption Model

We assume that each static sensor generates one data packet per round to be transmitted to mobile sink or other sensors, and each packet has size k bits. Further, each static sensor s_i has a battery with finite, non-replenishable energy E_i. Now we discuss the energy consumption for data transmission and reception. Whenever a sensor transmits or receives a data packet it uses some energy from its battery. In this paper, we use the same radio model as discussed in [8] which is the first order radio model. In this model, a radio dissipates ϵ_{elec}=$50nJ/bit$ to run the transmitter or receiver circuitry and ϵ_{amp}=$100pJ/bit/m^2$ for the transmitter amplifier. Thus, the energy consumed by a static sensor s_i in receiving a k-bit data packet is given by $R_{X_i} = \epsilon_{elec} \times k$. The energy consumed in transmitting a data packet to static sensor s_j is given by, where $d_{i,j}$ is the distance between s_i, s_j. $T_{X_{i,j}} = \epsilon_{elec} \times k + \epsilon_{amp} \times F(d_{i,j})^2 \times k$. So, we denote $f_{i,j}$ as the total number of packets that node i (*e.g.* a static sensor) transmits to node j (*e.g.* a static sensor or mobile sink). The energy constraint at each sensor satisfies as follows, where $i = 1, 2, \cdots, n$,

$$\sum_{j=1}^{n+1} f_{i,j} \cdot T_{X_{i,j}} + \sum_{j=1}^{n} f_{j,i} \cdot R_{X_i} \leq E_i \tag{1}$$

2.4 Problem Formulation

With the objective of minimizing the energy cost of a mobile hybrid sensor network, we decompose this global optimization problem into two subproblems to be solved by mobile sink and static sensors.

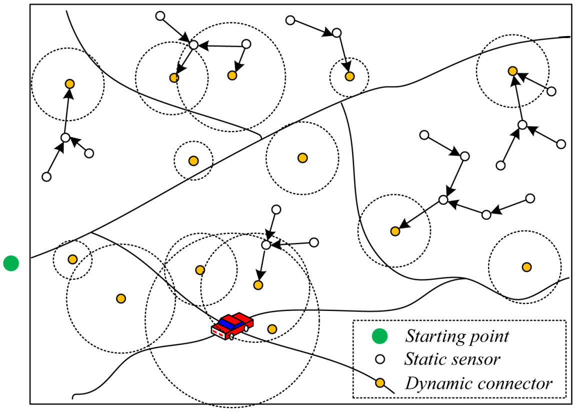


Fig. 2. Multi-hop data gathering paradigm

In one-hop structure, we focus on studying the primary energy cost which comes from the motion of mobile sink. Generally, some static sensors that are far away from the road segments, need transfer the data to certain sensors via multi-hop relay. As shown in Fig.2, these certain sensors can be regard as *dynamic connector*. While moving along the paths, mobile sink can collect data from dynamic connector instead of approaching each static sensor. In this way, the mobile sink can collect data from the static network even for a general case, which is more likely to happen in realistic sensing environment. Here, we specify a base station located outside the subregion as a *virtual sink* for data gathering. We define the lifetime T of network to be the number of rounds until the first static sensor runs out of its energy. A data gathering schedule specifies, for each round, how the data packets from all the static sensors are collected and transmitted to the virtual sink. Particularly, a schedule can be considered as a collection of T directed trees, each rooted at the virtual sink and spanning all the static sensors. Intuitively, in this situation, the network lifetime (*i.e.* energy consumption) is intrinsically connected to the data gathering schedule.

Then, we are ready to formulate the pending problem as follows.

Problem 1. Given n static sensors over a monitoring region Ω, a mobile sink m can move along the pre-given roads and directly gather data from nearby static sensors via one-hop communication, we want to design energy-efficient scheduling algorithms to minimize the data gathering energy cost in a mobile hybrid sensor network with the following two objectives:

(i) ***single-hop structure***: finding a movement scheduling of mobile sink that will minimize the energy cost.

(ii) ***multi-hop structure***: finding an energy-efficient manner in which data should be gathered from all the static sensors, such that the system lifetime is maximized.

3 Single-Hop Data Gathering Paradigm

In this section, we study on how to optimize the motion energy cost of mobile sink in single-hop data gathering paradigm. In the process of network formation, we assume that all the static sensors can reach its nearest path(s) at the minimum transmitter power level. As far as an individual static sensor is concerned, this radio model is also the most energy-efficient one.

Before a mobile sink starts a data gathering tour, the first issue to tackle is determining the anchor points need be visited. During data collection, it is unnecessary for mobile sink to approach the position of a static sensor but within its transmission range. Generally, a static sensor's transmission disk can intersect the pre-given roads with one or more intersection points, which can be defined as *anchor points*. Then when mobile sink moves to one of these anchor points, it can exchange data with certain static sensor and complete the data gathering. Firstly, for each static sensor, we calculate its corresponding anchor points. For the sake of discussion, we assume that the maximum of anchor points generated by the intersection a transmission disk with roads is a constant k. Specifically, we can regard a static sensor s_i as a group g_i, where $1 \leq i \leq n$, and thus $g_i = \{ap_1^{(i)}, ap_2^{(i)}, \cdots, ap_j^{(i)}, \cdots, ap_l^{(i)}\}$ $(l \leq k)$ which contains a subset of anchor points. Especially, considering the mobile sink starts from starting point and return this point after finishing data gathering task, we define the starting point as an extra group $g_0 = \{ap^{(0)}\}$. In this way, for n static sensors in the network, we can totally get $n+1$ groups contain the universe set of anchor points $\Upsilon = \sum_{i=0}^{n} g_i$ in the region. An example is shown in Fig.3.

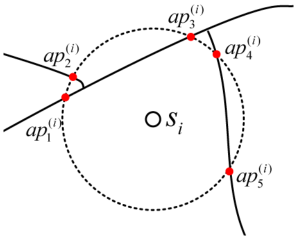


Fig. 3. An example of defining anchor points

A data gathering tour of the mobile sink consists of a subset of anchor points and road segments connecting them. For example, $\Upsilon = \sum_{i=0}^{n} g_i$ denotes a universe set of anchor points. Then, the data gathering tour of mobile sink can be represented by $P = \{ap^{(0)} \to ap_a^{(1)} \to ap_b^{(2)} \to \cdots \to ap_c^{(i)} \to \cdots ap_z^{(n)} \to ap^{(0)}\}$ $(a, b, c, z \in [1, k])$, which can be depicted in Fig.4. Thus, the optimal problem can be regarded as the problem of determining the subset of anchor points and the sequence to visit them with the constraint that the union of anchor points must cover the whole static sensors, so that the length of tour can be minimized. We further discuss the case when $R \to 0$, the mobile sink must visit the location of every static sensor one by one to gather data. In this situation, the pending problem is reduced to the well-known *Traveling Salesman Problem* (TSP) for discrete points in a plane [9]. The data gathering tour can be expressed by $P = \{m \to s_1 \to, s_2, \cdots, \to s_n \to m\}$.

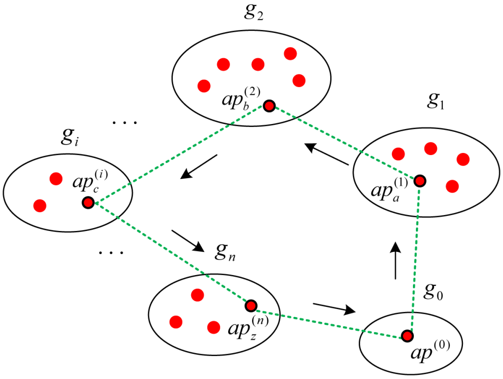


Fig. 4. Data gathering tour of a mobile sink

Our pending problem can be converted into the classic mathematic problem, *Group Steiner Tree* (GST), which has been introduced by *Reich* and *Widmayer* to solve wire routing with multiport terminals in physical VLSI design [10]. Group Steiner Tree problem is a generalization of the Steiner tree problem, and therefore NP-hard [11]. In fact, it is also a direct generalization of the even harder set-covering problem.

Specifically, we are given an undirected weighted graph G=(V,E) with the cost function c: $E \to \mathbb{R}_+$, and sets of vertices $g_1, g_2, \cdots, g_n \subset V$. We call $g_1, g_2, \cdots, g_n$ groups. The objective is to find the minimum cost subgraph G' of G that contains at least one vertex from each of the sets g_i. Formally, find $G' = (V', E')$ that minimizes $\sum_{e \in E'} c_e$, such that $V' \bigcap g_i \neq \emptyset$ for all $i \in \{1, 2, \cdots, n\}$. Considering that the staring point $ap^{(0)}$ is the only element of group g_0, it must be included in a subgraph G', which can ensure the assumption that mobile sink starts from *starting point*, traverses the monitoring area along road map, and then returns. Since G is connected, there exists an edge $e_{a,b}^{(i,j)}$ for each pair of vertices $ap_a^{(i)}$ and $ap_b^{(j)}$ (where $ap_a^{(i)}$, $ap_b^{(j)} \in V$ and $i \neq j$, $1 \leq a, b \leq k$), with weight $w_{a,b}^{(i,j)}$ defined as the shortest distance on the determinate roads for a mobile sink to move from $ap_a^{(i)}$ to $ap_b^{(j)}$. We can solve an all-pairs shortest-paths problem by running a single-source shortest-paths algorithm $|V|$ times, once for each vertice as the source. Since all edge weights are nonnegative, we can use *Dijkstras algorithm* whose time complexity is $O(|V|\log|V| + |E|)$. Then, we use a polynomial time algorithm proposed in [4] that with high probability finds a group steiner tree of cost within $O(\log k \log n{+}1)$ times the optimal solution, where k is the maximum size of a group and $n+1$ is the number of groups (here n denotes the number of sensors in a network). Compared to the heuristic solution presented in [2], we can provide a theoretical bound which proves that the total cost is within a constant factor of the optimum. Due to the limited space, the algorithm description is omitted.

4 Multi-hop Data Gathering Paradigm

In multi-hop data gathering paradigm which characterizes a more actual situation, we consider finding an efficient schedule X to gather and transmit the data from all the static sensors to the virtual sink, such that the energy consumption of a static network is minimized and thus its lifetime T is maximized. We define a set of *dynamic connector* as $C = \{c_1, c_2, \cdots, c_k\}$, including the static sensors nearby the paths and are likely to become dynamic connectors. In this paradigm, data generated from static sensors can be first delivered to the dynamic connectors, which are marked as orange points in Fig.2, stored there and then relayed to the mobile sink as it passes by. Finally, mobile sink returns and uploads the data to a base station (*i.e.* virtual sink).

The schedule X induces a flow network $G = (V, E)$. The flow network G is a directed graph having as nodes all the static sensors and the virtual sink, and having edges (i, j) with capacity $f_{i,j}$ where $f_{i,j} > 0$. Next, we consider the problem of finding a flow network G with maximum T (*i.e.* minimal energy usage will achieve the maximum lifetime). Clearly what needs to be found are the capacities of the edges of G. We call such a flow network G is a feasible flow network with lifetime T. A feasible flow network with the maximum lifetime is called an *optimal feasible flow network*. Now, the problem becomes a multi-source, multi-sink maximum flow problem We can reduce it to an ordinary maximum flow problem by that firstly adding a supersource s and an edge with infinite capacity from s to each of the multiple sources; then adding a supersink t (*i.e.* virtual sink) and an edge to each (c_i, t) pair and set $r_j^{(i)}$ ($j \in [1, m]$) as the weight of the edge from each of the multiple sink (*i.e.* dynamic connector) to t. $r_j^{(i)}$ denotes the minimum required transmission range of dynamic connector c_i to "touch" the road. Recall that a static sensor can communicate with sensors or mobile sink which are within its transmission range with the same energy consumption. Hence, we can still select a subset of anchor points generated from dynamic connectors to gather data by the scheduling algorithm proposed in Section 3, in order to achieve the minimal motion energy consumption of mobile sink.

In this way, our problem can be viewed as a *maximum flow* problem with energy constraints at the sensors, and can be solved by the following integer program with linear constraints. The integer program, in addition to the variables for the lifetime T and the edge capacities $f_{i,j}$, uses the following variables: for each static sensor $k = 1, 2, \cdots, n$, let $\pi_{i,j}^{(k)}$ be a flow variable indicating the flow that a sensor s_k sends to the virtual sink over the edge (i, j). The integer program is given by,

$$\max T \tag{2}$$

subject to the energy constraint (1) and the flow conservation constraints below, for each $k = 1, 2, \cdots, n$,

$$\begin{cases} a) & \sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{(n+1)} \pi_{i,j}^{(k)} \\ b) & T + \sum_{j=1}^{n} f_{i,j} = \sum_{j=1}^{n+1} f_{j,i} \\ c) & 0 \leq \pi_{i,j}^{(k)} \leq f_{i,j} \\ d) & \sum_{i=1}^{n} \pi_{i,n+1}^{(k)} = T \end{cases} \tag{3}$$

where all the variables are required to be non-negative integers. For each $k = 1, 2, \cdots, n$, constraint (3) consists of following restrictions: *a)* and *b)* enforce the flow conservation principle at a sensor; *c)* ensures that the capacity constraints on the edges of the flow network are respected and *d)* ensures that T flow from sensor s_k reaches the virtual sink. When all the variables are allowed to take fractional values, the linear relaxation of the above integer program can be computed in polynomial-time. First, we fix the values of the $f_{i,j}$ variables to the floor of their values from the linear relaxation, and then solve the linear program subject to constraints (1)-(3). The solution is guaranteed to have integer values for all the variables, since it is a *max-flow problem* with integer capacities.

Observe that this solution provides us readily with a schedule for collecting the data packets from all the static sensors, during the lifetime of the system. A simple way to construct such a schedule would be to take the flow network obtained from the solution, and push data packets from each static sensor on one or more paths (with available capacities) to the virtual sink. This approximate solution provides a near-optimal system lifetime that is efficiently computable.

5 Simulation and Numerical Results

In this section, we use real data traces gathered from GreenOrbs Testbed to evaluate the performance of the proposed solutions in terms of energy cost and lifetime. We select a rectangle region in GreenOrbs Testbed with size about $200m * 150m$. The road map for the selected region can be obtained by relevant topographic information and the real node locations of GreeOrbs Testbed are illustrated in Fig.5. Assume that the origin ($0m$, $0m$) is at the bottom left of the road map, and mobile sink starts from start point ($0m$, $30m$) and returns to the same point after each tour.

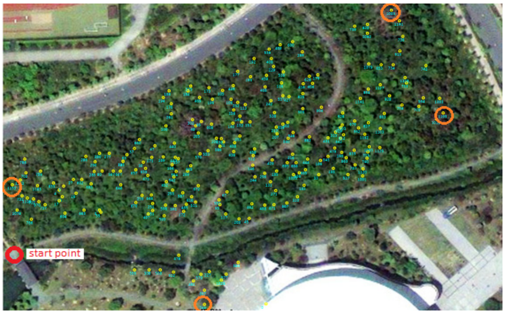


Fig. 5. In real GreenOrbs Testbed, 232 static sensors are deployed in a 200m*150m region. The 2D location coordinates of four boundary sensors are #1003 (8934.861,3103.606), #1153 (8763.517,3182.47), #744 (8850.326,3236.395) and #1193 (8958.139,3149.958), respectively.

5.1 Tour Length of Mobile Sink

We first evaluate the performance of tour length in one-hop structure. Assume that a static sensor has 3 options for transmission ranges: 20m, 40m and 60m in our experiments. The total length of paths that a mobile sink can walk in the road map is about 800m. Each result shown here is the statistical average of 20 experimental results. Fig.6 compares the shortest tour length calculated by our movement scheduling algorithm with the total length of pre-defined paths from different number of static sensors (n). We can get that with the increase of the number of static sensors, the relative tour ratio will increase exponentially. Generally, the value of relative tour ratio might be greater than 1. When n=40 and 100, the relative tour ratio is about 1.18 and 1.35 in our specific experiment scenario. It is mainly because the shortest tour length correlates closely to the structure of road map, *e.g.* shape and length. To return the start point after gathering data from all the static sensors, the mobile sink may traverse a certain road segment twice.

By scheduling the working/sleeping mode of static sensors, we focus on the number of static sensors (n) and evaluate its effect on performance of tour length. We compare our solution with the optimal solution and spanning tree covering

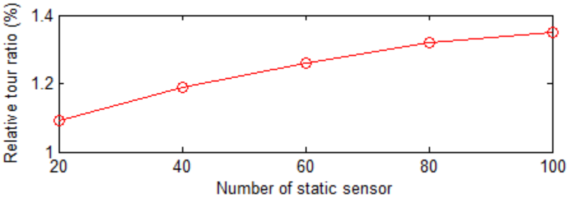
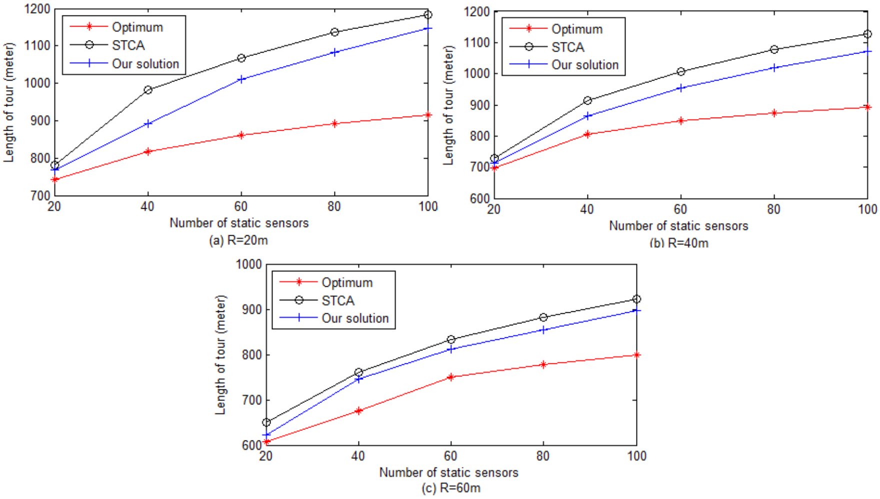


Fig. 6. Comparison between the shortest tour length calculated by our movement scheduling algorithm and the total length of pre-defined paths from different n

algorithm (STCA for short) proposed in [2]. Recall that each static sensor can reach the nearest path using the minimum radio power level. Here, we assume that the discrete transmission range of a static sensor is varied to be $20m$, $40m$ and $60m$, and the number of active static sensors n in the network is changed from 20 to 100 with an increment of 20.



(a) R=20m

(b) R=40m

(c) R=60m

Fig. 7. Comparison the length of data gathering tour with the optimal solution and STCA with different transmission ranges

From the curves in Fig.7, we can observe that when the number of active static sensors is small (*e.g.* n=20), and the transmission range is relatively short compared to the average distance between a sensor and its nearest neighbor, both our solution and STCA have very close to the optimal solution. With the increase of n, the length of tour will increase continuously. It is unnecessary for mobile sink to arrive at the nearest location of each static sensor but within its transmission range, so our solution performs better than STCA. As n increases (directly leads to an increase in the number of anchor points), the computational complexity of our solution will become not ideal. Analyzing the data shown in Fig.8, with the fix of n, the length of tour decreases gradually as the transmission range R increases. For example, when n=60, R=$20m$ and $60m$, the difference between the shortest tour length is almost $200m$. It should be pointed out that motion energy cost will reduce at the expense of the increase of data transmission energy cost. We need balance the two parts of energy cost to achieve the minimal energy cost of a hybrid sensor network.

5.2 Lifetime of Static Network

We evaluate the performance of the scheduling algorithm in terms of the network lifetime in multi-hop data gathering paradigm. Assume that each static sensor has an initial energy of $1J$ and the base station and generates data packets with the size of 800 bits. To construct a network Graph G' defined in Section 3, we calculate the weight of each edge in G as a function of the shortest moving distance along the pre-given road map. We compare our data gathering scheduling algorithm with that obtained from a chainbased hierarchical protocol (*LRS* for short) proposed in [13]. Recall that the (integral) solution given by our algorithm is an approximation of the optimal (fractional) solution. As shown in Fig.8, the lifetime of the schedule given by our algorithms always significantly outperforms that given by the *LRS* protocol. In case of data collection, our algorithm performs 1.98 to 2.11 times better than the *LRS* protocol in terms of system lifetime.

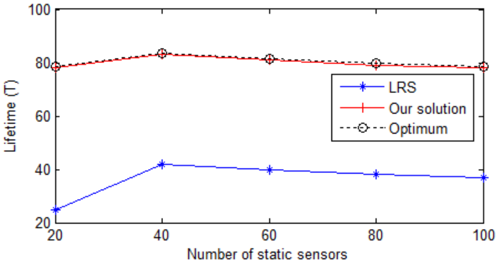


Fig. 8. Data gathering

6 Related Work

Recently, mobility in sensor networks has received much attention [1] [2] [4] [5] [14] [15] [16] [17] [19] [20]. In the context of network with mobile sink, most existing works investigate how to plan the moving trajectory for the mobile sink to achieve an efficient data collection. In [15], a number of mobile collectors, called data mules, traverse the sensing field along parallel straight lines and gather data from sensors. This scheme works well in a large scale, uniformly distributed or randomly distributed sensor network. However, in practice, data mules may not always be able to move along straight lines. In [5] employed a single mobile collector called SenCar, and focused on optimizing the data gathering tour. Considering that utilizing only a single SenCar may lead to a long data gathering cycle and data buffer overflow at sensors, the author of [16] explored data gathering with multiple SenCars and SDMA technique. *Ma et al.* [2] proposed a data gathering mechanism with multiple mobile collectors, in which the mobile data collectors will collect data from all sensors through single-hop communications. In some existing works, data gathering with controlled sink movement is to optimize the network performance such as energy consumption [17] [18], network lifetime [19] [20], or detection delay [1] [3], controlled sink mobility are considered such that a sink moves along predetermined optimized trajectory, or paths meeting certain criteria. *Luo et al.* [18] investigated how to minimize energy consumption in a wireless sensor network with a mobile sink while also considering multi-hop propagation effects. Using analytical means they estimated the optimal trajectory of the mobile sink to be a cycle and calculate its optimal positioning and radius. The author of [1] proposed a number of deployment strategies for the static sensor network and mobile sensor network respectively in order to satisfy certain reaction delay requirement while minimizing the total cost.

However, those works mentioned above assumed that mobile collector can move randomly in the monitoring area, which is not practical. In this paper, we explore the restricted sink mobility in constrained moving region for data gathering to handle the natural obstacles problem in reality.

7 Conclusion

In this paper, we investigate low cost data gathering algorithm for hybrid sensor networks by introducing mobile sink into the network. Firstly, we design a movement scheduling algorithm so that the energy cost of mobile sink will be minimized in single-hop structure. By using Group Steiner Tree method, we can provide a theoretical performance guarantee. Further, we formulate the mobile data gathering problem with integer program in multi-hop structure, which is near optimal and can be computed in polynomial time. The experimental data obtained from practical GreenOrbs Testbed demonstrate the effectiveness of our solutions.

Acknowledgement. We wish to thank Prof. Xiangyang Li and Dr. Xufei Mao for their help with practical suggestions. The research of authors are partially supported by the National 973 Project of China under Grant No.2011CB302701, the National Science Fund for Distinguished Young Scholars under Grant No. 60925010, the National Natural Science Foundation of China under Grant No. 60833009, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No.20100009120004, No.20110009110001, the Fundamental Research Funds for the Central Universities under Grant No.2011JBM011, and the open Projects of Beijing Key Lab of Intelligent Telecomm. Software and Multimedia.

References

1. Tang, S., Li, X., Yuan, J., Wang, C., Chen, G., Dai, G.: DREAM: On the Reaction Delay in Large Scale Wireless Networks with Wireless Networks with Mobile Sensors. In: IEEE IWQoS, pp. 1–9 (2010)
2. Ma, M., Yang, Y.: Data gathering in wireless sensor networks with mobile collectors. In: IPDPS, pp. 1–9 (2008)
3. Xu, X., Luo, J., Zhang, Q.: Delay tolerant event collection in sensor networks with mobile sink. In: Proceedings of IEEE Infocom (2010)

4. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., Rubenstein, D.: Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In: Architectural Support for Programming Languages and Operating Systems (2002)
5. Ma, M., Yang, Y.: SenCar: An energy efficient data gathering mechanism for large scale multihop sensor networks. IEEE Transactions on Parallel and Distributed Systems 18(10) (2007)
6. Mo, L., He, Y., Liu, Y., Zhao, J., Tang, S., Li, X., Dai, G.: Canopy closure estimates with GreenOrbs: long-term large-scale sensing in the forest. In: ACM SenSys (2009)
7. Mallinson, M., Drane, P., Hussain, S.: Discrete radio power level consumption model in wireless sensor networks. In: Second International Workshop on Information Fusion and Dissemination in Wireless Sensor Networks (2007)
8. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy efficient Communication Protocols for Wireless Microsensor Networks. In: HICSS (2000)
9. Skiena, S.S.: Traveling salesman problem. In: The Algorithm Design Manual, pp. 319–322. Springer, New York (1997)
10. Reich, G., Widmayer, P.: Beyond Steiner's Problem: A VLSI Oriented Generalization. In: Nagl, M. (ed.) WG 1989. LNCS, vol. 411, pp. 196–210. Springer, Heidelberg (1990)
11. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the Group Steiner Tree problem. In: Proceedings of SODA, pp. 253–259 (1998)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms, 2nd edn. The MIT Press (2001)
13. Lindsey, S., Raghavendra, C., Sivalingam, K.: Data gathering in sensor networks using the energy-delay metric. In: Proc. of IPDPS Workshop on Issues in Wireless Networks and Mobile Computing (2001)
14. Tang, S., Yuan, J., Li, X., Liu, Y., Chen, G., Gu, M., Zhao, J., Dai, G.: DAWN: Energy efficient data aggregation in WSN with mobile sinks. In: The 18th International Workshop on Quality of Service (IWQoS), pp. 1–9 (2010)
15. Jea, D., Somasundara, A., Srivastava, M.: Multiple controlled mobile elements (data mules) for data collection in Sensor Networks. In: 2005 IEEE/ACM International Conference on Distributed Computing in Sensor Systems (2005)
16. Zhao, M., Yang, Y.: Data gathering in wireless sensor networks with multiple mobile collectors and SDMA technique sensor networks. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2010)
17. Xing, G., Wang, J., Shen, K., Huang, Q., Jia, X., So, H.C.: Mobility-assisted spatiotemporal detection in wireless sensor networks. In: ICDCS (2008)
18. Luo, J., Hubaux, J.P.: Joint mobility and routing for lifetime elongation in wireless sensor networks. In: Proc. IEEE INFOCOM (2005)
19. Basagni, S., Carosi, A., Melachrinoudis, E., Petrioli, C., Wang, Z.M.: Controlled sink mobility for prolonging wireless sensor networks lifetime. Wireless Networks 14(6), 831–858 (2008)
20. Banerjee, T., Xie, B., Jun, J., Agrawal, D.: Increasing lifetime of wireless sensor networks using controllable mobile cluster heads. Wireless Communications and Mobile Computing 10(3), 313–336 (2009)

Debugging the Internet of Things: A 6LoWPAN/CoAP Testbed Infrastructure

Daniel Bimschas, Oliver Kleine, and Dennis Pfisterer

Institute of Telematics, University of Lübeck
Ratzeburger Allee 160, 23538 Lübeck, Germany
{bimschas,kleine,pfisterer}@itm.uni-luebeck.de

Abstract. This paper is based on two fundamental assumptions about a future Internet of Things (IoT): i) The amount of wireless, resource-constrained devices will outnumber the amount of devices in the current internet by several orders of magnitude and ii) those devices will be connected to the Internet over multi-hop wireless links. We argue that the experimental validation in testbeds is imperative to make those networks robust. However, there are only limited means to support researchers in "debugging" the actual communication on the wireless medium and often developers can only guess why their protocols don't work in a given environment. In this paper, we present such a framework which extends the WISEBED testbed federation. Our contribution allows an easy-to-use browser-based experimentation and evaluation of wireless multi-hop protocols in all WISEBED-compatible testbeds (nine testbeds with 1000 sensor nodes and the SmartSantander [17] smart city testbed which will offer up to 20, 000 IoT devices). Using a generic packet tracking framework for multiple platforms, researchers can easily detect hotspots and bottlenecks in the network and follow the routes of individual packets as they are forwarded. Experiment configurations can be shared on the web so that experiments can easily be repeated to verify published results. We demonstrate the usability of our approach by means of a real-world use-case.

Keywords: Experimentally-driven Research, Testbeds, Internet of Things, 6LoWPAN, CoAP.

1 Introduction

This paper is based on two fundamental assumptions about a future Internet of Things (IoT): i) The amount of wireless, resource-constrained devices will outnumber the amount of devices in the current internet by several orders of magnitude and ii) those devices will be connected to the Internet over

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 207–220, 2012.

multi-hop wireless links using standardized protocols such as 6LoWPAN[1] and the Constrained Application Protocol (CoAP)[2].

The integration of resources and services available in the traditional Internet with novel real-world services offered by IoT devices such as sensor nodes, gives rise to a completely new class of applications. However, the development of applications exploiting this combined infrastructure is cumbersome and difficult. This is due to the massively distributed nature of these networks combined with the heterogeneity of the devices, severe resource-constraints, and the difficulties of wireless multi-hop networking. In the past, simulations were a predominant means to test and optimize networking protocols. However, especially in wireless networking environments, many of these simulations lack realism and results strongly depend on the actual parameters. Apart from the fact that these were often not even reproducible [11,20], the simulated environments and their implicit assumptions are also hardly comparable to any real-world setting.

As a result, IoT-testbeds such as MoteLab [19], Kansei [1], w-iLab.t Testbed [3], and WISEBED [4,5] have become an important means to improve this situation by allowing researchers to evaluate the performance of protocols and applications [8] in real-world environments. This so-called experimentally-driven research has been instrumental in designing protocols that work efficiently in real-world settings. In the past virtually all experiments have been limited to the wireless network and there has been no or only very limited interaction with the Internet. If our two assumptions hold, a novel kind of testbeds is required that allows conducting experiments to exploit the merged infrastructure of the Internet and the Internet of Things. In this paper, we present

1. an extension to IoT testbeds to conduct experiments using standard Internet technologies such as IP and HTTP,
2. a framework for tracking packets in the wireless network, and
3. a tool for controlling, managing, and evaluating experiments using JavaScript only.

For our extension, we have chosen the WISEBED testbed federation, which, in contrast to virtually any other testbed, offers a well-defined web service API to allow for such extensions. Our contribution allows an easy-to-use browser-based experimentation and evaluation of wireless multi-hop protocols in all WISEBED-compatible testbeds (nine testbeds with 1000 sensor nodes and the SmartSantander [17] smart city testbed which will offer up to 20,000 IoT devices). Using a generic packet tracking framework for multiple platforms, researchers can easily detect hotspots and bottlenecks in the network and follow the routes of individual packets as they are forwarded. In addition to this, we present a reverse proxy that allows researchers to test the integration of WSNs into the Internet.

[1] 6LoWPAN [9] is a lightweight IPv6 adaptation layer allowing resource-constrained sensors to exchange IPv6 packets with the Internet.

[2] CoAP [15] is a draft by IETF's CoRE working group and provides a lightweight alternative to HTTP using a binary representation and a subset of HTTP's methods (GET, PUT, POST, and DELETE).

The remainder of this paper is structured as follows. Section 2 introduces our framework and discusses the extension to the WISEBED testbed federation, the reverse proxy architecture, and the packet tracking infrastructure. Section 3 presents a use-case in optimizing a 6LoWPAN- and CoAP-based application using our approach. Finally, Section 4 concludes this paper with a summary.

2 Architecture

Figure 1 depicts the extension to the WISEBED testbed federation (described in detail in Section 2.1), the reverse proxy architecture (Section 2.2) and their relationship with the overall testbed architecture. Section 2.3 discusses how packet tracking can be used on this architecture to help optimizing network protocols.

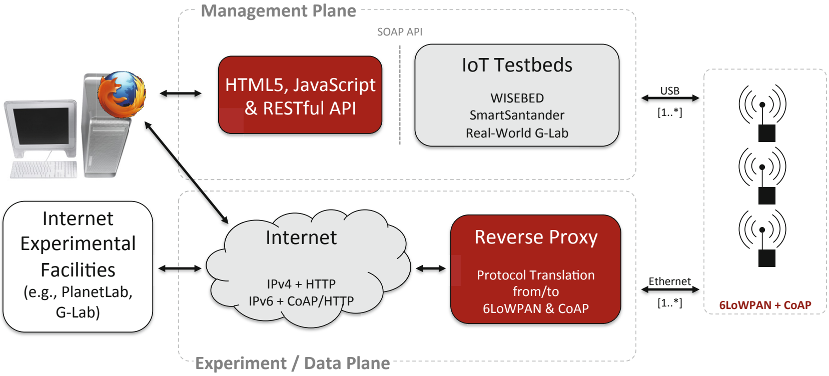


Fig. 1. Framework components overview

2.1 Testbed Extension

A variety of different testbeds for wireless sensor networks has been developed in the past that are now available to researchers. Each testbed has different characteristics and specific benefits [8]. However, the majority of them are also hardy extensible due to a proprietary user interface. These are often web-based and can only be operated by humans which prevents automated experimentation or extensions. An exception to this is the WISEBED [4,5] testbed federation, which has defined a set of SOAP-based web service APIs that separate the interface of a testbed from the actual backend implementation and allows the implementation of generic clients and tools.

Consequently, we have chosen to build upon the WISEBED APIs to implement our extension. As a result, our extension is compatible with all WISEBED-compatible testbeds where currently several thousand sensor nodes from more than 10 different vendors are available. These APIs allow creating web-, console- and desktop-based clients with the ability to exchange messages with sensor nodes via their serial interface (e.g., to parameterize nodes to explore the parameter space in an experiment) as well as to reset crashed nodes, reprogram nodes, send control commands, etc. This allows a very high degree of interactivity with the experiment in which experimenters can control the experiment at runtime. For an in-depth overview, we refer the reader to the web page of the WISEBED project (http://wisebed.eu).

The goal of our extension is twofold: On the one hand, we provide a web-friendly interface to testbeds based on HTTP web services instead of SOAP-based ones to allow a broader acceptance. On the other hand, the goal was to build a modern web-based user interface that support browser-based experimentation (e.g., for packet tracking as discussed in Section 2.3). The motivation for this extension is the observation, that only a very limited subset of users was actually creating custom experimentation clients based on the SOAP API but instead used the simple ones shipped by the WISEBED consortium.

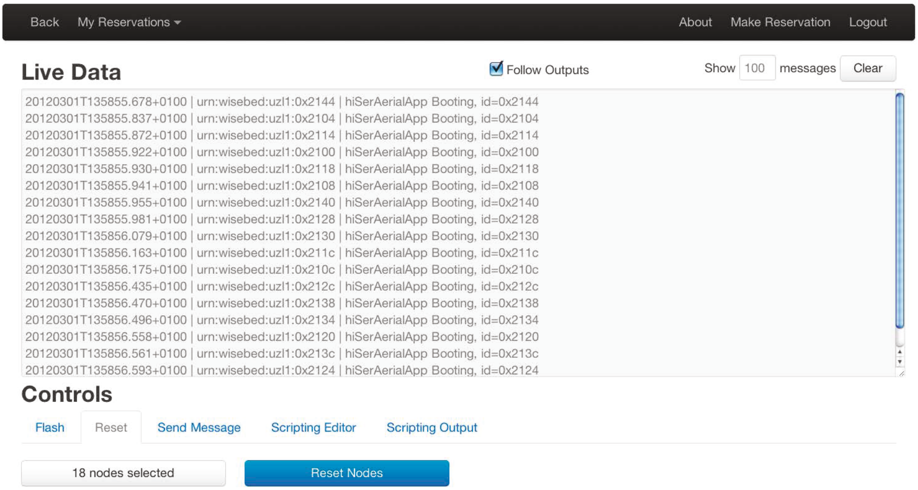
Additionally, the support for SOAP-based web services in languages such as JavaScript, Ruby, Python or others is limited and they often lack support for the WS-I web service interoperability standards. However, all these languages excel in their support for HTTP-based web services, also known as RESTful HTTP web services [7]. The extension presented here is based on easy-to-use technologies such as JavaScript, AJAX and JSON that allow users to create browser-based experiments with a very flat learning curve. In the remainder of this section, we now first introduce the RESTful HTTP-based web service API and then present the architecture of the new web-based user interface that uses this API and enables browser-based scriptable experimentation control. An example for such a scriptable execution is packet tracking to debug wireless communication as discussed in Section 2.3.

RESTful HTTP-Based Web Service API. The new HTTP-based web service API adheres to the REST architecture style and provides the same functionality as the SOAP-based APIs. Scripting experiments is now possible using virtually any interpreted or compiled programming language that supports HTTP and JSON (JavaScript Object Notation), thereby overcoming the issues with the SOAP-based APIs. Operations such as authentication, reservation, reprogramming, and resetting nodes are done by sending simple HTTP GET, PUT, POST and DELETE requests. Data exchange with sensor nodes is realized via *WebSockets* that allow bidirectional socket-like data exchange on top of, e.g., HTTP and avoid the problems of polling in AJAX-based web applications.

Both, the reference implementation and the documentation of the RESTful APIs are available as open-source projects at https://github.com/wisebed/rest-ws. The component is designed as a proxy mediating between the HTTP-based API and the SOAP-based APIs and is hence able to serve *all* WISEBED testbeds.

Web-Based User Interface Supporting Experiment Scripting. In addition to the HTTP-based API we developed a graphical user interface called

WiseGui that is solely based on cutting-edge HTML5 technologies. This includes new JavaScript and CSS features that are being subsumed under this standard which is already supported by all major browsers. The WiseGui already includes all required features for conducting experiments such as signing up, logging in, making reservations, "connecting" to a reservation, programming and resetting nodes, sending messages to nodes, and displaying messages from nodes in a terminal-like window (cf. Figure 2). WiseGui is available for usage by everyone at http://wisebed.itm.uni-luebeck.de but may also be deployed on local testbed installations or embedded into other web sites.

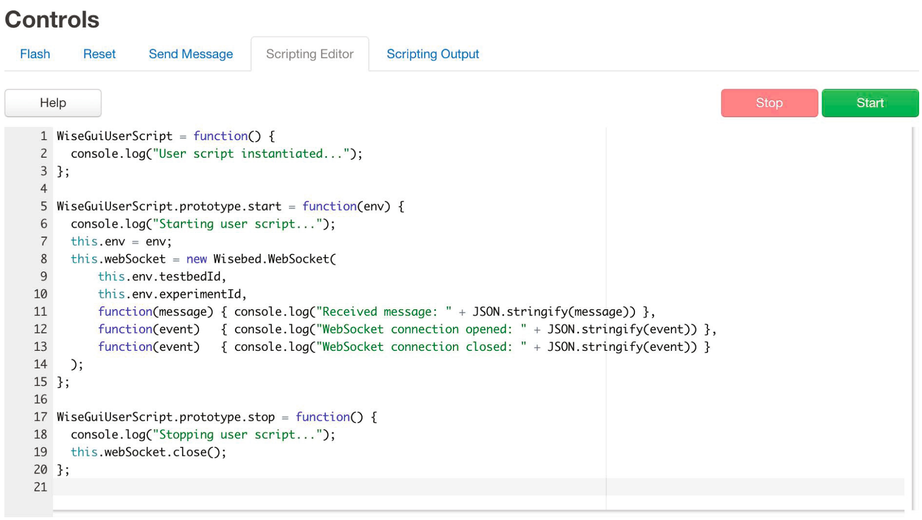


```
20120301T135855.678+0100 | urn:wisebed:uzl1:0x2144 | hiSerAerialApp Booting, id=0x2144
20120301T135855.837+0100 | urn:wisebed:uzl1:0x2104 | hiSerAerialApp Booting, id=0x2104
20120301T135855.872+0100 | urn:wisebed:uzl1:0x2114 | hiSerAerialApp Booting, id=0x2114
20120301T135855.922+0100 | urn:wisebed:uzl1:0x2100 | hiSerAerialApp Booting, id=0x2100
20120301T135855.930+0100 | urn:wisebed:uzl1:0x2118 | hiSerAerialApp Booting, id=0x2118
20120301T135855.941+0100 | urn:wisebed:uzl1:0x2108 | hiSerAerialApp Booting, id=0x2108
20120301T135855.955+0100 | urn:wisebed:uzl1:0x2140 | hiSerAerialApp Booting, id=0x2140
20120301T135855.981+0100 | urn:wisebed:uzl1:0x2128 | hiSerAerialApp Booting, id=0x2128
20120301T135856.079+0100 | urn:wisebed:uzl1:0x2130 | hiSerAerialApp Booting, id=0x2130
20120301T135856.163+0100 | urn:wisebed:uzl1:0x211c | hiSerAerialApp Booting, id=0x211c
20120301T135856.175+0100 | urn:wisebed:uzl1:0x210c | hiSerAerialApp Booting, id=0x210c
20120301T135856.435+0100 | urn:wisebed:uzl1:0x212c | hiSerAerialApp Booting, id=0x212c
20120301T135856.470+0100 | urn:wisebed:uzl1:0x2138 | hiSerAerialApp Booting, id=0x2138
20120301T135856.496+0100 | urn:wisebed:uzl1:0x2134 | hiSerAerialApp Booting, id=0x2134
20120301T135856.558+0100 | urn:wisebed:uzl1:0x2120 | hiSerAerialApp Booting, id=0x2120
20120301T135856.561+0100 | urn:wisebed:uzl1:0x213c | hiSerAerialApp Booting, id=0x213c
20120301T135856.593+0100 | urn:wisebed:uzl1:0x2124 | hiSerAerialApp Booting, id=0x2124
```

Fig. 2. WiseGui screenshot showing streamed output from the nodes' serial interface

Apart from basic experimentation support, WiseGui has the outstanding feature to allow experimenters to script experiments directly in the browser by writing JavaScript in an embedded editor (cf. Figure 3). This completely eliminates the need for any client-side software installation. Scripts are executed right in the same browser window as WiseGui and can react to incoming messages from nodes, can generate messages to be sent to nodes, or can invoke any of the aforementioned functionalities to control the experiment (e.g., reprogram nodes, send configuration parameters, or reset nodes). Furthermore, scripts may modify the Document Object Model (DOM) of the web page. This for instance allows creating live visualizations of experiment data or updating tables displaying aggregated data. A use-case for this feature is introduced in Section 3.

A major issue with experimental facilities is repeatability of experiments by other researchers, e.g., to verify results published in papers. To facilitate such endeavors, the WiseGui enables "importing" of experiment configurations from any web page. Such a configuration contains a description of which nodes are



```
WiseGuiUserScript = function() {
  console.log("User script instantiated...");
};

WiseGuiUserScript.prototype.start = function(env) {
  console.log("Starting user script...");
  this.env = env;
  this.webSocket = new Wisebed.WebSocket(
      this.env.testbedId,
      this.env.experimentId,
      function(message) { console.log("Received message: " + JSON.stringify(message)) },
      function(event)   { console.log("WebSocket connection opened: " + JSON.stringify(event)) },
      function(event)   { console.log("WebSocket connection closed: " + JSON.stringify(event)) }
  );
};

WiseGuiUserScript.prototype.stop = function() {
  console.log("Stopping user script...");
  this.webSocket.close();
};

```

Fig. 3. WiseGui script editor

to be programmed with which binary image. Figure 4 shows an example defining two sets of nodes to be programmed with different binary images. The first set is determined by the URL query string `?capability=pir&filter=0x21` and consists of all nodes that have a passive-infrared sensor (PIR) and contain the string `0x21` in their description. The second set is defined likewise for temperature sensors. Upon passing the URL of this configuration to the WiseGui it will first resolve the sets of nodes, then download the binary images from the URL defined by the (relative) path in `binaryProgramUrl` and finally flash the individual node sets with the correct image file.

A future addition will be to include a URL to an evaluation script. This allows researchers to include links in publications pointing to the configurations that were used to generate the published results. As such a configuration includes the binary program images, the mapping of programs to nodes, and the evaluation script other researchers can repeat each experiment by just pasting this link into the WiseGui.

2.2 Reverse Proxy

As mentioned in the introduction, resource-constrained IoT devices are typically not capable of providing direct IP connectivity or even web services via HTTP. Because of this, an IPv6 adaption layer (6LoWPAN [9]) and a lightweight alternative to HTTP (CoAP [15]) have been developed. For integration into the Internet, a (transparent) protocol conversion is required. In the context of the EU-project SPITFIRE, we have developed such a reverse proxy architecture [12], which has been extended for the work presented here. This so-called Smart Service Proxy (SSP) acts as a border device between the Internet and an IoT

```
{ "configurations" : [
    {
      "nodeUrnsJsonFileUrl" : "http://wisebed.itm.uni-luebeck.de/rest/2.3/uzl/experiments/nodes?capability=pir&filter=0x21",
      "binaryProgramUrl"    : "bin/JN5148/Nodes.bin"
    }, {
      "nodeUrnsJsonFileUrl" : "http://wisebed.itm.uni-luebeck.de/rest/2.3/uzl/experiments/nodes?capability=temperature&filter=0x21",
      "binaryProgramUrl"    : "bin/JN5148/Gateway.bin"
    }
  ]
}
```

Fig. 4. Experiment configuration

network (e.g., a sensor network) as illustrated in Figure 1. The SSP provides a transparent protocol translation on different layers: on the network layer, it converts IPv6 to 6LoWPAN and on the transport and application layer, it converts all three protocol combinations {HTTP, TCP, IPv4}, {HTTP ,TCP, IPv6}, and {CoAP, UDP, IPv6} to {CoAP, UDP, 6LoWPAN}.

The core of the SSP is based on Netty [10], a Java framework for asynchronous network I/O. Internally the SSP is organized into modules with different responsibilities. These include protocol conversions, mime-type conversions, caching, compression, and more. These modules are arranged as a stack and protocol packets are passed from bottom to top through the modules for analysis, processing and response creation and back from top to bottom afterwards.

The topmost module is called backend because it either provides the requested data itself or obtains the data from a network behind. In our case it is responsible for application layer protocol translation from HTTP to CoAP and vice versa and forwarding the translated request to the sensor node, resp. the translated response to the Internet client.

The network-layer conversion allows a direct integration of IoT devices into the Internet and enables a seamless exchange of IP packets. The application-layer conversion provides integration on a service-level, i.e., integration into the World Wide Web. The service-level integration offered by the SSP is agnostic of the actual version of IP and accepts HTTP requests over IPv4 and IPv6. This is of special importance as IPv4 is still the predominant protocol on the Internet.

Regarding the data link layer, the SSP provides Ethernet on the Internet side and a 802.15.4 radio on the WSN side. Since data link layer protocols may vary on the route between client and server this is not exactly a protocol conversion but is mentioned for the sake of completeness. Note, that the IPv6/6LoWPAN conversion is not part of the publicly available source code at https://github.com/ict-spitfire/smart-service-proxy. Actually, this is realized using an external box having two data link layer interfaces. Its ethernet

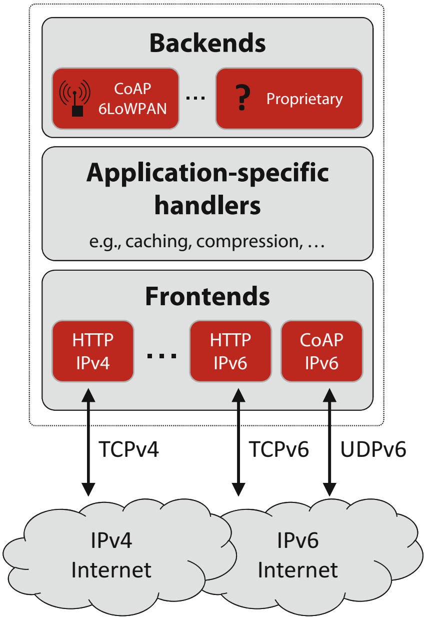


Fig. 5. SSP's protocol conversion architecture

interface is connected to the SSP whereas its 802.15.4 radio interface is the default gateway for the WSN built up of sensor nodes. The network connection between both devices is realized using a dedicated IPv6 net. From routing perspective the SSP acts as default gateway for the external box whereas the external box acts as default gateway for the WSN. By this means, we have three nets involved. The Internet (connected to the SSPs eth0), the net 2001:638:70a:c002::/64) between the SSP (eth1) and the external box (eth) and 2001:638:70a:c005::/64 for the WSN. However, for the sake of understandability the combination of both devices is refered to as SSP in the following.

Network- and Transport-level Conversion. Converting IPv6 to 6LoWPAN is a straightforward thing to do and thus not further introduced here. However, on transport level, things are more complex since sensor nodes are too resource-constrained to support TCP [14]. Consequently, the SSP intercepts TCP connection attempts and accepts them on behalf of the sensor node. From a client's perspective the TCP connection appears to be end-to-end to the sensor node.

Assume a sensor node with the IPv6 address 2001:638:70a:c005::2 running a CoAP web server on top of UDP. The gateway to the Internet is 2001:638:70a:c005::1 (the IPv6 address of the SSPs 802.15.4 interface). Furthermore let's assume an Internet client using a web browser with HTTP over TCP.

The browser tries to establish an end-to-end TCP connection with the sensor node. This connection attempt is intercepted by the SSP as illustrated in Figure 6. Each arrow represents the transfer of a TCP packet, either between

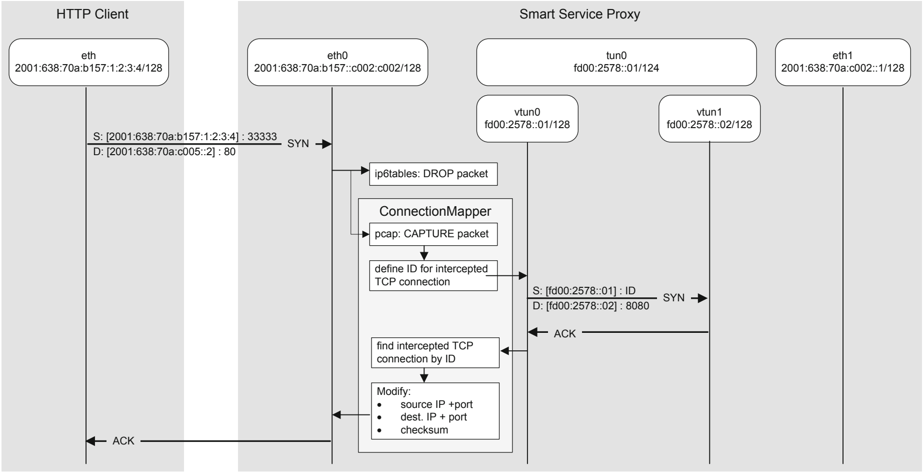


Fig. 6. Intercepting a TCP connection attempt

two sockets or with method calls within the application. The source port 33333 is just an example but in fact randomly chosen by the HTTP client. The TCP connection attempt starts with a SYN message from the client.

The original packet is dropped using the following ip6tables Linux firewall rule: `ip6tables -A FORWARD -i eth0 -j DROP`. PCAP (see [18] for Linux and [13] for Windows OS) captures the packet before being dropped and passes it to the so-called *Connection Mapper*, which is responsible for the TCP interception handling. The *Connection Mapper* is a software component of the SSP that uses jNetPcap [16] to access the operating system-specific libpcap library [18]. Each TCP connection is uniquely identifiable by the quadruple {source IP, source port, destination IP, destination port}. The Connection Mapper memorizes the quadrupel of the captured TCP connection and assigns an ID (range 1 to 65535) to the quadrupel.

In order to establish a virtual TCP connection without the need for reimplementing TCP, virtual interfaces are used. For this, the SSP creates a virtual IP interface (tunX) assigned with a /124 IPv6 network from the unique local address space. After assigning the original TCP connection attempt (SYN from Internet client) with an ID, the Connection Mapper uses this ID as source port to establish a local TCP connection between two virtual tun interfaces (vtun0 and vtun1 in Figure 6). Since the port equals the ID, the Connection Mapper can always map the ID to the quadruple.

Service-level Conversion. A client in the Internet has several options to contact a CoAP-based RESTful web service [7] offered by a sensor node using either {HTTP, TCP, IPv4}, {HTTP ,TCP, IPv6}, or {CoAP, UDP, IPv6}. The first and the second require a conversion from HTTP to CoAP. In addition, TCP must be ”split up” to UDP (port addressing) and CoAP (reliability) and for

IPv4 also a conversion to IPv6 is required. In the following, we explain these conversions in detail.

To allow an IPv4-based access to CoAP/6LoWPAN-based services, a mapping between IPv4-based HTTP URLs (e.g., http://127.0.0.1/some/service) and IPv6-based CoAP/6LoWPAN-based URLs (e.g., coap://[2001:638::CDEF]/some/service) is required. Our approach is based on a wildcard DNS entry that maps a sub-domain to a single IPv4 address. We resolve *.coap2.wisebed.itm.uni-luebeck.de to the IPv4 address of the SSP. The IPv6 address of a sensor is encoded in the hostname of the sub-domain by replacing each colon with a minus.

A client would then set the *Host:* field of the HTTP request to http://2001-638–CDEF.coap2.wisebed.itm.uni-luebeck.de and the SSP uses this field to extract the IPv6 address of the sensor node. Converting HTTP to CoAP is done as follows: The SSP translates the HTTP requests to CoAP requests, forwards them to the corresponding sensor nodes, waits for the CoAP response, translates it into a HTTP response, and sends the response to the client.

As described in [15] the protocol translation regarding the mapping between HTTP header fields and CoAP options is quite straightforward and thus not further introduced here. The main pitfall is the underlying transport protocol which is the connection based TCP for HTTP and the connectionless UDP for CoAP. CoAP provides optional reliability and by this means provides a core functionality of TCP based on UDP. Thus, the SSP must keep the intercepted TCP connection and disperse its functionality on UDP and CoAP. CoAPs reliability bases on acknowledgements (ACKs) to be sent in answer to a received message. The ACK message can either contain a response code and payload (a representation of the requested resource) or be empty, indicating that the server is willing to answer the request but needs more time to do so. However, if there was no non-empty CoAP response within a predefined period of time, the SSP sends a HTTP time-out message to the Internet client and a RST message to the CoAP server to cut the request processing of.

2.3 Packet Tracking

The development of applications for wireless sensor networks is cumbersome and error-prone, in particular due to the faulty wireless communication channels they typically use. This is especially painful when experimenting with new or evolving routing schemes in multi-hop environments. Often experimenters observe that packets sent over multiple hops get lost but the exact reason is unknown. Experienced researchers may guess the correct reason (e.g., the wireless channel is occupied due to excessive flooding or the like) and may even be capable to solve some of the causes. However, virtually always, developers must resort to ”println()-debugging” to be able to track down the root cause of the problem by manually analyzing debugging data received via the serial interfaces.

This situation is aggravated by the fact that some problems only occur in large-scale networks [6], which basically eliminates debugging on the developer's desk. In such situations, experimental facilities are helpful that allow large-scale experimentation. However, while supporting large-scale experiments, there is virtually no support beyond println()-debugging.

We propose a generic mechanism for tracking packets in a testbed to analyze where hotspots are, where the medium is heavily loaded or to track individual packets as they are forwarded in the network. Our approach is based on a contract between the wireless sensor nodes and an evaluation script. The nodes forward any activity on the wireless medium to the serial interface (i.e., received and transmitted packets). An evaluation script running in the WiseGui (cf. Section 2.1) will receive all data from all nodes in a testbed and can then run custom evaluations.

We have already implemented support for this kind of debugging in the iSense operating system and will be finalizing an implementation for the Wiselib [2] template library soon, which adds support for TinyOS, Contiki, and Scatterweb2 to name only a few. The packet format that nodes use on their serial port for this purpose is depicted in Figure 7. `TYPE` indicates the type of the packet (e.g. an IPv6 packet), `DIRECTION` indicates if a packet was received or sent, `SRC_MAC` and `DST_MAC` contains the 64-bit MAC addresses of sender and recipient of the packet and `PACKET` contains the packet itself. The node that emitted the trace packet can be determined through the metadata that is delivered together with the trace packet. An example of how this format can easily be used to determine communication hotspots is presented in Section 3.

```
+------+-----------+---------+---------+--------+
| TYPE | DIRECTION | SRC_MAC | DST_MAC | PACKET |
+------+-----------+---------+---------+--------+
```

Fig. 7. Packet Tracking format for the serial interface

3 Use-Case: Optimizing a CoAP Implementation

In preparation of this paper we found ourselves in exactly the same situation as described above – having an implementation of a standardized routing algorithm that wouldn't perform on the testbed, resulting in only a few nodes being reachable from the Smart Service Proxy. To verify our assumption that the poor routing performance resulted from very high traffic load in certain network regions we developed a simple JavaScript-based visualization application running in the WiseGui to find the communication "hot spots".

Therefore, we employed the packet tracking concept described earlier to generate trace packets for all packets being sent and received over the sensor nodes radio interfaces. The trace packets are forwarded to the visualization application running on the experimenters browser. The testbeds self-description is used to determine the physical positions of the individual sensor nodes and to draw them to a canvas accordingly. Upon every trace packet received a packet count for the individual node is incremented and the visualization is updated. Figure 8 shows a screenshot of the visualization. Nodes are displayed as circles containing the number of packets received and sent so far and the radius of the circle corresponds to the packet count.

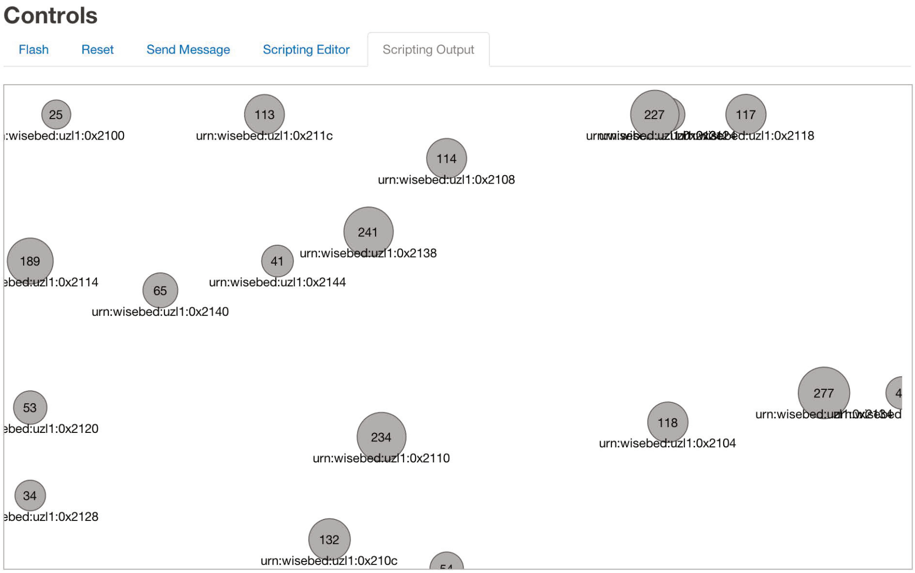


Fig. 8. Packet tracking visualization in the WiseGui

The experiment was executed on the WISEBED testbed at the University of Lübeck. The experiment configuration file, the binary images for the sensor nodes as well as the visualization script are available on http://goo.gl/0AZXl.

4 Conclusion

In this paper, we have motivated the need for appropriate testbed environments that support researchers in conducting experimentally-driven research. By appropriate, we mean that it is possible to debug the wireless communication efficiently. In the past, this was mainly done manually by sending data to the serial interface and by interpreting these results. However, very often this resulted in a lot of guessing about what exactly the source of the problem is. We argue that a testbed infrastructure should inherently support such functionality as some issues only occur at large-scale and cannot be tracked down on a desktop-scale deployment.

As a result, we have presented a framework for debugging the wireless communication in multi-hop networks by extending the WISEBED testbed infrastructure. The overall goal was to create an easy-to-use platform that also allows that other researchers can repeat experiments to verify published results. Our approach has been to use up-to-date web standards in order to be able to run and evaluate experiments in a browser without the need to install additional software. To achieve this, we have designed a RESTful HTTP-based web service API acting as a proxy for SOAP-based WISEBED testbeds and a web-based user interface called WiseGui that supports to script experiments. In addition, we have presented a generic packet tracking framework where nodes emit information about sent and received packets over their serial interface. This data is evaluated by such scripts and we have shown in our use-case how this technology can be used to optimize multi-hop protocols by pinpointing communication hotspots.

We strongly believe that the presented framework has the potential to fundamentally change the way we conduct experimentally-driven research and how results can be verified.

Acknowledgements. This work has been partially supported by the European Union under contract number ICT-2009-258885 (SPITFIRE) and by the Federal Ministry of Education and Research of Germany (Förderkennzeichen 01BK0905, Real-World G-Lab).

References

1. Arora, A., Ertin, E., Ramnath, R., Nesterenko, M., Leal, W.: Kansei: A high-fidelity sensing testbed. IEEE Internet Computing 10, 35–47 (2006)
2. Baumgartner, T., Chatzigiannakis, I., Fekete, S., Koninis, C., Kröller, A., Pyrgelis, A.: Wiselib: A Generic Algorithm Library for Heterogeneous Sensor Networks. In: Silva, J.S., Krishnamachari, B., Boavida, F. (eds.) EWSN 2010. LNCS, vol. 5970, pp. 162–177. Springer, Heidelberg (2010), http://ewsn2010.uc.pt/, ISBN 978-3-642-11916-3
3. Bouckaert, S., Vandenberghe, W., Jooris, B., Moerman, I., Demeester, P.: The w-iLab.t Testbed. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 145–154. Springer, Heidelberg (2011)
4. Chatzigiannakis, I., Fischer, S., Koninis, C., Mylonas, G., Pfisterer, D.: WISEBED: An Open Large-Scale Wireless Sensor Network Testbed. In: Komninos, N. (ed.) SENSAPPEAL 2009. LNICST, vol. 29, pp. 68–87. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-11870-8_6
5. Coulson, G., Porter, B., Chatzigiannakis, I., Koninis, C., Fischer, S., Pfisterer, D., Bimschas, D., Braun, T., Hurni, P., Anwander, M., Wagenknecht, G., Fekete, S.P., Kröller, A., Baumgartner, T.: Flexible experimentation in wireless sensor networks. Communications of the ACM 55(1), 82–90 (2012), http://doi.acm.org/10.1145/2063176.2063198
6. Exscal Research Group, Ohio State University: Extreme scale wireless sensor networking (2010), http://ceti.cse.ohio-state.edu/exscal/
7. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis, University of California, Irvine (2000)
8. Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T.: A survey on facilities for experimental internet of things research. IEEE Communications Magazine 49, 58–67 (2011), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6069710
9. Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919 (Informational) (August 2007), http://www.ietf.org/rfc/rfc4919.txt

10. Netty Project: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients, http://netty.io
11. Pawlikowski, K., Jeong, H.D.J., Lee, J.S.R.: On credibility of simulation studies of telecommunication networks. IEEE Communications Magazine 40(1), 132–139 (2002), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=978060
12. Pfisterer, D., Römer, K., Bimschas, D., Kleine, O., Mietz, R., Truong, C., Hasemann, H., Kröller, A., Pagel, M., Hauswirth, M., Karnstedt, M., Leggieri, M., Passant, A., Richardson, R.: SPITFIRE: Toward a semantic web of things. IEEE Communications Magazine 49, 40–48 (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6069710
13. Riverbed Technology: winpcap (2012), http://www.winpcap.org
14. Rothenpieler, P.: Poster abstract: Distributed protocol stacks for wireless sensor networks. In: 9th European Conference on Wireless Sensor Networks (EWSN 2012), Trento, Italy (February 2012)
15. Shelby, Z., Hartke, K., Bormann, C., Frank, B.: Constrained application protocol (CoAP) (CoRE working group) (2011) Online version at, http://www.ietf.org/id/draft-ietf-core-coap-08.txt (November 01, 2011)
16. Sly Technologies: jNetPcap, http://jnetpcap.com
17. SmartSantander consortium: SmartSantander EU FP7 project (2010), http://www.smartsantander.eu/
18. TCPDUMP: libpcap, http://www.tcpdump.org
19. Werner-Allen, G., Swieskowski, P., Welsh, M.: Motelab: a wireless sensor network testbed. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN 2005. IEEE Press, Piscataway (2005), http://portal.acm.org/citation.cfm?id=1147685.1147769
20. Wittenburg, G., Schiller, J.: A quantitative evaluation of the simulation accuracy of wireless sensor networks. In: Proceedings des 6. Fachgespraechs "Drahtlose Sensornetze" der GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme", Aachen, Germany, pp. 23–26 (July 2007), http://page.mi.fu-berlin.de/wittenbu/research/wittenburg07quantitative.pdf

Evaluating the Effectiveness of a QoS Framework for MANETs in a Real Testbed

Álvaro Torres, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni

Department of Computer Engineering
Universitat Politécnica de Valéncia, Spain
atcortes@batousay.com, {calafate,jucano,pmanzoni}@disca.upv.es

Abstract. Despite all the research efforts in the two previous decades, only a few mobile ad-hoc network (MANET) testbeds have actually been deployed, and even fewer were able to offer Quality of Service (QoS) support. The main problems hindering actual deployment have to do with the distributed effects of mobility, channel contention, and interference. Using simulation or analytical models, several QoS protocols, architectures and algorithms have been presented with the aim of improving QoS support in MANETs. When attempting to translate these research efforts to real testbeds, the difficulty to represent issues like feasibility in real systems, implementation complexity, node deployment and experiment repeatability have prevented their validation. In this paper we present a real implementation of DACME, the QoS framework we propose for mobile ad-hoc networks, and we test its effectiveness in an IEEE 802.11e enabled testbed. Experimental results show that the developed solution is able to achieve good QoS levels, offering sustained bandwidth levels and bounded delay.

Keywords: Quality of Service, MANETs, testbed, distributed admission control, performance evaluation.

1 Introduction

Nowadays, mobile ad hoc networks (MANETs) provide a cheap and infrastructureless form of communication. When combined with an appropriate routing protocol, the IEEE 802.11 standard [1] allows to easily deploy a MANET, which can be very useful in areas where the provision of a central infrastructure is limited or not possible. Typical MANET users share messages and collaborate with each other [2].

Since the IEEE 802.11 standard has been widely used in most wireless LAN environments, the IEEE 802.11e working group [3] proposed an extension to provide QoS support at the MAC level, improving the performance of AP based wireless networks, but also the different networks based on this standard, as in the case of MANETs. The 802.11e extension to the original IEEE 802.11 standard introduces four new traffic categories: Voice, Video, Best Effort, and Background (ordered according to their priority). These four categories provide

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 221–234, 2012.

traffic differentiation by adopting per-category values for the Contention Window (CW) and the Inter Frame Space (IFS) parameters.

Despite the enhancements that IEEE 802.11e has brought, it is still not enough when facing QoS flow concurrency. In fact, one of the most crucial components of a system attempting to provide QoS guarantees is the Admission Control Module (ACM). This module should be able to estimate the resources of the network and decide when application flows should be admitted or rejected, avoiding to interfere with previously active flows. Unfortunately, this is not an easy task since MANETs are highly dynamic environments, and thus flow admission does not guarantee good QoS conditions throughout time. So, although significant efforts have been done in this area, the solution to the problem is not trivial.

In this paper we develop and evaluate a real implementation (i.e., using a real IEEE 802.11e enabled testbed) of DACME [4], a low power consumption and low complexity end-to-end admission control module. The proposed solution imposes no constraints on the intermediate nodes of the communication other than being able to route packets. Experimental results confirm the goodness of DACME at enhancing QoS support in wireless multi-hop environments. Then, based on the acquired experience, we propose an enhancement to the DACME decision module which offers better adaptability to the network bandwidth fluctuations under wireless interference conditions.

The rest of this paper is organized as follows: in section 2 we review the state of the art on admission control solutions for MANETs. Section 3 presents a description of DACME, the admission control system we selected to implement and test, along with the proposed enhancement to DACME's decision module. Section 4 offers a brief description of the testbed used for testing. Then, section 5 presents the experimental results and discussion. Finally, section 6 presents the conclusions of this work along with future works.

2 Related Works

In the literature we can find several theoretical admission control (AC) algorithms for MANET environments. The main drawback associated with the different solutions we have reviewed is that they have only been tested on simulated environments, and, to the best of our knowledge, none of them has been implemented and tested in a real environment.

According to the guidelines provided on the survey by Hanzo et al. [5], the different AC algorithms available can be divided in two large groups: *routing coupled* and *routing decoupled*. In the first group we can find different algorithms such as ACRMP [6], or MACMAN [7]. All of these AC algorithms require modifying the routing algorithm to support the AC extension. This strategy has some benefits, such as shorter admission times and less overhead of the AC protocol, mainly because they use routing packets to measure the state of the network. However, this first group also presents an important drawback: since they are coupled with a specific routing algorithm, a different routing protocol can not be used without losing the AC module. Another drawback is the strong requirements imposed on nodes, forcing every node in the network to adopt the modified routing protocol to support the AC module. This means that even low power nodes, and nodes which do not require the AC module, will have to dedicate additional resources to support it.

With respect to routing decoupled algorithms, their main advantage is that they allow using any routing protocol for MANET environments. Within the routing decoupled algorithms group, an additional division can be made between *stateful* and *stateless* protocols. Stateful algorithms save certain information about the state of the links in every node. This strategy allows AC algorithms such as INSIGNIA [8] or MPARC [9] to store information about QoS conditions relative to past flows, and decide, based on this information, whether to accept or reject a flow. Thus, similarly to what occurs for routing coupled protocols, they impose several restrictions on intermediate nodes and require more computing power from these nodes. Within the stateless, routing decoupled protocols group, we can find solutions such as DACME [4], which do not impose any restriction on intermediate nodes since they need not store any information about past flows, nor must they have a high computing power.

In this paper we will develop and test DACME. We chose this proposal since it has been proved to be a powerful and efficient AC protocol, and yet easy to implement and deploy in real systems.

3 DACME Overview

In this section we present a brief description of DACME (Distributed Admission Control for MANET Environments) [4]. DACME is a distributed admission control system which allows achieving per flow QoS requirements in terms of bandwidth and delay. One of the main advantages of DACME is that it does not impose any specific requirements on MANET nodes besides the use of IEEE 802.11 and having routing support; in fact, DACME agents are only required at the communication endpoints. Since DACME does not have MAC level constraints, it can be implemented in an easy way on all systems supporting the standard TCP/IP architecture.

3.1 Admission Decision Algorithms Meeting Bandwidth Requirements

DACME learns about the network status using a probe/response strategy. In particular, to achieve an accurate bandwidth estimation, DACME uses a burst of probe packets periodically generated by the source in a back-to-back fashion. These packets arrive to the destination node with an average inter-packet time gap which allows the destination node to make an estimation of the available end-to-end bandwidth. When the destination gathers all the data required, it sends a response packet with the current bandwidth estimation (BM) back to the source. An illustration of this strategy can be found in figure 1.

When receiving a response, the source applies an algorithm to decide whether the flow is accepted or rejected. The admission control algorithm adopted by

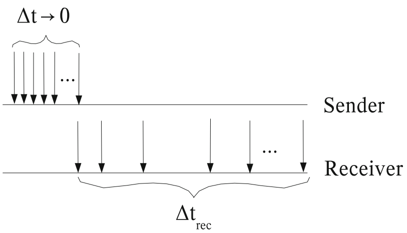


Fig. 1. Bandwidth estimation strategy

Algorithm 1. The CIB-DA decision algorithm

```
*After receiving each* $BM_i$ ***do*** {
  $\mu_i \leftarrow \frac{(i-1)\cdot\mu_{i-1}+BM_i}{i}$, $\sigma_i \leftarrow \sqrt{\frac{(i-2)\cdot(\sigma_{i-1})^2+(BM_i-\mu_i)^2}{i-1}}$
  ***if*** $(\mu_i - t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}} > B_R)$
    ***then*** *Flag(BW)* $\leftarrow$ *1*
  ***else if*** $(\mu_i + t_{i-1,0.95}\frac{\sigma_{p,i}}{\sqrt{i}} < B_R)$
    ***then*** *Flag(BW)* $\leftarrow$ *0*
  ***else if*** $(i < 5)$
    ***then*** *send a new probe* }
```

DACME relies heavily on bandwidth estimations to decide whether to admit or deny a flow. For this reason, in this section we will focus explicitly in this DACME algorithm. In particular, we first describe CIB-DA, the original decision algorithm proposed in [4], and we then propose HRCI-DA, a novel mechanism we have developed based on experimentation, which offers an improved behavior in real environments compared to the former.

CIB-DA: Confidence Interval Based Decision Algorithm

The CIB-DA algorithm uses the values of up to five probes to obtain a 95% confidence interval for the available bandwidth value, based on which flow acceptance/denial decisions are made. Algorithm 1 shows the pseudo-code that describes the behavior of the CIB-DA algorithm. Previous works [4] have shown that this algorithm is highly effective in simulated MANET environments.

CIB-DA is executed every time a probe reply is received. Decisions are based on statistical confidence levels; therefore, i refers to the current iteration, $t_{i-1,0.95}$ to a Student's t-distribution with $i-1$ degrees of freedom, and for a confidence level of 95%. Parameter B_R refers to the bandwidth required by the application, while BM refers to the bandwidth measurement explained previously. If the application is solely bandwidth constrained, the value of the bandwidth flag - *Flag(BW)* - will determine whether the QoS flow can be accepted. If the application has delay requirements as well, the value of the delay flag is also considered. These issues are addressed in later sections.

Algorithm 2. The HRCI-DA decision algorithm

```
After receiving each $BM_i$ do {
  $range \leftarrow max(\{BM\}) - min(\{BM\})$
  $\mu_i \leftarrow \frac{(i-1)\cdot\mu_{i-1}+BM_i}{i}$, $\sigma_i \leftarrow \sqrt{\frac{(i-2)\cdot(\sigma_{i-1})^2+(BM_i-\mu_i)^2}{i-1}}$
  find the unbiased bandwidth estimator $\upsilon_{p,i}$
  if $(\frac{range}{2} > t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}})$ then
    $(\hat{BW}_{low}, \hat{BW}_{high}) \leftarrow (\mu_i - t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}},\ \mu_i + t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}})$
  else    $(\hat{BW}_{low}, \hat{BW}_{high}) \leftarrow (\mu_i - \frac{range}{2},\ \mu_i + \frac{range}{2})$
  if $(\hat{BW}_{low} > B_R)$ then $Flag(BW) \leftarrow 1$
  else if $(\hat{BW}_{high} < B_R)$ then $Flag(BW) \leftarrow 0$
  else if $(i < 5)$ then send a new probe }
```

HRCI-DA: Hybrid Range/Confidence Interval Decision Algorithm
Contrarily to what occurs in simulations, lots of problems exist in real testbeds, not only at the transmission level (e.g. interferences, packet loss), but also at the application, kernel, and hardware levels. Examples of effects occurring at these levels include (but are not limited to) CPU Usage, RAM paging, time measurement delays, and loss of synchronization.

Focusing on the interferences problem, one of the main differences between simulation and real testbed experiments has to do with the wireless channel. While in simulation experiments wireless channels are free from external interferences, in testbeds this is rarely true. In fact, in real environments, our implementation of DACME using the CIB-DA algorithm exhibits significant interferences, which caused estimated bandwidth to experience frequent and drastic fluctuations. This impeded obtaining low confidence intervals in most situations since the CIB-DA algorithm used a t-Student function with only a few degrees of freedom for calculating those intervals. Such large confidence intervals provoked that most flows were not accepted, or decisions could not be made even when the minimum bandwidth values measured were higher than the demanded ones.

To avoid this problem we proposed the HRCI-DA algorithm, an improvement to the CIB-DA measurement algorithm. The main goal of HRCI-DA is to bound the interval used to make flow acceptance/denial decisions, never allowing it to become higher than the half range, that is, half the difference between the maximum and minimum measurements. Algorithm 2 shows the pseudo-code for HRCI-DA.

In this algorithm two different intervals are used to determine the value of the bandwidth flag - *Flag(BW)* -, being one based on the range of the values, and the other based on confidence intervals. The former is used whenever the values produced by the latter strategy are excessive.

3.2 Admission Decision Algorithms Meeting Delay Requirements

For delay estimations, DACME employs a ping-pong method without any delay between consecutive request/response patterns. At the sender node, the measured RTT of the end-to-end path is used to estimate network delay. To obtain the delay value and determine whether a new flow is going to be accepted, an adjustment function is used; this function is described in [4], and its main purpose is to make short-term measurements match long-term ones. Only when both delay and bandwidth restrictions are achieved can a flow be accepted, being blocked otherwise. An illustration of the basic delay estimation strategy is shown in figure 2.

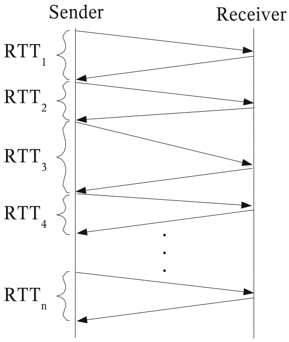


Fig. 2. Delay estimation method

3.3 Implementation Details

To evaluate the effectiveness of DACME in a real environment, we developed an application level library[1] that interacts with both the applications and the kernel to achieve all the required functionality. Figure 3 shows the interaction between the different DACME elements.

Initially the application must register with DACME by using a modified socket interface that also accepts flow QoS specifications (bandwidth required, maximum delay, maximum jitter) as input. This interface, which is part of the developed library, exchanges information with both the operating system (creating a regular socket for communications) and DACME's core. The flow is registered with DACME by including not only the QoS specifications, but also the connection details (source port, destination port, destination IP). In a second step, DACME's QoS measurement module will probe the end-to-end path using the techniques described above. By interacting with the DACME agent at the destination, the DACME agent at the source will gather information that will allow it to make admission control decisions using any of the bandwidth-based decision

[1] Freely available at: `http://dacme.sourceforge.net/`

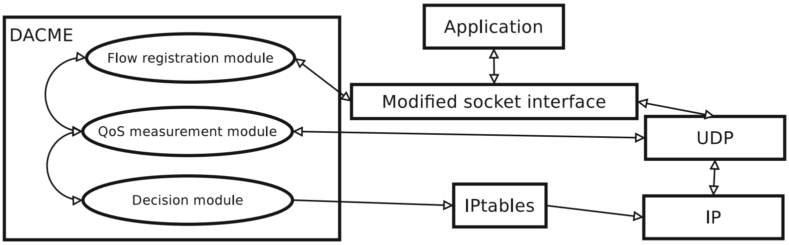


Fig. 3. DACME diagram illustrating the dependencies among the different elements involved

algorithms presented in section 3.1, and the delay-based decision algorithms presented in section 3.2. According to the decision made, this module relies on the *iptables* tool [10] to dynamically block flows during periods of congestion, and unblocking them when QoS conditions are again met. The IP ToS header field of accepted flows is also modified in order to take advantage of the IEEE 802.11e Video and Voice medium access categories. Notice that the IEEE 802.11e MAC driver will automatically map the IP ToS values to the four different medium access categories available.

4 Testbed Setup

For testing the effectiveness of DACME in a real testbed we used a set of Asus EeePC netbooks, regular laptops, and a desktop computer. All the netbooks have a Ralink RT2860 wireless card, and the laptop and desktop systems use a Linksys WUSB600N USB wireless card which employs the Ralink RT2870 chipset. All wireless cards support the IEEE 802.11n draft3 standard, which includes IEEE 802.11e QoS extensions by default. The drivers employed were available in Linux kernel version 2.6.32, allowing us to build a realistic system with all its inherent characteristics and problems.

The stations are wirelessly distributed and connected to achieve a seven-hop ad-hoc network, which allows us testing with different hop number combinations per flow. The different stations involved in the tests are also interconnected via Ethernet for remote experiment control, and as a return channel to measure the delay of the UDP packets injected. To avoid high CPU usage, we select different source/destination pairs for the different traffic flows. To introduce variable degrees of congestion in our tests, we also relied on Best Effort traffic flows, and all the video flows share a same link which becomes the network's bottleneck.

To manage the repeatability of the experiments we used Castadiva [11], which we extended to provide DACME compatibility. Castadiva allows us to automate large sets of experiments and collect all the statistics required.

In our experiments we vary both the number of best effort traffic flows and QoS (video) traffic flows. To introduce variable degrees of congestion, each best effort traffic flow consists of a 1.5 Mbit/s UDP stream; by varying the number of best effort flows we were able to achieve different channel congestion levels. With respect to video flows, they consisted of synthetic traffic at a rate of 1 Mbit/s (unidirectional) demanding to DACME a maximum end-to-end delay of 300ms. Table 1 summarizes the number of hops for each flow. Aditionally, one of the links is shared by all traffic flows to aggravate the problem of congestion. Notice that all hop count values are representative of typical MANET studies.

Table 1. Flow endpoint definition for both video and best effort traffic flows

Video	*Number of hops*	*Best effort*	*Number of hops*
Flow #1	3 hops	Flow #1	7 hops
Flow #2	3 hops	Flow #2	3 hops
Flow #3	3 hops	Flow #3	5 hops
Flow #4	3 hops	Flow #4	4 hops

5 Experimental Results

In this section we perform a detailed evaluation to assess the correct functionality of our implementation of DACME, as well as the improvements introduced by both DACME AC algorithms (CIB-DA and HRCI-DA). With this goal we created three scenarios: in the first one we varied congestion by increasing the number of best-effort flows (background traffic), in the second one we increased the number of competing QoS flows, and, in the last one, we varied the maximum delay restriction for QoS flows. Our goal was to study the QoS stability of DACME flows in a real environment. Thus, for each test, the performance parameters under analysis were: throughput, delay, total activity time per-flow, and mean number of DACME on/off state transitions. This last parameter is interesting as the less a flow switches it's state, the more stable the flow is for an end-user.

5.1 Varying the Number of Best Effort Flows

In this first set of experiments we study how best effort traffic affects the stability of video flows. In our experiments we have three concurrent video flows which start at random times during the first 20 seconds of the experiment, and then they last for 80 additional seconds. Also, for each test, we increase the number of best effort flows from zero to four to assess the impact of congestion on QoS performance. Notice that, for all the presented results, each measurement corresponds to the mean value of 25 independent tests. Throughput and delay values include 95% confidence intervals.

Figure 4 (left) shows the mean time each QoS flow is active. Notice that, if we do not use DACME, the activity time is always the maximum. The usage of

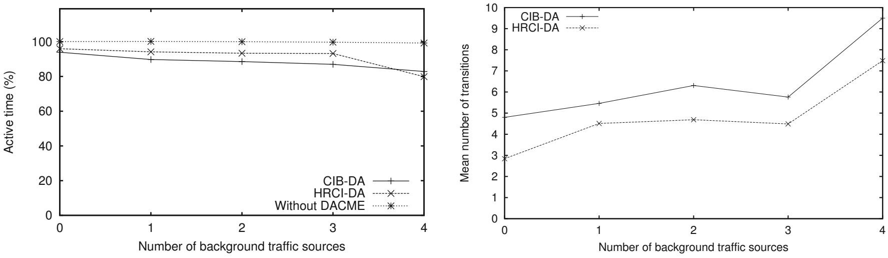


Fig. 4. Mean active time per flow (left) and mean number of state transitions for DACME flows (right) when varying the number of background traffic sources

DACME implies that, during certain periods, the flow can not be accepted since QoS requirements are not met; however, we find that, on average, the activity periods are maintained high, as desired. Focusing on the differences between using and not using DACME, we can see, significant benefits in terms of both bandwidth and delay values.

Figure 5 (left) shows the mean throughput per video flow during periods of activity. We can see that bitrate values are maintained close to the maximum if DACME is used (for both admission control algorithms). When DACME is not used, we can clearly observe the negative impact of background traffic, drastically affecting the QoS of the video flows. The confidence intervals presented further evidence the goodness of both DACME decision algorithms, showing that bitrate variability associated with these algorithms is much lower compared to the "Without DACME" situation. In particular, we found that the standard deviation when using DACME was never higher than 19% while, when not using DACME, it is never lower than 30%, surpassing 100% in the worst case.

Concerning delay, figure 5 (right) shows that the differences between using DACME and not using it are again quite noticeable. In particular, we can see that, when the video flows are managed by DACME, the delay is typically lower than 100ms, therefore being adequate for real-time communication. On the contrary, if we do not use DACME, the mean delay rises up to 700ms, being the lowest value of about 250ms. Similarly to what occurs for throughput, the confidence intervals for the delay are very low if DACME is used, becoming quite high otherwise.

If we look at the differences between CIB-DA and HRCI-DA, we find that the former is clearly more restrictive, typically introducing more flow blockage in order to meet the QoS requirements. This can be observed in Figure 4 (right), which shows the mean number of state transitions per flow, that is, transitions from active to blocked state, or the opposite. Since DACME performs periodic bandwidth measurements, it can block flows during ongoing communication as soon as QoS loss is detected. Although these dynamic decisions are mandatory to handle the effects of mobility, it is important to maintain this value as low as possible. Results show that our HRCI-DA algorithm offers significant improvements compared to CIB-DA, reducing the mean number of state transitions by

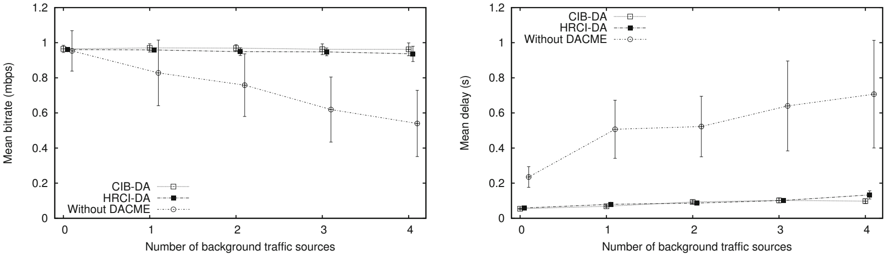


Fig. 5. Mean values for throughput (left) and end-to-end delay (right) when varying the number of background traffic sources

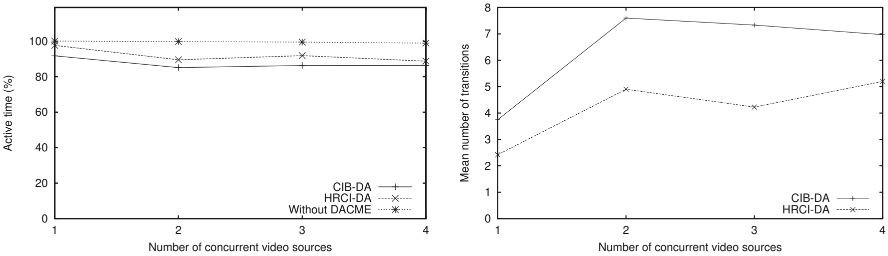


Fig. 6. Mean active time per flow (left) and mean number of state transitions for DACME flows (right) when varying the number of concurrent video sources

up to 40%, and making communication more fluid, while not being differenciable in terms of throughput or delay from CIB-DA.

5.2 Varying the Number of Video Flows

The goal of this second set of experiments is to assess the performance of our implementation of DACME when handling a variable number of QoS flows, as well as analyzing the interactions between these flows. With this purpose our experimental settings are similar to those of the previous section, but we now fix the number of best effort flows to three, while increasing the number of video flows from one to four.

Figure 6 (left) shows the results of the active time per flow. Similarly to the previous section, we find that the percentage is quite high for both DACME algorithms, and that increasing the number of video sources does not cause a proportional decrease in terms of activity time. This is mostly due to the distributed nature of wireless channel access in MANETs.

While being able to maintain a high activity time, in terms of throughput, figure 7 (left) shows that both DACME AC algorithms allow achieving a similar throughput (nearly 1 Mbit/s) while, when DACME is not used, this value drops to about 0.6 Mbit/s. As in the previous set of tests, the confidence interval obtained is much lower when we use DACME, meaning that variability is strongly

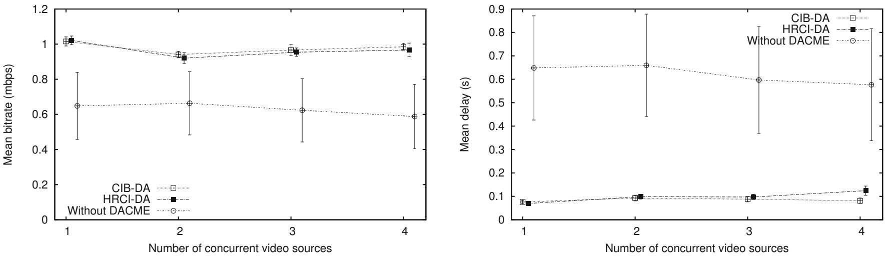


Fig. 7. Mean values for throughput (left) and end-to-end delay (right) when varying the number of concurrent video sources

reduced compared to the "Without DACME" case. This set of experiments allows concluding that, even when using IEEE 802.11e to achieve traffic differentiation at the MAC layer, the Video traffic category at the MAC layer is still highly affected by Best Effort traffic interferences; thus, although IEEE 802.11e allows some differentiation between the different traffic categories, it is not powerful enough to provide, by itself, full QoS guarantees in MANET scenarios.

Focusing on the delay (see figure 7, right), the behavior is similar to the previous tests. Again, the delay experienced by QoS traffic becomes excessive when DACME is not used. Additionally, we find that, when using either DACME decision algorithm, the delay experienced by the video flows is maintained low, and mostly immune to the increase of video sources.

Comparing both decision algorithms, we find that HRCI-DA offers better results in terms of both activity time (figure 6, left) and mean number of state transitions (figure 6, right). In fact, the latter experiences a reduction of up to 30%, which shows the effectiveness of HRCI-DA at improving video streaming stability in real environments compared to its predecessor.

5.3 Varying the Maximum Delay Restriction

In this third and last set of experiments our goal is to validate our implementation of DACME when varying the maximum delay allowed for the video flows to determine the degree of compliance achieved. With this aim we fix the number of Video and Best Effort flows to three each, and we vary the maximum delay restrictions from 100 to 600 ms, again comparing both DACME AC algorithms against a solution where DACME is not used.

Figure 9 (left) shows that DACME is able to sustain the bitrate values at near-optimum levels and with little variability, contrarily to the "Without DACME" situation.

With respect to the delay experienced (Figure 9, right), we find that the mean delay values are very low when we use DACME compared with the "Without DACME" situation. If we look deeper into the delay restriction accomplishments, for the lowest value (100 ms), about 70% of the packets comply with the

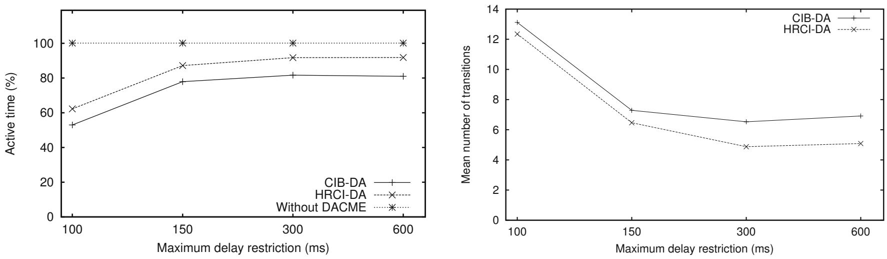


Fig. 8. Mean active time per flow (left) and mean number of state transitions for DACME flows (right) when varying the maximum delay restriction

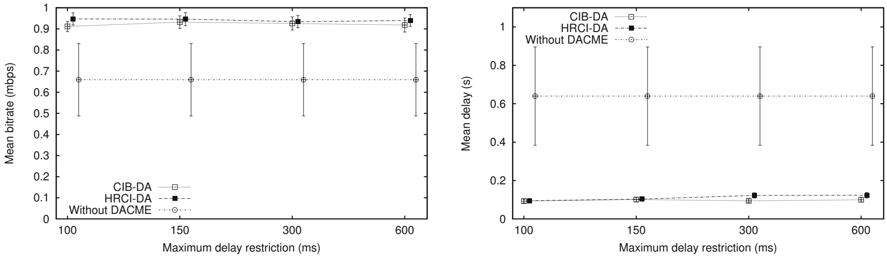


Fig. 9. Mean values for throughput (left) and end-to-end delay (right) when varying the maximum delay restriction

maximum delay restriction, while for 150ms the level of delay compliance is of 83%, growing to more than 94% for the last two cases (300 and 600 ms).

Comparing both DACME algorithms, Figure 8 (left) shows the mean time of activity when increasing the maximum delay allowed. We find that low delay requirements have a significant impact on activity time, reducing it up to 46% for both AC strategies. Again, HRCI-DA is able to improve the mean time of activity achieved by CIB-DA, achieving an increase of up to 19%. For this metric, the "Without DACME" values remain constant since the different delay restrictions are only meaningful within the scope of DACME.

Varying the maximum delay restriction has also a significant impact in terms of the mean number of state transitions, as shown in Figure 8 (right). Again HRCI-DA shows a better behavior, in this case by reducing the number of state transitions involved. It is also worth noticing that the main differences detected occur when increasing the maximum delay restriction from 100 to 150 ms, being the overall behavior mostly maintained afterward. This occurs because most of the delay values measured are in the 100-150 ms range.

Overall, the results presented in this section validate the effectiveness of DACME in real testbeds, and evidence the improvements introduced by the HRCI-DA decision algorithm compared to its predecessor (CIB-DA) in real environments. Our HRCI-DA algorithm is able to increase the overall activity time and reduce the number of transitions, while maintaining good QoS values in terms of both throughput and delay. Hence, we can conclude that our DACME implementation mostly retains the QoS properties inferred based on simulation results, although some adjustments can help at further boosting performance in real environments.

6 Conclusions and Future Work

In this paper we validated a real implementation of a distributed admission control system for MANETs that was previously evaluated through simulation. We test its effectiveness in a real testbed using different performance indexes such as throughput, delay, and total time of activity.

To cope with bandwidth estimation accuracy problems occurring in real environments, we proposed an enhanced decision algorithm (HRCI-DA) for the admission control module that offers significant performance improvements compared to the previous version (CIB-DA). HRCI-DA reduces the number of on/off state transitions for QoS flows, and improves the total activity times, while also maintaining good throughput and delay values.

Overall, the results presented in this paper clearly show that: (i) traffic differentiation provided by IEEE 802.11e is not enough in real multi-hop ad-hoc networks, and so a distributed admission control like DACME becomes essential; (ii) the DACME QoS architecture was fully effective in a real testbed, successfully validating the previous simulation results obtained; (iii) the proposed HRCI-DA algorithm improves the original one by providing greater stability to QoS flows, increasing their total activity time and reducing the total number of on/off state transitions; and (iv) despite the greater number of active QoS-flows, bandwidth and delay values are maintained or even improved by HRCI-DA compared to CIB-DA.

As future work we plan to test our implementation with scalable video streams, adapting DACME decision algorithms to the multi-level quality characteristics inherent to such streams, thus offering the possibility for adaptive real-time multimedia traffic in MANETs.

Acknowledgments. This work was partially supported by the Ministerio de Ciencia e Innovación, Spain, under Grant TIN2011-27543-C03-01, and by the Ministerio de Educación FPU (Formación de Profesorado Universitario) grant program - AP2009-2415.

References

1. IEEE 802.11 WG. International Standard for Information Technology - Telecom. and Information exchange between systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999(E) IEEE Std. 802.11 (1999)

2. Cano, J., Cano, J.-C., Toh, C.-K., Calafate, C.T., Manzoni, P.: EasyMANET: an extensible and configurable platform for service provisioning in MANET environments. IEEE Communications Magazine 48(12), 159–167 (2010)
3. IEEE 802.11 WG. 802.11e IEEE Standard for Information technology- Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements (2005)
4. Calafate, C.T., Malumbres, M.P., Oliver, J., Cano, J.C., Manzoni, P.: QoS Support in MANETs: a Modular Architecture Based on the IEEE 802.11e Technology. IEEE Transactions on Circuits and Systems for Video Technology 19(5), 678–692 (2009)
5. Hanzo II, Tafazolli, R.: Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey. IEEE Communications Surveys & Tutorials 11(4), 78–108 (2009)
6. Derhab, A., Bouabdallah, A.: Admission control scheme and bandwidth management protocol for 802.11 ad hoc networks. In: 4th International Conference on Innovations in Information Technology, IIT 2007, pp. 362–366 (November 2007)
7. Lindgren, A., Belding-Royer, E.M.: Multi-path admission control for mobile ad hoc networks. In: Proceedings of Mobiquitous (2005)
8. Lee, S.-B., Ahn, G.-S., Zhang, X., Campbell, A.T.: INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. Journal of Parallel and Distributed Computing 60, 374–406 (2000)
9. Yang, Y., Kravets, R.: Throughput guarantees for multi-priority traffic in ad hoc networks. Ad Hoc Networks 5(2), 228–253 (2007)
10. The netfilter.org iptables project, http://www.netfilter.org/ (accessed July 28, 2011)
11. Hortelano, J., Cano, J.-C., Calafate, C.T., Manzoni, P.: Testing applications in manet environments through emulation. EURASIP Journal on Wireless Communications and Networking 2009, Article ID 406979, 20 pages (2009), doi:10.1155/2009/406979

Wireless Sensor Network for Continuous Temperature Monitoring in Air-Cooled Data Centers: Applications and Measurement Results

Thomas Scherer, Clemens Lombriser, Wolfgang Schott,
Hong Linh Truong, and Beat Weiss

IBM Zurich Research Laboratory (ZRL), Rüschlikon, Switzerland
tsc@zurich.ibm.com

Abstract. Temperature monitoring in data centers is essential for reliably operating the data processing equipment and minimizing the required cooling energy. For this purpose, we track the temperatures at key locations in the data center with low-cost sensors and forward the captured information via the ZRL Data Center Wireless Sensor Network (DCWSN) to a monitoring client. Applications include continuous temperature monitoring, data collection for thermal modeling, and temperature sensing for real-time control of cold air flow and workload allocation. The DCWSN has been successfully deployed in production data centers.

Keywords: data center, wireless sensor network, energy efficiency, temperature monitoring, thermal modeling, thermal management.

1 Introduction

Data centers with centralized computing and storage resources are an integral part of modern information technology infrastructure. A recent study shows that power consumption in data centers today accounts for about 1.5 % of the total electricity use in the world [1]. A large part of the electric power consumed in data centers is used for cooling to prevent device overheating and to ensure maximum availability and reliability of the data processing equipment used for computing, storage and communication.

In air-cooled data centers, racks are typically arranged to form alternating cold and hot aisles as shown in Figure 1. The air inlets of the data processing equipment in the racks face the cold aisles, where chilled air from the computer room air conditioners (CRACs) is provided via the raised-floor plenum through perforated floor tiles placed directly in front of the racks. The hot exhaust air from the air outlets at the rear of the racks intermixes with ambient air and eventually circulates back to the CRACs. The reference value for the cold air supply temperature is chosen such that the inlet air temperatures do not exceed the maximum admissible temperature specified by the device manufacturers. In many data centers, cooling energy is wasted because the cooling system is operated at significantly lower temperatures than actually necessary. This approach

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 235–248, 2012.

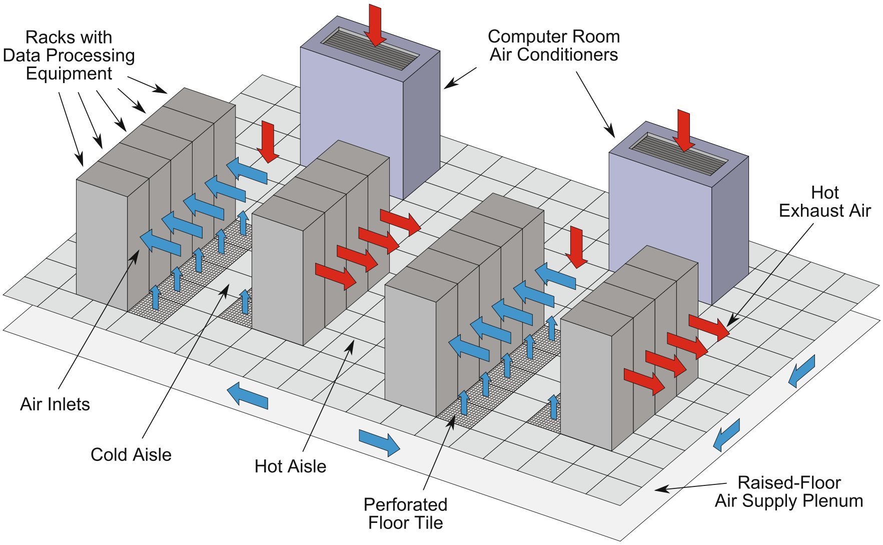


Fig. 1. Layout of an air-cooled data center with alternating hot and cold aisles

allows compensating local hot spots and reduces the potential risk of reacting too late to a harmful temperature increase in the data center, but leads to up to 5 % more cooling energy consumption for each degree Celsius below the upper temperature limit. To enable data center operators to run their cooling system closer to the economically attractive upper limit, continuous temperature monitoring at thermally critical locations in the data center is required. In addition, thermal models based on measurement data can be used to analyze and optimize layout, air flow and workload distribution in the data center. Changing operating parameters with sophisticated control concepts based on real-time temperature information can optimize the cooling-efficiency even further.

A low-cost wireless sensor network (WSN) is well suited for gathering the required data for continuous temperature monitoring, thermal modeling and real-time control without any changes to the existing data center infrastructure. The sensors can be quickly deployed and easily repositioned if data processing equipment in the data center is relocated or replaced. We propose to use the ZRL Data Center Wireless Sensor Network (DCWSN) that is tailored to the needs of data center monitoring applications. The battery-powered wireless sensor nodes in the DCWSN capture temperature data at key locations in the data center and and forward the data via relay nodes to a monitoring client. The protocol stack [2] performs all required network functions and uses the publish/subscribe messaging protocol MQTT-S [3] for communicating between sensor nodes and the monitoring application.

The rest of this paper is structured as follows: Section 2 gives an overview of wireless sensor network applications in data centers. In Section 3, we present the ZRL Data Center Wireless Sensor Network, a solution designed for these applications. Finally, the deployment of the DCWSN in a production data center, measurement results and network performance are discussed in Section 4. The paper concludes with a brief summary.

2 WSN Applications in Air-Cooled Data Centers

2.1 Temperature Monitoring

Thermal problems in data centers may occur at any time, for example because of a sudden malfunction of a cooling device or after adding, replacing or relocating data processing equipment in the data center. It is important to systematically detect such problems to take appropriate actions as early as possible. For reliably operating all data processing equipment in the data center, the temperatures at the air inlets of these devices have to be monitored continuously to ensure that the maximum admissible inlet air temperature specified by the device manufacturers are not exceeded.

There are several possible approaches to monitor temperatures in data centers. For example, most data processing devices are equipped with internal temperature sensors. To detect thermal problems in the data center, however, data from these internal sensors is generally not the first choice because it reflects the activity of the device rather than the environmental conditions of the data center. Furthermore, high installation and configuration effort is required to collect and aggregate this data, especially in environments with heterogeneous devices from different manufacturers. There exist several solutions with external wired sensors, but in practice they are not widely adopted, mainly because of the difficulty to deal with changes in the data center layout. A wireless sensor network, on the other hand, offers a low-cost non-intrusive way to gather temperature data at key locations in the data center. The sensors can be quickly deployed and easily repositioned if data processing equipment is relocated or replaced.

Temperature data must be accessible from remote clients in real-time to visualize the temperature field in the data center and trigger alarms if a problem occurs. Sampling the temperature every 30 seconds is usually sufficient for monitoring applications. Archived data can be used to assess temperature trends and perform analytics to optimize the cooling concept of the data center.

2.2 Thermal Modeling

Thermal models are essential tools for optimizing data center cooling concepts. They allow studying the effects of layout changes and parameter variations on the temperature field without interfering with the data center operation.

Computational Fluid Dynamics (CFD) Simulations. Air flow in data centers is generally turbulent and can be modeled with the Navier-Stokes equations. For solving these nonlinear partial differential equations coupled with the energy equation numerically, the fluid domain is discretized into a number of control volumes. Computer room air conditioning units, servers and other data

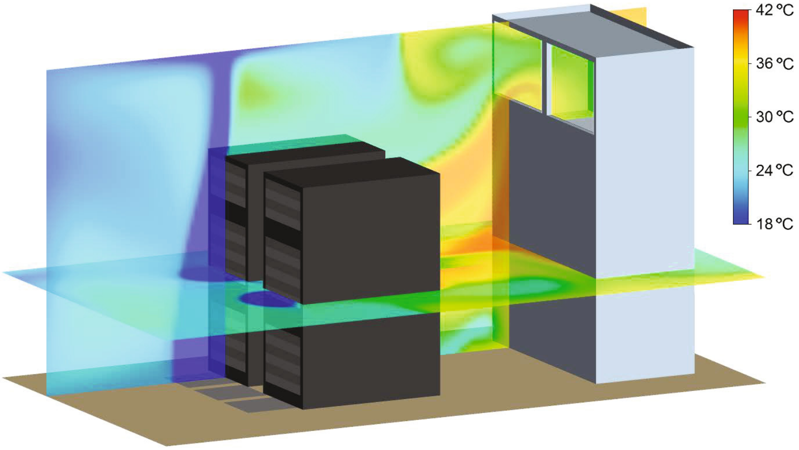


Fig. 2. CFD simulation result: Temperature field in a small data center compartment

processing equipment in the data center are commonly modeled as blocks with prescribed inlet and outlet boundary conditions. We rely on measured data to define these boundary conditions and for validating the CFD models. For validation, we require a high spatial resolution in thermally critical areas of the data center. Figure 2 shows a CFD simulation result for a small data center compartment with two racks and one air conditioning unit.

Simplified Physics-Based Models. CFD simulations require a high computational effort and are thus not suitable for real-time applications or for solving multidimensional optimization problems. A simplified physics-based approach is to model the heat flow between racks or individual devices, further on referred to as server nodes. For a server node i, the relationship between power consumption P_i, inlet and outlet air temperatures T_i^{in} and T_i^{out}, and volumetric air flow rate f_i is given by

$$P_i = \rho c_p f_i \left(T_i^{out} - T_i^{in} \right) , \tag{1}$$

where ρ and c_p are the density and specific heat of air, respectively. In [4], Tang et al. introduced a sensor-based fast thermal evaluation model for data centers that characterizes the heat flow as cross-interference among server nodes: The cross-interference coefficients a_{ij} correspond to the fraction of heat flowing from server node i to server node j. With this notation, the heat flow $Q_i^{out} = \rho c_p f_i T_i^{out}$ at the outlet of server node i can be modeled by

$$Q_i^{out} = \sum_{j=1}^{N} a_{ji} Q_j^{out} + Q_i^{sup} + P_i \, , \quad \text{for } i = 1, \ldots, N, \tag{2}$$

where $Q_i^{sup} = \rho c_p \left(f_i - \sum_{j=1}^{n} a_{ji} f_j \right) T^{sup}$ is the heat flow of the cold air from the CRAC to server node i and N is the number of server nodes in the data center.

The N^2 cross-interference coefficients can be determined with temperature data of N independent operating points of the data center. This heat flow model represents a simple, but efficient method to estimate the outlet air temperatures for a given workload distribution.

Reduced-Order Models. Reduced-order modeling techniques can be used to extract the dominant characteristics of a system and are well suited to accomplish low-dimensional turbulence modeling for data centers. In [5], Samadiani and Yoshi proposed a thermal modeling approach based on the proper orthogonal decomposition (POD) for rapidly computing the temperature field T of a data center as a function of a set of system design parameters such as air flow and temperature. The POD-based method expands a set of observed data onto a set of m linear independent basis functions ψ according to

$$T = T_0 + \sum_{i=0}^{m} b_i \psi_i \,, \tag{3}$$

where $\psi_i, i = 1, \ldots, m$, are referred to as the POD modes and $b_i, i = 1, \ldots, m$, as the corresponding POD coefficients. To obtain a POD-based reduced-order model of the temperature field of a data center, first the physical parameters of interest are changed n times to generate a set of n linearly independent observations of the temperature field. Averaging the observations yields the reference temperature field T_0. In the next step, the POD modes are calculated by solving an n-dimensional eigenvalue problem. Since the energy captured by each POD mode is proportional to each eigenvalue, the eigenvalues are sorted in a descending order to ensure that the first POD mode captures the largest energy. Finally, the POD coefficients are computed as a function of each set of system design variables. POD-based reduced-order models represent a powerful tool for rapidly evaluating the temperature distribution in the data center based on sparse temperature measurements.

Requirements for the Wireless Sensor Network. For parameter identification, definition of boundary conditions, and model validation, we require measured temperature data with high spatial and/or temporal resolution. Number and locations of the sensors as well as sampling rate heavily depend on the type and purpose of the model. Generally, inlet and outlet air temperatures of air conditioning units and data processing equipment in the data center and temperatures below the raised floor are of particular interest.

2.3 Real-Time Control

CFD simulations and experiments show that the cold air supply temperature can be significantly increased if the air flow is adapted dynamically based on measured inlet air temperatures of the data processing equipment. In data centers with significantly varying workload, this approach can yield cooling energy savings of up to 20 % [6]. Floor tiles in thermally critical areas of the data center have to be equipped with controllable dampers such as louvers or sliding grates.

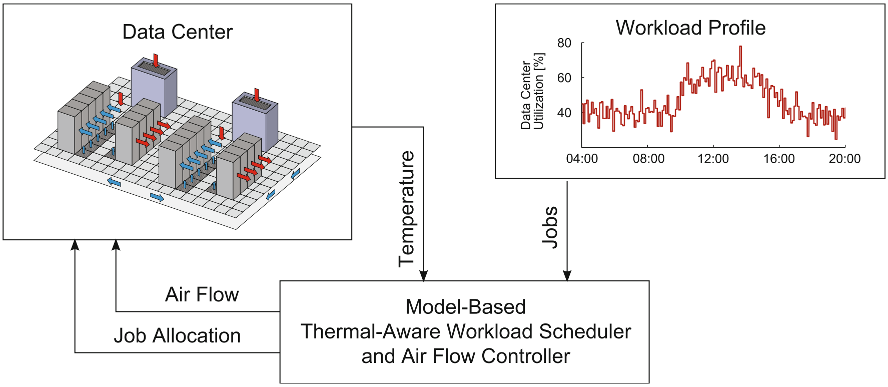


Fig. 3. Combined thermal-aware workload scheduling and air flow control concept

Constant pressure control at the CRAC ensures that the right amount of air is provided via the air supply plenum below the raised floor.

Another approach to balance the temperature distribution is to move workload between servers in the data center. This is known as thermal-aware workload scheduling. In [7], we proposed to combine thermal-aware workload scheduling with air flow control to further optimize the system. The operating point of the data center is determined based on temperature measurements and an estimation of the workload. Tasks are allocated to servers such that the outlet temperatures remain close to the set point. The air flow is only adjusted if the workload scheduler is not able to maintain the desired set point anymore and the measured temperatures deviate too much from the set point. A high level description of the proposed control strategy is shown in Figure 3.

Air flow control and thermal-aware workload scheduling concepts rely on real-time temperature information collected in the data center. High reliability, frequent sampling, and low latency are key requirements for wireless sensor networks used in real-time control applications.

3 Data Center Wireless Sensor Network

The data center environment and the specific applications described in Section 2 lead to some challenging requirements for the WSN:

- Since sensor nodes may be added, removed or repositioned frequently because of equipment upgrades in the data center, the network should require as little configuration effort as possible.
- For easy deployment, the sensors need to be battery-powered and the WSN has to implement a power-efficient protocol stack to ensure long battery life.
- As accurate capturing of the temperature field in a data center typically requires a large number of monitoring points, the network must scale for large numbers of sensor nodes. In particular, the nodes may be densely deployed and thus strongly interfere with each other.

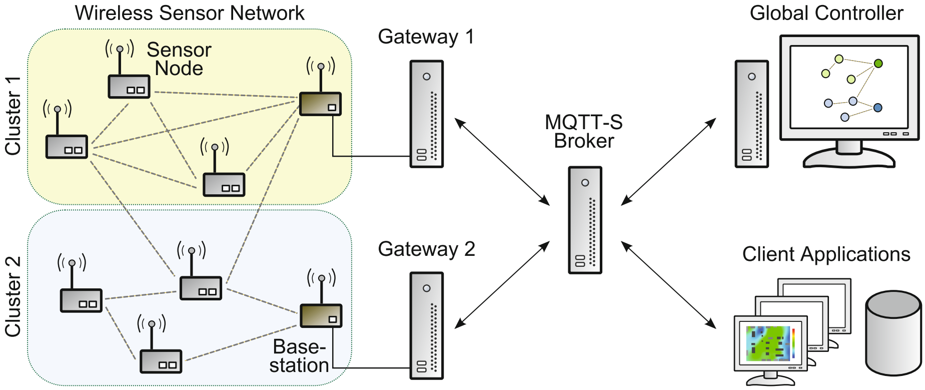


Fig. 4. Data Center Wireless Sensor Network with two clusters

- Because numerous metallic obstacles, such as racks, cabling shafts and piping, may disturb wireless transmission in the data center and consequently not all nodes may be able to reach the sink directly, a multi-hop network is needed.
- As temperature data will typically be used by several applications concurrently, they have to be forwarded to several distributed clients by a robust messaging mechanism.
- Some application require updated temperature information in less than every 10 seconds. In particular, air flow control and thermal-aware workload scheduling require frequent sampling and low latency.

In this section, we present the ZRL Data Center Center Wireless Sensor Network (DCWSN) for continuous temperature monitoring in large-scale data centers. Battery-powered temperature sensor nodes, equipped with an IEEE 802.15.4 compatible radio transceiver, are placed at key locations in the data center to periodically measure the local temperature. The architecture of the DCWSN is shown in Figure 4. It uses the so-called IMPERIA protocol stack, which implements a centrally managed low-power multi-hop wireless network. The global controller partitions the sensor nodes into one or multiple clusters based on network size and topology. Each cluster requires one permanently installed basestation associated with a gateway. The gateways collect the sensor data from the nodes within their cluster and send the data to a broker. Client applications subscribe to the data on the broker, which then forwards the temperature values to the subscribers. Global controller, gateways, broker and client applications communicate with each other by using the topic-based publish/subscribe messaging protocol MQTT-S.

The DCWSN operates in three modes: management mode, data collection mode, and sleep mode. In management mode, the radio transceivers of the sensor nodes are continuously enabled. This mode is used for network topology discovery, link quality assessment, and sensor node configuration. In data collection mode, the sensor nodes only enable their radio transceivers within their assigned slots according to a time division multiple access (TDMA) based schedule to either send or receive data for transmitting collected temperature data and maintaining synchronization. If connectivity with the basestation is lost for an extended time period, the sensor nodes automatically switch to sleep mode to save battery power. In sleep mode, the sensor nodes wake up periodically to listen for messages for a short time period.

3.1 Management Mode: Network Discovery and Configuration

All steps for network discovery and configuration are initiated by the global controller, which selects one of the available gateways to perform the required tasks and report back the collected information to the global controller. The wireless nodes use a source routing algorithm and a carrier sense multiple access (CSMA) protocol to transmit the messages to their destination. If some sensor nodes are in sleep mode, the gateway first instructs the basestation to broadcast a series of wake up messages. Sensor nodes receiving such a message, broadcast wake up messages themselves and then switch to management mode. To set up the network for data collection, the following steps are performed:

1. **Network Discovery** – The gateway instructs the basestation to broadcast a series of neighbor discovery messages, collect the identifiers of the responding sensor nodes, and report them to the gateway. The gateway then iteratively instructs all newly discovered nodes, one at a time, to find and report their own neighbors.
2. **Link Probing** – The gateway successively instructs each node to broadcast a number of link probe messages. The neighboring nodes count the received messages and record link performance indicators. This information is then collected by the gateway.
3. **Clustering and Routing** – The global controller uses a combined clustering and routing algorithm to identify for each sensor node the route with minimal expected number of transmissions to one of the basestations and assign it to the corresponding cluster. To avoid transient links that only temporarily have a good packet reception rate (PRR), the link quality is calculated based on measured PRR and received signal strength indicator.
4. **Scheduling** – The gateway determines the TDMA-based schedule for the data collection mode based on the routing trees calculated in step 3, the size of the messages, and the size of the message buffer at each node. This includes selecting the main basestation and defining the broadcast tree for the network wide synchronization.
5. **Sensor Node Configuration** – The basestation is instructed to transmit the configuration data provided by the global controller to the individual sensors nodes. The configuration data contains the individual slot numbers for sending and receiving and the address of the parent node.

After all sensor nodes have been configured, the basestation can be instructed to switch to data collection mode and broadcast a first synchronization beacon. Each sensor receiving this beacon switches to data collection mode.

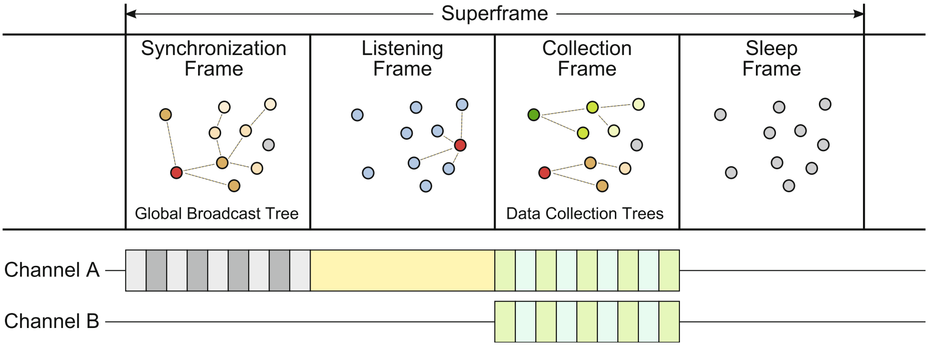


Fig. 5. Superframe structure used in data collection mode for a network with two clusters: Each cluster uses an individual communication channel for the collection frame. Messages for network management and synchronization beacons are always transmitted on channel A.

3.2 Data Collection Mode

In data collection mode, the sensor nodes transmit and receive messages according to their assigned TDMA-based schedule. The superframe structure illustrated in Figure 5 is periodically repeated to synchronize the wireless sensor network and deliver updated sensor data to the broker:

1. **Synchronization Frame** – Synchronization beacons are broadcast via the network wide broadcast tree to ensure that all sensor nodes will start the subsequent TDMA slots at the same time. The broadcast tree is rooted at the main basestation and each parent node is assigned a slot to broadcast its synchronization beacon to all of its children. The synchronization beacons include additional flags, e.g. for enabling the optional listening frame or for requesting status reports.
2. **Listening Frame** – The listening frame, which is enabled for one superframe if the corresponding flag is set, may be used to search for newly added or lost sensor nodes.
3. **Collection Frame** – The sensor nodes forward their data along the collection tree to the basestation of their cluster. Each cluster uses an individual communication channel such that the data transmissions can be performed in parallel. Within each cluster, only one node is sending and one node is receiving at a time.
4. **Sleep Frame** – The radio transceivers of all sensor nodes are turned off to save energy.

4 Measurement Results

We have successfully deployed the ZRL Data Center Wireless Sensor Network in production data centers. This section describes measurement results of a temporary deployment in a data center with a raised-floor area of about 2200 m^2.

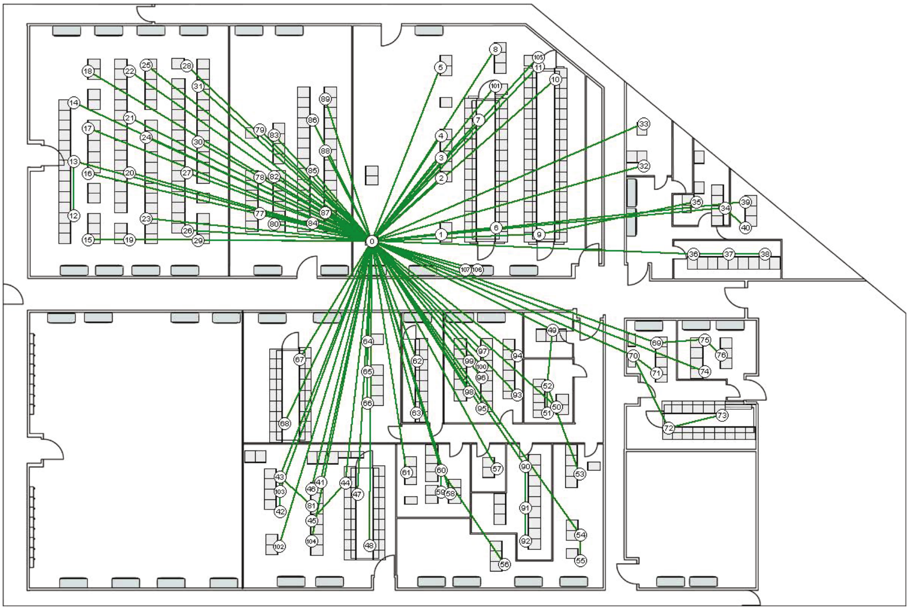


Fig. 6. Routing tree used for the DCWSN deployed in a production data center with 400 racks and 40 CRACs: The network has mostly a star topology

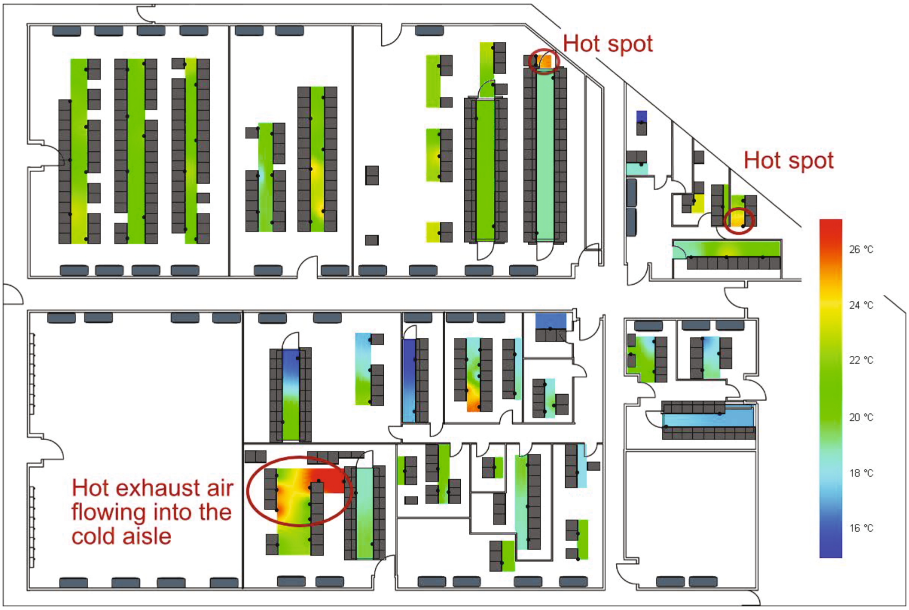


Fig. 7. The temperature map of the cold aisles in the data center indicates several potentially harmful hot spots at the air inlets of data processing equipment

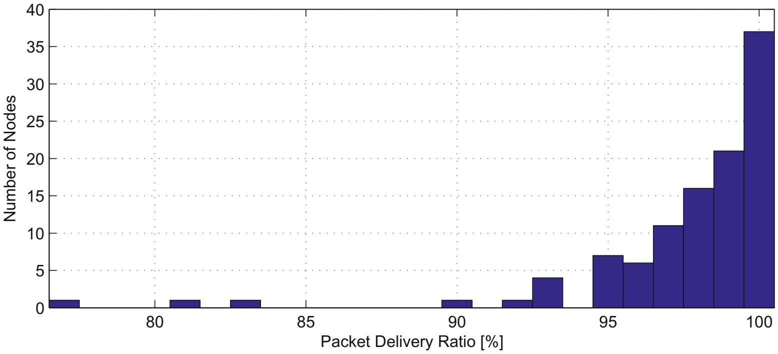


Fig. 8. Histogram of the packet delivery ratio

The data center houses 400 racks with heterogeneous data processing equipment and is cooled by 40 computer room air conditioners. The temperature changes in the cold aisles of the data center were tracked with 108 sensors during an upgrade of the cooling system. Since the regular data center operation could not be interrupted during the upgrade, continuous monitoring of the temperature distribution in the cold aisles of the data center was very important to timely detect potentially harmful temperature increases at the server air inlets, and immediately combat them by locally improving the air flow and temperature distribution. The sensor nodes were attached to the front side of the racks at a height of 1.6 m to measure the inlet air temperatures of the data processing equipment.

After placing the sensor nodes and testing network connectivity, management tasks and monitoring were done remotely. The DCWSN includes a client application for real-time temperature monitoring, network state information and network configuration. A web interface provides access to real-time temperature maps, temperature curves, and archived temperature data. It also gives an overview of the current network state and allows monitoring the battery voltages of the individual sensor nodes.

4.1 Network Performance

The DCWSN was set up with a single cluster. The basestation, placed at a central location in the data center, was equipped with an antenna providing a 3 dB stronger gain than the ones used by the sensor nodes. The network topology used for data collection is shown in Figure 6, where most of the sensor nodes send their data directly to the gateway and just a few sensor nodes require relay nodes. The packet delivery ratio (PDR), given by the number of messages successfully received by the gateway divided by the number of messages transmitted by a sensor node, is illustrated in the histogram in Figure 8. The mean PDR was 97.63 %, and only 3 sensor nodes had a PDR lower than 90 %. This is achieved thanks to the TDMA-based schedule used in data collection mode, where only

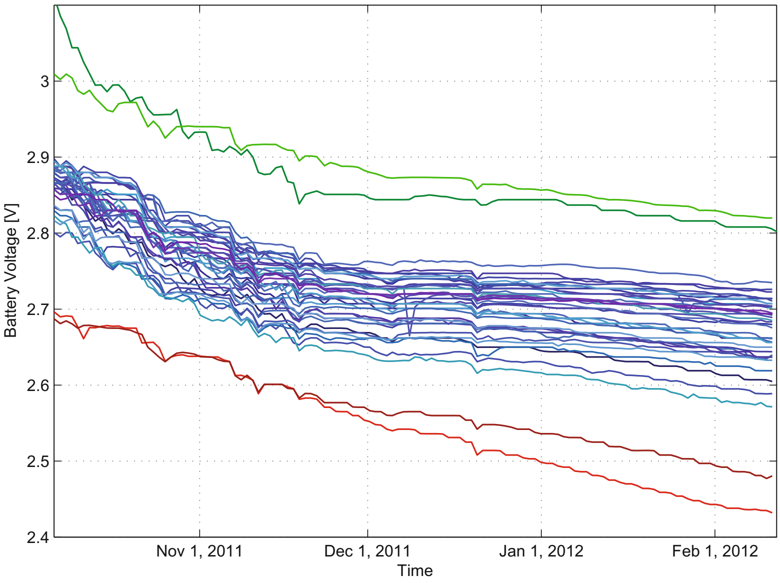


Fig. 9. Battery voltage of sensor nodes

one node is allowed to send at a time. Since most of the sensor nodes deployed in the data center were within transmission range of each other, a CSMA-based approach would suffer from a high collision probability and could thus not perform nearly as well.

Each sensor node is powered by two AA alkaline batteries. The power consumption of the DCWSN is traced by measuring the battery voltage of the individual nodes. Figure 9 shows the measured battery voltages for a cluster with 40 nodes deployed in another data center. This DCWSN has been collecting data for 130 days, transmitting updated temperature measurements every 10 seconds. At the start, two sensor nodes (green) were equipped with new batteries, while the other nodes (blue and red) started with battery voltages below 2.9 V. Concatenating a green, a blue and a red curve shown in Figure 9 covers a time frame of 390 days. Based on these results, we expect that the DCWSN can be operated for more than 12 months without exchanging batteries. Considerably longer battery lifetimes could be reached by increasing the interval between two temperature measurements from 10 seconds to 30 seconds or several minutes, depending on the requirements of the application.

4.2 Temperature Measurements

Figure 7 shows an interpolated temperature map of the cold aisles in the data center. The temperatures measured in the cold aisles at the air inlets of data processing equipment are mostly below 25 °C, but the temperature map also indicates several isolated hot spots due to insufficient cooling. Adding new

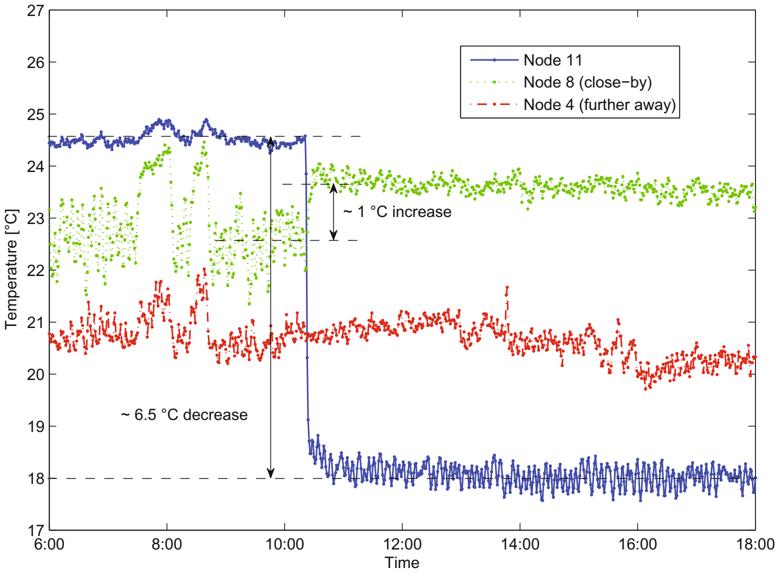


Fig. 10. Temperature measurement during replacement of a perforated floor tile: While the temperature at sensor node 11 decreased by 6.5 °C, sensor node 8 measured a temperature increase of 1 °C. Sensor node 4 did not measure a temperature increase.

perforated floor tiles or replacing existing ones by floor tiles with bigger open area was an easy way to provide more cold air to areas with critical temperature, given that the pressure level in the air supply plenum below the raised-floor was sufficiently high in this data center. For example, the perforated floor tile in front of node 11 was exchanged by one with bigger open area. This yielded a reduction of 6.5 °C of the air inlet temperature at the top of the rack. The overall room temperature was not affected by this action. A detailed view of the temperature measurements at the time of the floor tile replacement is shown Figure 10. While the temperature at node 11 decreased by 6.5 °C, node 8, which was attached to a rack in a neighboring aisle, measured a slight temperature increase. This can be explained by the pressure drop in that particular area of the air supply plenum due to the higher air flow through the new perforated floor tile. Node 4, which was attached to a rack further away from the new perforated floor tile, did not sense a temperature increase.

All hot spots highlighted in Figure 7 were eliminated as described in the example discussed above by adding and replacing perforated floor tiles to optimize cold air supply and by rearranging some racks to prevent hot exhaust air from streaming into the cold aisles. This allowed increasing the reference temperature of the computer room air conditioning units in the data center by 3 °C, thus achieving a significant cooling energy reduction without risking device overheating.

5 Conclusion

Wireless sensor networks offer a low-cost non-intrusive solution to gather temperature information in data centers. The sensors can be quickly deployed and easily repositioned if data processing equipment in the data center is relocated or replaced. Applications include continuous temperature monitoring, data collection for thermal modeling and temperature sensing for real-time control.

Continuous temperature monitoring is essential to prevent device overheating while operating the cooling system close to the upper temperature limit for increased energy efficiency. We propose using the ZRL Data Center Wireless Sensor Network to capture temperature data at key locations in the data center with low-cost battery-powered sensors. The DCWSN forwards the captured information from the sensors to a monitoring client by executing the IMPERIA/MQTT-S protocol stack. This allows continuous monitoring of the temperature in thermally critical areas of the data center, visualizing the temperature field in real-time, and triggering alarms if a thermal problem occurs. Archived data can be used to analyze temperature trends and to build thermal models for optimizing the cooling concept of the data center.

Deployments in production data centers have shown that the DCWSN performs well in terms of configuration effort, reliability, and power efficiency. Moreover, we have demonstrated in a data center with 400 racks that the cooling efficiency of a data center can be significantly increased by improving the air flow and temperature distribution based on measurement data from the DCWSN.

References

1. Koomey, J.: Growth in Data Center Electricity Use 2005 to 2010. Analytics Press (August 2011)
2. Weiss, B., Truong, H.L., Schott, W., Munari, A., Lombriser, C., Hunkeler, U., Chevillat, P.: A Power-Efficient Wireless Sensor Network for Continuously Monitoring Seismic Vibrations. In: 8th IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks, SECON (2011)
3. Hunkeler, U., Truong, H.L., Stanford-Clark, A.: MQTT-S: A Publish/Subscribe Protocol for Wireless Sensor Networks. In: Proceedings of the Workshop on Information Assurance for Middleware Communications, IAMCOM (2008)
4. Tang, Q., Mukherjee, T., Gupta, S.K.S., Cayton, P.: Sensor-Based Fast Thermal Evaluation Model for Energy Efficient High-Performance Datacenters. In: Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP), pp. 203–208 (2006)
5. Samadiani, E., Joshi, Y.: Proper Orthogonal Decomposition for Reduced Order Thermal Modeling of Air Cooled Data Centers. Journal of Heat Transfer 132(7) (2010) 071402.1–071402.14
6. Biller, P., Chevillat, P., de Lorenzi, F., Scherer, T., Schott, W., Ullmann, R., Vömel, C.: Efficient cooling of data centers. In: Proceedings of the 4th World Engineer's Convention (WEC), Geneva, Switzerland (September 2011)
7. Vasic, N., Scherer, T., Schott, W.: Thermal-aware Workload Scheduling for Energy Efficient Data Centers. In: Proceedings of the 7th International Conference on Autonomic Computing (ICAC), Washington, DC, USA, pp. 169–174 (June 2010)

Open Platform Semi-passive RFID Tag

Tzu Hao Li[1], Alexey Borisenko[2], and Miodrag Bolic[2]

[1] Research In Motion, Ottawa, On., Canada
[2] School of Electrical Engineering and Computer Sciences, University of Ottawa, On., Canada

Abstract. This paper presents the development of a prototype of a semi-passive Ultra-High Frequency (UHF) Radio Frequency Identification (RFID) tag that is compatible with the leading UHF RFID standard EPCGlobal Generation 2 Class 1. The design allows the addition of external analog and digital sensors and in that way the tag acts as a semi-passive wireless sensor node. A standard UHF RFID reader can acquire sensor data. The tag is designed as an open platform so that the firmware in the tag can be easily modified. Test results of our open platform semi-passive UHF RFID tag demonstrated that it can achieve a read rate above 50% when an open platform semi-passive UHF RFID tag is placed four meters from the reader antenna and the reader output power is set to 21 dBm.

1 Introduction

RFID is a rapidly emerging technology that enables automatic remote identification of objects. Passive and semi-passive RFID systems can be distinguished from other forms of wireless systems, because the RFID tags (transponders) communicate by way of backscatter. In addition, passive tags derive their energy from the RF signal emitted by the reader[1].

Semi-passive UHF RFID systems can provide much longer read range than passive RFID systems. In addition, some semi-passive tags contain sensors. However, the field of semi-passive RFID is still under development, and so far there are no open development platforms available.

In this paper, we designed a new semi-passive UHF RFID tag. The work involved creating a printed circuit board (PCB), firmware, and application software to meet the following objectives: the tag is compatible with EPCGlobal Class 1 Generation 2 standard [2], it is low power, and allows the addition of sensors.

The open platform semi-passive UHF RFID tag has been designed to provide a friendly development environment for researchers. The RFID open platform tag will create new experimentation opportunities, and enable research advances in several areas including: physical and link layer protocols, RFID sensor networks, security and privacy, and antenna design. All of that is possible since the code running on the microcontroller of the semi-passive tag can be easily modified.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 249–259, 2012.

We demonstrated two applications in this paper. In the first one, the temperature sensor is attached to the tag and the tag was used as an RFID sensor node. In the second application, another antenna was added to the tag in order to mitigate the effects of deep fading.

In Section 2, we present related work. Hardware and software design of an open platform tag was shown in Section 3. Experimental results are shown in Section 4. They include performance characterization of the semi-passive tag such as read rates and power consumption as well as the above mentioned two applications. The paper is concluded in Section 5.

2 Related Work

Table 1 illustrates some of the short range wireless technologies and their corresponding products. Every technology has its advantages and weak points. For example, WISP has an unlimited device life time, but its read range is maximum three meters, while the life of Wireless Sensor Networks (WSN), active and semi-passive RFID systems devices is limited by the battery.

Table 1. Wireless sensor system devices

	WISP	TelosB	Identec I-Q310	TU Graz
Wireless technology	Passive UHF RFID	ZigBee	Active RFID	Semi-passive UHF RFID
Protocol	EPCglobal C1 G2	ZigBee	ISO18000-7	EPCglobal C1 G2
Power consumption	< 1 mA	25 mA	17 - 20 mA	38 mA
Maximum range (m)	3	50	100	15
Network topology	Star	Star and Mesh	Star and broadcast	Star
Anti-collision	Slotted Aloha	Contention based CSMA/CA contention free GTS	ISO18000-7	Slotted Aloha
Unique naming	ONS and when manufactured	none	ONS and when manufactured	ONS and when manufactured
Frequency	902-928MHz	2.4GHz	433MHz	860-928MHz
Lifetime	Unlimited	Battery life cycle	Battery life cycle	Battery life cycle
Max Symbol Rate	64Kbps	62.5Kbps	27.7Kbps	100Kbps (FM0)
Extensibility	Open Platform	Open Platform	No	Limited

Several research efforts have focused on the hardware development of programmable RFID readers and tags. Angerer et. al. [3] presented a dual frequency RFID reader testbed that allows some modification of reader functionality. Ying [4] developed a verification platform for a Gen 2 RFID reader based on a soft-core processor residing within an FPGA. Buettner et. al. [5] developed a GNU radio based monitor for protocol and communication analysis of Gen 2 RFID systems. Modifiable RFID tag hardware platforms with potential for incorporation of sensors are presented in [6], [7].

One of the existing passive RFID systems is WISP designed by the Intel Research group. It is an open platform RFID programmable passive sensor tag [7]. WISP contains an on-board microcontroller and digital sensors. WISP was designed as an EPCglobal Gen 2 Class 1 passive RFID tag. However, the trade-off of battery-less design is less operating range; WISP's read range is about ten feet (three meters). In most wireless sensor networks, this operating range is not acceptable. Also, due to very low processing power, the WISP firmware can not handle advanced signal processing or complex algorithms.

TU Graz UHF Demotag designed by IAIK is a semi-passive re-programmable RFID tag. It contains an on-board microcontroller and supports the EPC-global Gen 2 Class 1 protocol. As it is semi-passive, TU Graz has an on-board power source to power the microcontroller. This provides longer operating range, greater processing ability, and extension support, (e.g. security or additional peripherals) compared to passive UHF RFID systems. The TU Graz tag is mainly designed for security functionality as it requires high speed processing and consumes 38 mA. Complete source code is not provided.

Active RFID is another RFID technology that can be applied in wireless sensor networks. It has a battery and an RF transmitter, and provides the longest transmission range compared to passive and semi-passive RFID tags. IdentecI-Q310 is one of the existing products that adapted active RFID technology. It is ISO 18000-7 standard compliant. As it contains an RF transmitter, IdentecI-Q310 consumes more power than semi-passive tags (approximately 17 to 20 mA in working mode), and requires extra 0 dBm power for transmitting purposes. It is not a totally open platform, so the user does not have full access to the source code.

TelosB wireless mote [9], developed by the University of California at Berkely is a successful open source platform. It is used in wireless sensor network applications. TelosB mote has an IEEE 802.15 radio with an integrated antenna. Its block diagram, schematics, and source code were published for the research community. TelosB mote is provided as an open platform for wireless sensor networks, and researchers can study and modify both the software and hardware design to meet their requirements. This open source platform helps to develop and evaluate wireless sensor network technology. Our goal with a semi-passive RFID tag is to provide an open UHF RFID platform to researchers.

3 Design

3.1 Hardware

Figure 1 shows the high level overall architecture of the open platform semi-passive UHF RFID tag. The RF signal received by the antenna is fed through a bandpass filter circuit to an envelope detector, which removes the carrier signal and extracts the baseband component. As the received signal strength can be quite low, a Dickson charge pump is used. It comprises of multiple stages of pairs of diodes connected in series to increase the voltage level of the signal. The power efficiency improves by adding up to five stages to the Dickson charge pump, which was determined to be the optimal arrangement for power efficiency [10].

The baseband analog signal is fed to a hysteresis comparator that compares the signal with its low pass version, then generates a digital output that contains the encoded information received by the tag. This signal is processed by the digital section that performs much of the MAC layer protocol activity and higher layer functionality as needed. On the transmit side, the information to be conveyed from the tag is appropriately encoded in the digital section and used to control the backscatter modulator that encodes the information to the signal reflected from the tag antenna.

The RF switch is another key component in our open platform UHF RFID tag. The switch operates in the 902 to 928 MHz frequency band with low insertion loss while in receive mode (switch on). At the same time the RF switch should backscatter the input RF power as much as possible when the switch is off.

We use Microchip PIC24FJ64GA004 as a microcontroller due to its low cost, extension support for different applications, the availability of low cost or no cost development tools, and its serial programming capability (and re-programming with flash memory).

The microcontroller will manage the EPCglobal Class 1 Gen 2 protocol, decoding the reader signal and encoding the tag backscatter signal. The RFID tag only supports a backscattering signal rate of 256KHz with Miller 4 encoding due to the microcontroller working frequency. As the working frequency is proportional to the power consumption [11], we must decrease the frequency as much as possible to reduce the overall power consumption. Our experiments found that the minimum frequency we can achieve that supports this backscattering rate is 4 million instructions per second which is 8 MHz system clock. Even though we target this frequency, researchers can adjust the system clock to meet their project requirement. The PIC24FJ64 family has two ADC and I^2C interface modules. Analog and digital sensors can be attached to the RFID tag. The RFID tag can use I^2C to communicate with other digital components, including memory or custom digital devices. ADC can be used to communicate with low frequency analog components as well.

Figure 2 shows the first prototype of an open platform semi-passive UHF RFID tag.

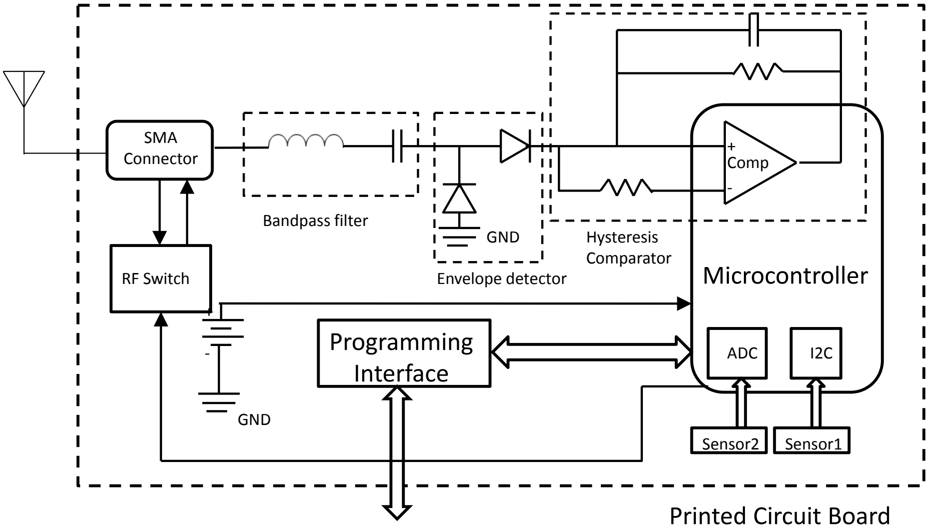


Fig. 1. Overall architecture of proposed tag platform

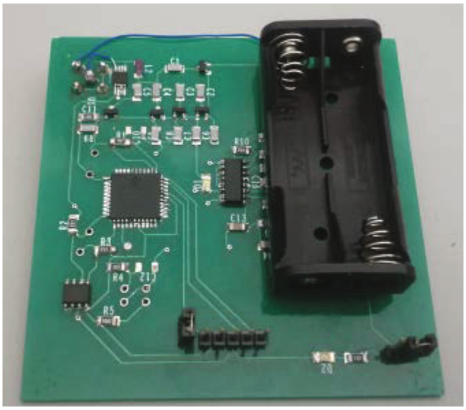


Fig. 2. Prototype of an open platform semi-passive RFID tag

3.2 Software

The firmware is developed in C and Assembly languages using the MPLAB Integrated Development Environment (IDE), which is a free, integrated gcc-based toolset for the development of embedded applications employing Microchip's PIC and dsPIC microcontrollers [12].

The firmware modules can be classified as:

1. Baseband Decoder – pulse interval encoding (PIE) decoder.
2. Baseband Encoder – Miller-4 (M4) encoder.

3. EPCglobal Gen 2 Class 1 state machine – main state machine that incorporates EPCglobal Gen 2 Class 1 standard.
4. Custom Sub Function – set of modules defined by user.

The RFID reader software is composed of a low-level reader protocol and a graphic user interface. Low-level reader protocol (LLRP) is used to control RFID air protocol operation timing and access to air protocol command parameters [13]. LLRP is a specification for the network interface between the reader and its controlling software or hardware.

4 Results and Discussion

4.1 Read Rate and Read Range

We first performed several experiments related to read rate and Received Signal Strength (RSS). Read rate is defined as the number of responses from the tag divided by the number of reader's queries sent. RSS illustrates the tag sensitivity and read range. If RSS is too low, the tag will not be able to process the signal properly for forward link, and the reader cannot decode tag backscattering signals for backward link. Both parameters can help evaluate the performance of an open platform semi-passive UHF RFID tag in a computer lab. The experiments were performed in a computer lab with dimensions of ten meters long, ten meters wide and three meters high.

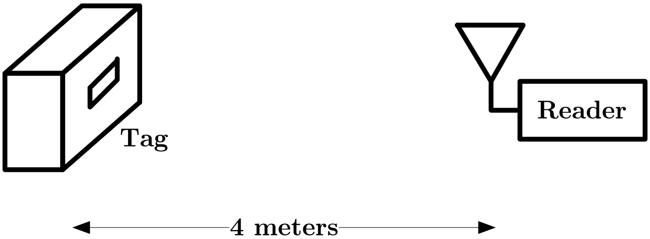


Fig. 3. Experimental setup for examining tag read rate

In order to examine the performance of our open platform semi-passive RFID tag, we used an experimental setup consisting of an EPCglobal Gen 2 Class 1 reader with a 6dBi gain circularly polarized patch antenna, a commercial semi-passive tag, and an open platform semi-passive RFID tag with a 3dBi linearly polarized patch antenna. The tags were placed four meters from reader antenna, and the reader output power was changed in fixed steps in the range from 20 to 30 dBm, so that when the reader transmitting power is less than 20 dBm, the open platform semi-passive UHF RFID tag read rate drops to zero, and 30 dBm is the maximum allowable reader transmitting power. We placed the tag four meters away from the reader antenna, as this distance provides the best read rate for both tags in the computer lab. The open platform semi-passive RFID tag and commercial semi-passive tag are placed in a plane on a single cardboard platform with the best possible orientation angle relative to the reader antenna. This is done to eliminate the influence of orientation sensitivity on the measurements. The experimental setup is shown in Figure 3. Initially, the reader output power was set at 20 dBm. A total of 1000 query rounds were sent by the reader, and the number of responses from the tag and average of RSSi value from all query rounds were noted. RSSi values were collected only when the reader was able to process the tag responding signal. We only test one tag at time, to eliminate the influence of shadowing effects [14]. The read rate results are shown in Figure 4, and the RSSi results in Figure 5.

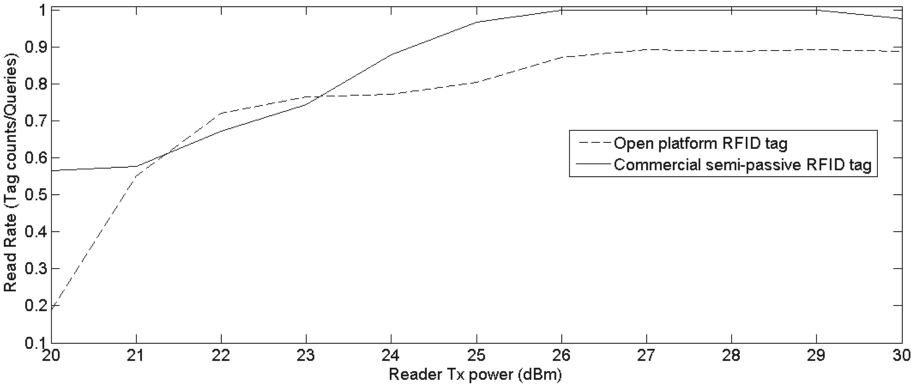


Fig. 4. Read rate of open platform semi-passive tag and commercial semi-passive RFID tag in a Computer lab

From the results, we also observed several issues for this prototype of an open platform semi-passive UHF RFID tag. Figure 4 shows the best read rate of the tag is approximately 90%. There are several reasons for this: the comparator did not digitize the signal properly due to analog baseband signals that were too weak after the low pass filter, and the tag input impedance was mismatched so the analog baseband signals were very weak at certain frequencies. These issues reduced the performance of the open platform semi-passive RFID tag. Another observation shown in Figure 4 illustrates the sensitivity of the open platform semi-passive RFID tag. When the reader transmit signal power is less than 21 dBm, and reader-tag distance is around four meters, the read rate of the tag drops exponentially. Figure 5 shows that when the RSSi falls below -87 dBm, the reader can no longer detect the tag.

In Figure 4, the read rate of the open platform semi-passive UHF RFID tag and a commercially available semi-passive tag drop to around zero at certain points, and the read rate varies, regardless of reader-tag separation. This is due

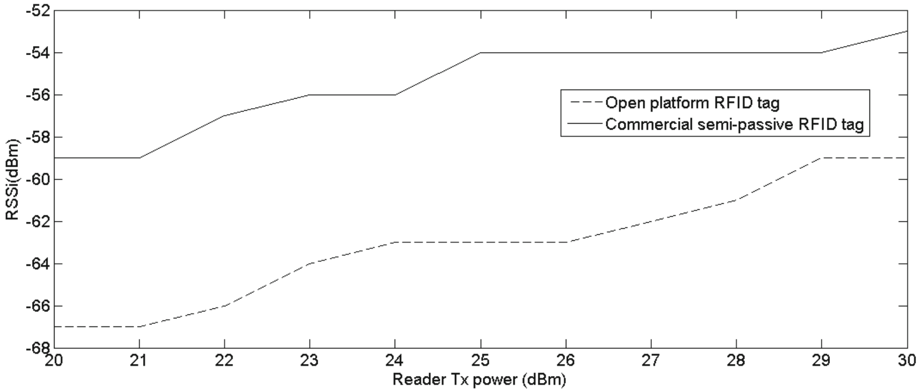


Fig. 5. RSSi of open platform semi-passive tag and commercial semi-passive RFID tag in a Computer lab

to null reading points (deep fading) [15]. At null reading point, the reader signal and reflected signals cancel each other out. When we increase the reader-tag separation, the tag is no longer in null reading points, so the read rate returns back to normal.

4.2 Improving Tag Operation by Using Two Antennas

The problem of deep fading is addressed by adding a second antenna to the open-platform tag. For this experiment the open platform tag had two antennas: a 3dBi patch antenna and 2.5dBi loop antenna. The Impinj UHF RFID reader with a 8dBi patch antenna was used as the interrogator. The reader parameters were set to Dense Reader Mode Miller-4 with 30dBm output power. The reader antenna is placed 1 meter above ground, and tag about 75 cm above ground. Figure 6 shows the RSSi obtained at the reader side vs. tag distance. As can be seen from the figure, the reader was able to read the tag with two antennas even in cases when the tag with one antenna was at a null point.

4.3 Backscattering Power Consumption

In this experiment, we measured the current vs input voltage when the open platform semi-passive UHF RFID tag is communicating with the reader, and when it is outside of the read zone of the reader. Table 2 shows the current measurement results for various input voltages.

Limiting the input voltage to 2V does not affect read rate and RSSi of the open platform semi-passive RFID tag and also saves considerable power which increases device life cycle. Table 2 shows us that the backscattering mode does not drain more current than non-backscattering mode. Thus, the semi-passive open platform RFID tag does not waste energy when transmitting data.

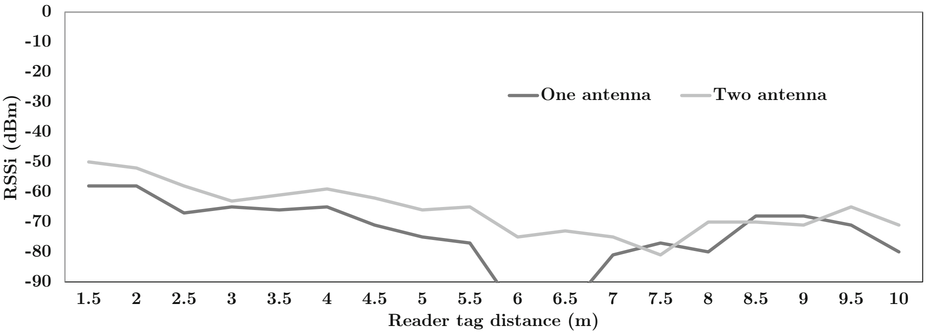


Fig. 6. RSSi versus distance of a tag with one and a tag with two antennas in a Computer lab

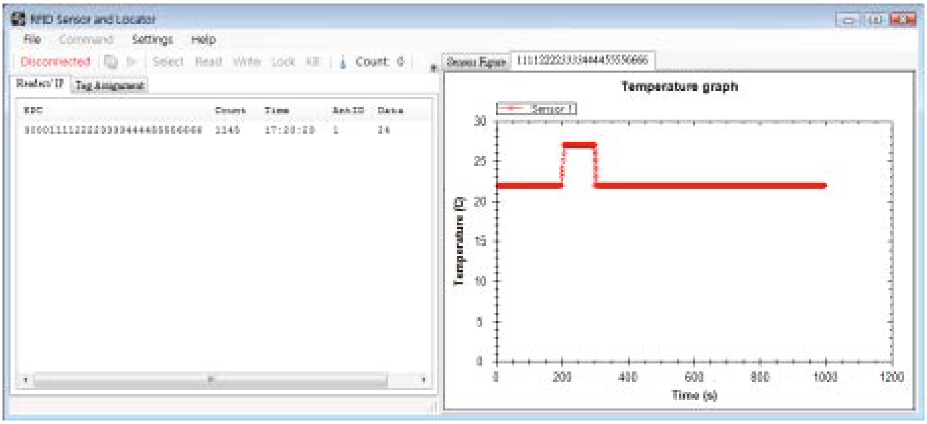
Table 2. Power consumption in executing mode and idle mode

Voltage (V)	Backscattering Mode (mA)	No backscattering mode (mA)
3	15.32	15.32
2.75	11.76	11.76
2.5	9.06	9.06
2.25	6.46	6.45
2	5.0	4.9

4.4 Application of the Open Platform Semi-passive RFID Sensor Network

In this section, we integrated a digital temperature sensor [16] into the open platform semi-passive RFID tag. The tag uses I^2C protocol to communicate with the sensor. The original state diagram of EPCglobal Gen 2 Class 1 was modified to fit the sensor network requirement. When the tag first boots up it does a pre-CRC calculation. When the reader requests temperature data, the tag only needs to feed the temperature, instead of the whole response packet, to the CRC generator, thus saving time. A full CRC calculation requires 225 ns when tag system frequency is set to 8MHz. According to EPGglobal Gen 2 Class 1 [2], the tag needs to respond in approximately 40 ns when the backscattering signal rate is set to 256KHz with Miller 4 mode, thus the tag will not have enough time to calculate a full CRC during response.

On the GUI side, we added the target tag's EPC into sensor tag list. By doing this, the GUI will automatically request data from this tag. This is specific functionality we implemented for this application. We set the interval of each query round to one second. At the beginning of the experiment, we place the tag in the computer lab. During the experiment, we placed a finger on the digital sensor for several minutes, then removed it. The results, in Figure 7, show that the temperature increase in middle of the experiment, and then drops back to room temperature.



EPC	Count	Time	Ant.ID	Data
0000111122223333444455556666	1146	17:28:28	1	24

Fig. 7. GUI that represents temperature variations obtained by acquiring the sensor data from the open platform semi-passive tag

5 Conclusion

The paper presented the design and architecture of an open platform semi-passive RFID tag. We experimented with the performance characteristics of the open platform tag, including range, reliability, and power consumption. Range and reliability were compared to commercially available semi-passive tags. A temperature sensor was integrated to demonstrate the extensibility of the tag in both hardware and software. Read rates were improved by installing an additional antenna to the open platform tag.

References

1. Finkenzeller, K.: RFID Handbook, 2nd edn. Wiley (2003)
2. EPCglobal Inc, EPC radio frequency identification protocols class 1 generation 2 UHF RFID protocol for communications at 860 MHz 960 MHz (2008), http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf (accessed February 22, 2012)
3. Angerer, C., Holzer, M., Knerr, B., Rupp, M.: A flexible dual frequency testbed for RFID. In: Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, ICST (2008)
4. Ying, C.: A Verification Development Platform for UHF RFID Reader. In: WRI International Conference on Communications and Mobile Computing (2009)
5. Buettner, M., Wetherall, D.: A Gen 2 RFID monitor based on the USRP. ACM SIGCOMM Computer Communication Review 40(3), 41–47 (2010)
6. Jones, A.K., et al.: A field programmable RFID tag and associated design flow. In: 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 165–174 (2006)

7. Sample, A.P., et al.: Design of an rfid-based battery-free programmable sensing platform. IEEE Transactions on Instrumentation and Measurements 57(11) (2008)
8. EM Microelectronic, 1 kbit read/write, ISO 18000-6c epc c-1 g-2 passive / battery-assisted passive contactless ic (2012), `http://www.emmicroelectronic.com` (accessed February 22, 2012)
9. MEMSIC, TELOSB mote platform (2011), `http://memsic.com/` (accessed February 22, 2011)
10. Dobkin, D.M., Weigand, S.M.: Environmental effects on rfid tag antennas. In: Microwave Symposium Digest (2005)
11. Microchip Technology, PIC24FJ64GA004 family (2012), `http://ww1.microchip.com/downloads/en/DeviceDoc/39881D.pdf` (accessed February 22, 2012)
12. Microchip Technology, Mplab integrated development environment (2012), `http://www.microchip.com/mplab` (accessed February 22, 2012)
13. EPCglobal Inc., Low Level Reader Protocol (LLRP), Standard Specification version 1.1 (2010)
14. Dobkin, D.M.: The RF in RFID: Passive UHF RFID in Practice. Elsevier Newnes (2007)
15. Bolić, M., Athalye, A., Li, T.H.: Performance of passive UHF RFID systems in practice. In: RFID Systems: Research Trends and Challenges. Wiley (2010)
16. STMicroelectronics, Digital temperature sensor and thermal watchdog (2011), `http://www.st.com/` (accessed February 22, 2012)

Study of the Optimum Frequency at 2.4GHz ISM Band for Underwater Wireless Ad Hoc Communications

Sandra Sendra, Jose V. Lamparero, Jaime Lloret, and Miguel Ardid

Instituto de Investigación para la Gestión Integrada de zonas Costeras
Universidad Politécnica de Valencia
46730 Grao de Gandia (Valencia), Spain
sansenco@posgrado.upv.es, josevi@lamparero.es,
jlloret@dcom.upv.es, mardid@fis.upv.es

Abstract. Underwater communications at low frequencies are characterized by the low data rate. But in some cases wireless sensors must be placed quite close to each other and need high data rates in order to accurately sense an ecosystem that could be contaminated by invasive plants or hazardous waste. Most researchers focus their efforts on increasing the data transfer rates for low frequencies, but, due to the wave features, this is very complicated. For this reason, we propose the use of high frequency band communications for these special cases. In this paper we measure the optimum working frequency for an underwater communication in the 2.4 GHz range. We measure the number of lost packets and the average round trip time value for a point-to-point link for different distances. These measures will be performed by varying the data rate, the type of modulation and the working frequency. We will show that we are able to transmit higher data transfer rates, by using higher frequencies, than the using acoustic waves.

Keywords: Underwater Wireless Ad Hoc Communications, 2.4 GHz, UWSN.

1 Introduction

Research related to underwater communications and ad-hoc networks are growing rapidly. One of the main research lines that are being studied, in ad-hoc networks, is the increase of the network lifetime [1, 2]. When we try to implement ad hoc underwater network, we encounter other problems, such as, the low performance of underwater communication systems.

Communication systems based on optical waves and acoustic techniques are being used in wireless communication deployments for underwater environments. But both transmission systems have advantages and disadvantages [3]. On one hand, the systems that are able to reach very high propagation speed are those based on optical communication. However, due to the suspended particles and the turbidity of the water, this system presents a strong backscattering, so it is not good option for long distances. On the other hand, systems based on acoustic waves are not so sensible to suspended particles and turbidity of water. Low frequencies are used in these kinds of systems, so there are problems with latency. Moreover, there is a low data rate.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 260–273, 2012.

Electromagnetic (EM) waves, in the RF range, can also be used for underwater wireless communication systems as a good option. These waves are less sensitive to reflection and refraction effects in shallow water, than acoustic waves. Moreover, suspended particles have very little impact on them. The speed of EM waves in the water is 2,25x10^8 m/s, meanwhile the speed of acoustic waves is around 1500 m/s. This parameter depends mainly on 4 environmental factors, which are: permeability (μ), permittivity (ε), conductivity (σ) and volume charge density ρ [4]. But there are some effects that can change the water nature. The wave propagation speed and absorption coefficient vary as a function of the presence of dissolved salts in water, which changes the electrical conductivity value associated to the medium. The conductivity is directly related to the working frequency. Conductivity presents different values for each case. Seawater has a conductivity average value around 4 S/m (this value changes depending on the tested sea), but in fresh water the typical value is 0.01 S/m (400 times less) and drinking water presents a conductivity around 0.005-0.05 S/m. References [5] and [6] show a relationship model that relates the changes of the frequency with the temperature, the salinity, and the permittivity of the seawater. Thus, the main problem of underwater communication with EM waves is the high attenuation, due to the conductivity of the water, and its increase when the frequency of EM waves increases. For this reason, the higher frequencies always register higher attenuation losses. Considering all these factors, we performed a practical study in underwater environments. We tested the behavior of EM signals in this medium. In order to perform it we used devices compatible with IEEE 802.11 standard [7].

This paper addresses the tests performed at different frequencies and modulations in order to check various parameters such as minimum depth, distance between devices and signal transmission characteristics. Tests have been performed in the first seven frequencies (specified in the IEEE 802.11 standard), that correspond with the frequency range from 2.412 GHz to 2.442 GHz. We performed and ad hoc communication between two devices, a Personal Computer (PC) and an access point (AP), in order to monitor the activity of the underwater point-to-point link. We have used the echo request and echo reply packets in order to perform our tests. The high attenuation given at these frequencies leaded us to think that underwater communications at 2.4 GHz band is unhelpful and impractical, but as we shall see at the end of this paper, there are many applications where the use of this technology will bring many benefits.

The rest of the paper is structured as follows. We finish the first section, showing some previous work. Section 2 overviews some aspects about the used modulations and data rate of IEEE 802.11b/g. Section 3 describes the scenario, hardware and software used in order to take the right measurements. The performance results are presented in Section 4. Finally, Section 5 shows the conclusion and future work.

1.1 Related Work

The most widely used waves in underwater communications are the acoustic waves. There is a huge variety of articles, which describe and propose underwater communications systems. However it is not so common the description of underwater communications systems using acoustic waves.

In [8], Chaitanya et al. show an example of path loss analysis given by the reflection and refractions. Moreover, we can see the effects of depth and temperature in this type of waves [9]. For systems based on optical communications, we can also find a great variety of studies about their propagation and losses [10].

As far as we know, due to the limited use of EM in underwater environments, there is very few literature published about them. We can find some generic papers, where the authors show the mathematical formulation that should be taken into account when working with EM waves [11]. One of them is the paper authored by Jiang et al. in [12]. They conducted a study of the EM wave's propagation in fresh water for frequencies between 23 kHz and 1 GHz. They also analyzed other parameters that are related to the waves transmission speed.

In previous experiments [13], the authors of this paper performed a study on RF communication in the 2.4 GHz ISM frequency band. We demonstrated that it is a feasible option to use the EM waves in order to establish an underwater wireless link and to transmit high data bit rates between two devices. We performed several tests for 1, 2, 5.5 and 11 Mbps at different frequencies.

Except the last paper presented in this section, which is our paper, we have not found any other paper in the related literature showing the performance of underwater communication tests at 2.4 GHz.

2 Modulations and Data Rate Overview

This section shows the parameters taken into account in our measurements: the modulation type and the data rates. We have analyzed another technology that use the same frequency as IEEE 802.11 [7]. It is the IEEE 802.15.4 standard [14]. Moreover, because our tests were performed using commercial devices, operating under the IEEE 802.11 b/g, we also discuss the standard and identify each type of modulation with the data rates specified in the standard.

IEEE 802.11 standard defines the value of maximum data transfer rates depending on the used modulation. Each variant can be chosen depending on the system where it is going to be applied. In our experiments, we used the Phase-Shift Keying (PSK) [15] and the Complementary Code Keying (CCK) [16] modulations. CCK and PKS modulations operate at a theoretical data rates up to 11 Mbps in the range of 2.400 GHz to 2.4835 GHz. BPSK and QPSK modulations are optimal from the error protection point of view. BPSK is used for low-cost transmitters that do not require high speeds. CCK modulation allows encoding multiple bits of data directly on a single chip with eight 64-bit sequences. Therefore, CCK method can achieve a maximum speed of 5.5 Mbps by encoding 4 bits at a time or up to 11 Mbps by encoding 8 bits of data.

Fig. 1 shows a comparison of the maximum data transfer rates of both wireless technologies. It shows that devices that use IEEE 802.15.4 standard have much lower data transfer rates than IEEE 802.11 standard.

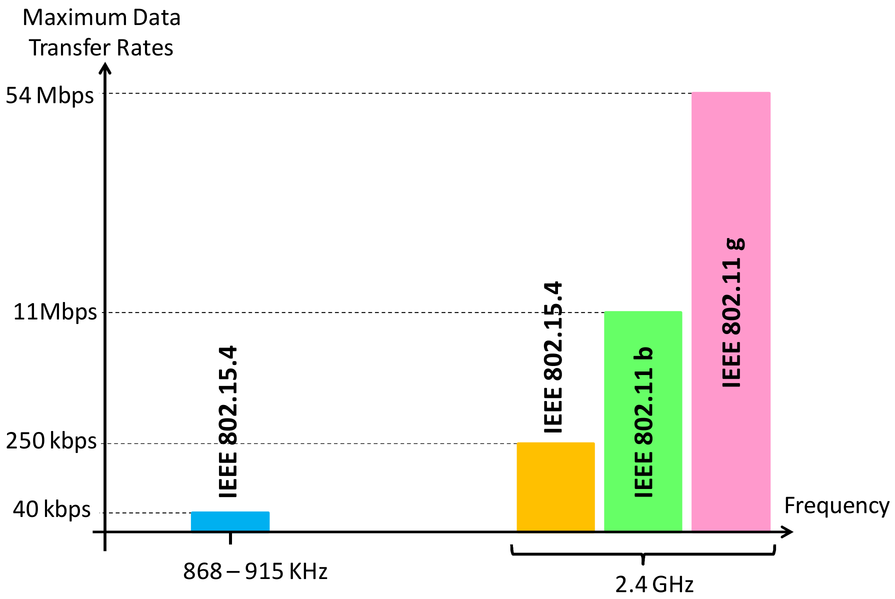


Fig. 1. Data transfer rates comparison of some wireless technologies

Although IEEE 802.15.4 has lower power consumption than the other technology, as we can see in figure 2, it also presents low data transfer rates. Our application needs data transfer rates higher than the ones offered by IEEE 802.14.5. For this reason, we have decided to sacrifice a little the power consumption in favor of enhancing the system data rates.

Table 1 identifies the used modulations and the maximum data rates for IEEE 802.11b/g variants. In order to take measurements, first we determined the distance between devices were the number of delivered packets without errors is higher than 50 % at least. We observed at 15.5 cm that the percentage of packets delivered successfully is quite high, while for 16 cm, these values begin to decrease. Then we measure the number of lost packets and the round trip time (RTT) value for each type of modulation and transfer rate for each frequency. We will also do the test for 17 cm, where these values are very low, as we saw in [13]. With these measures, we aim to see if varying the frequency and modulation scheme, we obtain better results. In order to determine which modulation and transmission schemes are good to be added in our tests, we performed some preliminary tests. We found that the OFDM transmission scheme presented worse behavior than the other three modulations (BPSK, QPSK and CCK). That is why we did not include it in our test. Table 1 shows data transfer rates for BPSK, QPSK and CCK.

Table 1. Modulations and data rates used in IEEE 802.11 b/g

Modulation	BPSK	QPSK	CCK	CCK
Data rates	1 Mbps	2 Mbps	5.5 Mbps	11 Mbps

3 Scenario, Hardware and Measurements Strategies

This section describes the scenario where measures have been taken and the hardware and software used for our tests. It also explains the preliminary performed tests.

3.1 Place to Take Measurements

We have placed the system in a swimming pool which has 32 m^2 surface (it as 8 meters length and 4 meters wide). It has a depth between 1.5 m and 1.80 m (depending on the side) and the brick walls are covered with small mosaic tiles. Because the swimming pool dimensions are much greater than the distance which the devices are located, we will avoid any reflection and refraction on the walls, ground and surface water (due to the change of medium). The measurements were taken in fresh water with a temperature of 26 °C. The pH value was 7.2 and the amount of chlorine and bromine dissolved in the water was 0.3 mg/l.

Fig. 2 shows the sketch of the swimming pool used to perform our measurements.

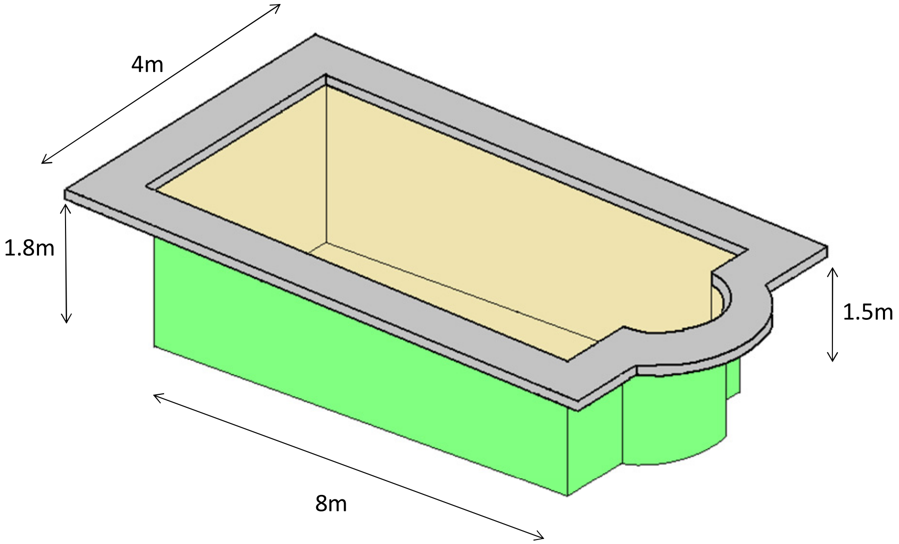


Fig. 2. Swimming pool where measures have been taken

3.2 Elements Used in the Tests

In order to perform the tests we used a wireless AP Dlink DWL-2100AP. This AP can work under IEEE 802.11b/g. It can be configured to work as a wireless AP, as a bridge for a point to point connection with another wireless bridge, as a bridge for a point-multipoint connection with another bridge or as a wireless client. Its output power is around the 16 dBm.

The AP uses a vertical monopole antenna with 2 dBi of gain. It is an antenna consisting of a single radiating arm straight vertically. This antenna has to be completed by a ground plane to operate properly. This ground plane can be natural (a water surface to facilitate electron conduction) or artificial (a number of drivers who are joined at the base of the monopole).

We also used a laptop (located outside the water) as a second device to monitor the wireless network from outside the water. In order to connect the antennae which are placed inside the water, to the devices which are outside, we used 2 pigtails of 3 meters.

Fig. 3 shows the topology of the test bench. It shows the AP, the computer and the two antennas inside the water.

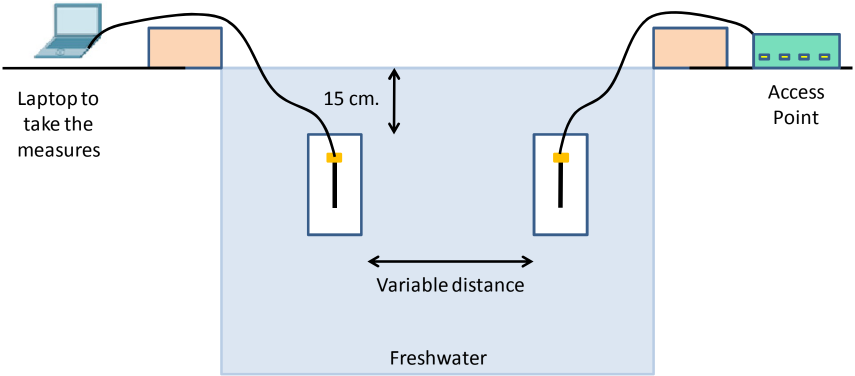


Fig. 3. Measurement setup

In order to take the measurements, we used the same method used to check the status of a network connection. Concretely, we used some common commands in the command-line shell interface. It let us sent a continuous packet flow, using the echo request and echo reply packets. Then, we collected and analyzed the obtained results. From these data, it is easy to extract the system performance in terms of communication distance, data transfer rate, average RTT and % of lost packets for each frequency. It is only needed a simple data processing.

3.3 Measurement Strategies and Scenario Preparation

First of all, it is important to ensure that the measurements taken are valid and the signal did not spread out of the water. Then, the first step was to determine the minimum depth where the antennae should be placed. We introduce the AP antenna in the water and we established an ad hoc wireless connection between the PC and the AP. When the laptop placed outside did not get any signal from antenna AP that was introduced in the water, we had obtained the minimum depth. We lost the signal from the AP when it is at 15 cm. deep. With this simple test, we were ensuring that the only signal received by the laptop is provided by the antenna placed inside the water.

In the second test, we check the effect of different power emissions. We tested different values of power, between 100 mW and 800 mW. Interestingly and contrary to what happens when we work with these devices in the air, increasing the transmission power, the maximum distance between devices, does not increase. In addition, we observed that the transmission behavior worsens. We therefore decided to work at 100 mW.

In the third preliminary test, we checked if the antenna emits when it is in contact directly with water. In this case, the antennas were sealed and plunged into the water. We observed that the antennas had to be very close to each other (almost touching). Therefore, we decide to put them in a watertight container, so the antenna could start emitting into the air and then the signal propagates through the water. We also tested the effect of container size of the antenna. By different studies on wireless signal propagation and path loss [17], we know that the greatest signal strength is found just one meter from the sending device. From this point, signal starts to decrease. We wanted to see if this is also repeated in the water. To do this, we take a container with a length of 1.5m and the antenna is situated inside, so that they had a 1 meter on the one side and 0.5 m on the other side. Both antennas were submerged in the water and we checked the maximum distance between the two antennas, without reducing the network performance. Several container sizes were checked and we saw that the performance does not improve, when distance between antennas increases. Finally, we used small containers of 5 cm in diameter and the distance is the same. Therefore, we conducted tests with small containers.

Several tests were conducted in the frequency range between 2.412 GHz and 2.472 GHz. These values correspond to the spectrum used by devices that work under the IEEE 802.11b/g. These tests allow us to characterize the behavior of an underwater communication, based on EM waves, which will allow high transfer data rates.

4 Performance Results

This section shows the obtained results. We have tested several frequencies specified in the IEEE 802.11 standard. These frequencies are 2.412 GHz, 2.417 GHz, 2.422 GHz, 2.427 GHz, 2.432 GHz, 2.437 GHz and 2.442 GHz. For higher frequencies the value of lost packets is around 90-100%, which is a very bad value for a communications system.

We analyzed the variation of the RTT between both devices, depending on the distance between the antennas. We also measured the amount of lost packets and the communication behavior, depending on the type of modulation. Each test was 3 minutes long. We distinguish two types of packets: packets successfully received and packets which were not received or were received wrong. For the second type of packets, we assigned the value of 3,000 ms. In this way, we denoted that no echo will be received for that cases. We know this due to the wave propagation speed through water and the distance between both antennas. We measured the behavior of the BPSK, QPSK and CCK modulations with data transfer rates up to 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps.

4.1 Measures for 1 Mbps

Fig. 4 shows the percentage of lost packets for a data transmission rate of 1 Mbps using BPSK modulation. The frequencies that recorded the highest lost packets values were 2.427 GHz, 2.437 GHz and 2.442 GHz, for a distance of 16 cm, while the highest losses for a distance of 17 cm are registered at 2.417 GHz, 2.437GHz and 2.442GHz.

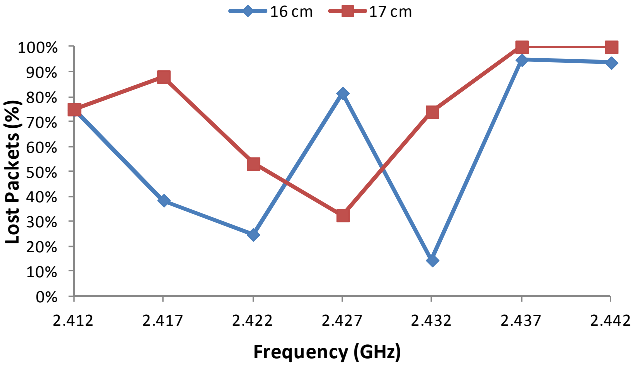


Fig. 4. Lost packets for 1 Mbps data rate

Fig. 5 shows the average RTT values in milliseconds for 1 Mbps data transfer rates, when BPSK modulation is used. The average RTT for both distances is relatively small (around 20 ms). In 2.437 GHz the RTT value for 16 cm increases up to 500 ms, while for 17 cm there are not packets registered and the RTT obtained is 3,000 ms.

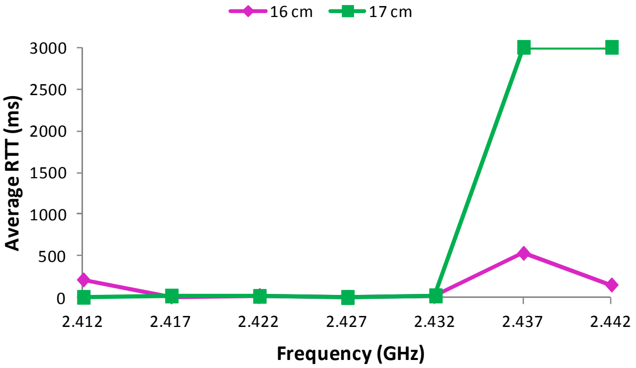


Fig. 5. Average RTT for 1 Mbps data rate

4.2 Measures for 2 Mbps

Fig.6 shows the percentage of lost packets for 2 Mbps data transfer rates, when QPSK modulation is used. In this case, the frequencies with the lowest lost packets percentage are 2.412 GHz, 2.427 GHz and 2.437 GHz, for a distance of 16 cm, while for a distance of 17 cm the lowest losses are given at 2.422 GHz.

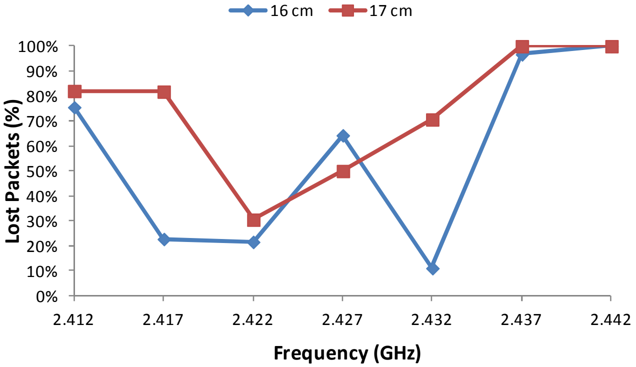


Fig. 6. Lost packets for 2 Mbps data rate

Fig.7 shows the average RTT, in milliseconds, for 2 Mbps data transfer rates when QPSK modulation is used. The average RTT values for both distances are kept below 500 ms for a frequency of 2.432 GHz. For a distance of 16cm, the average RTT value at 2.437 GHz is around 900 ms and, finally, this value increases up to 3,000 ms, at the same frequency, for a distance of 17cm.

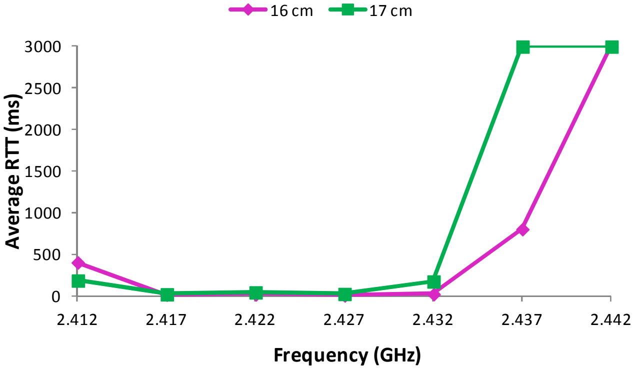


Fig. 7. Average RTT for 2 Mbps data rate

4.3 Measures for 5.5 Mbps

Fig. 8 shows the percentage of lost packets for 5.5 Mbps data transfer rates when CCK modulation is used. We can see that the lost packets percentage has worsened almost

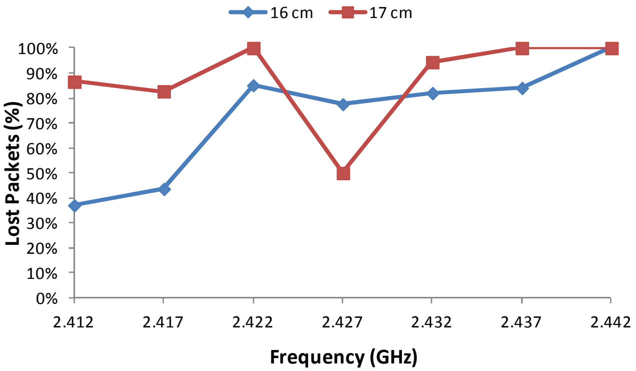


Fig. 8. Lost packets for 5.5 Mbps data rate

threefold at 2.412 GHz and 2.417 GHz for both distances. In addition, for 16 cm, only 2.412 GHz and 2.417 GHz frequencies had losses below 50%, meanwhile, for 17 cm, the frequency that registers the lowest lost packets percentage is 2.427 GHz.

Fig. 9 shows the average RTT, in milliseconds, for 5.5 Mbps data transfer rate, using CCK modulation. In this case, the RTT values for both distances are less than 500 ms from 2.412 GHz to 2.432 GHz, while at 2.437 GHz the RTT value increases above to 2,000 ms in both cases.

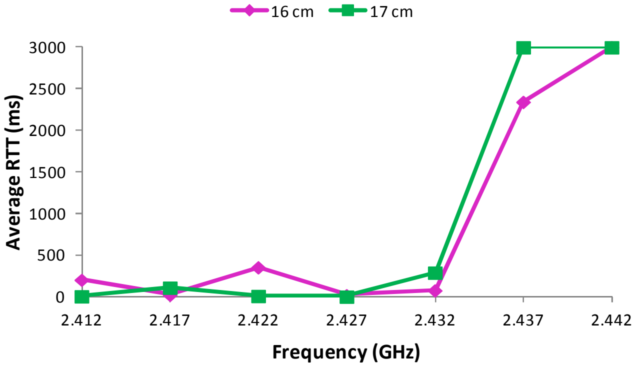


Fig. 9. Average RTT for5,5 Mbps data rate

4.4 Measures for 11 Mbps

Fig. 10 shows the percentage of lost packets for 11 Mbps data rate, when CCK modulation is used. We see that the percentage of lost packets for 16 cm increase almost linearly with the frequency. Just the amount of lost packets for 2.412 GHz and 2.417 GHz, are below 70%. Analyzing the behavior of all frequencies for 17cm, the system presents lost packets values above 70%.

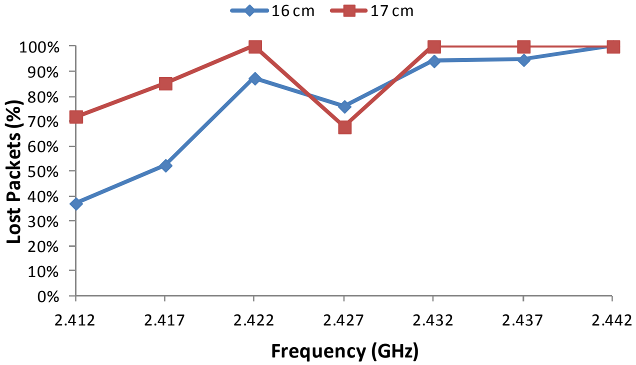


Fig. 10. Lost packets for 11 Mbps data rate

Fig. 11 shows the average RTT in milliseconds for 11 Mbps data transfer rates, when CCK modulation is used. The average RTT values obtained for 16 cm remain around 400-600 ms at frequencies below 2.437 GHz, while in 2.442 GHz we did not receive any packet. In 17cm, the obtained average RTT values are very low for 2.412 GHz, 2.417 GHz and 2.427 GHz, but it reached 3,000 ms. for all other frequencies.

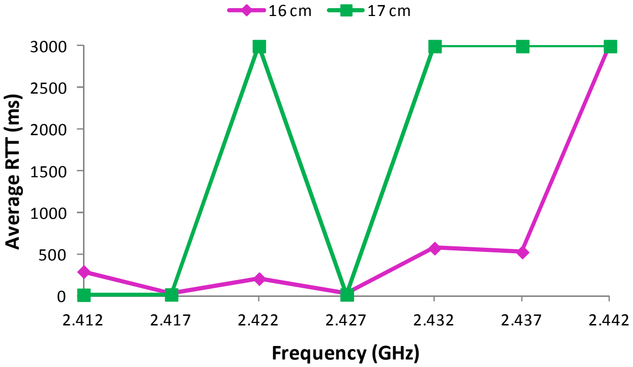


Fig. 11. Average RTT for 11 Mbps data rate

4.5 Summary of Results

After having produced and presented all the results of our measurements, we summarize in this section the values obtained.

Table 2 shows a summary for a distance of 16 cm, with the best results of each case of the measurements previously shown. It specifies the frequencies that showed the lowest lost packets values and the average RTT values in milliseconds. Table 3 shows a summary of the same values, for a distance of 17cm.

Table 2. Summary of results for 16 cm.

Modulation	Best frequencies (GHz)	% of lost packets	Average RRT (ms)
1 Mbps	**2.422** and **2.432**	20% to 30%	28 and 20
2 Mbps	**2.417**, **2.422** and **2.432**	10% to 20%	18, 20 and 7
5.5 Mbps	**2.412** and **2.417**	40% to 50%	204 and 25
11 Mbps	**2.417** and **2.422**	10% to 20 %	24, 208 and 547

Table 3. Summary of results for 17 cm

Modulation	Best frequencies (GHz)	% of lost packets	Average RRT (ms)
1 Mbps	**2.427**	40%	28
2 Mbps	**2.422**	30%	46
5.5 Mbps	**2.427**	50%	3
11 Mbps	**2.427**	70 %	17

As we can see, the amount of lost packets and average value of RTT does not affect them equally at all frequencies. The performance worsens starting from 2.432 GHz to upper frequency values. However, the first frequency does not present a notable degradation of the performance with the increase of frequency.

5 Conclusion

Research on underwater communications and the use of Underwater Wireless Sensor Networks are becoming a very hot topic because of the appearance of new marine/oceanographic applications. Communications based on EM wave transmission offer great benefits such as the increase of the bandwidth of the link to transmit more information.

In this paper, we performed several tests at different frequencies and modulations to check several parameters such as the minimum depth, distance between devices and signal transmission characteristics. These tests have been done in the first seven frequencies that are specified in the IEEE 802.11 standard.

We note several factors. On the one hand, we see that the modulation (thus the data transfer rates) that show better performance are BPSK and QPSK, with percentage of lost packets lower than 30% for distances up to 16 cm. For 17 cm, we also obtained a percentage of lost packets of 30% when QPSK modulation is used. In addition, we observed that the RTT values for 16 cm are around 25 ms, when the system was working at 2.432 GHz. Thus, contrary to what we initially thought (the higher frequency, the higher attenuation), it seems that the global system performance improves slightly when it works at 2.432 GHz, compared with the results of the measurements obtained when it is working at 2.412 GHz.

As we have told, due to our proposal provides short communication distances in UWSN, it is easy to think that because the water has a high attenuation of these frequencies, underwater communications in the 2.4 GHz band, is unhelpful and impractical.

However, there are very specific applications where the use of EM waves to transmit information at very short distances, offer great benefits. We can use it, for precision monitoring such as ecosystems contaminated by invasive plants (especially in ponds where there are some poisonous plants that can contaminate the water) or hazardous waste (e.g. in swamps, the quality of the water is different depending on the season because the water may contain some organic material that may be affected when it is warmer because the pH is different). In both cases the water cannot be used for human consumption, but, in some cases, it can be used by industries to run their plants and supply the water cooling system.

We also would like to use this underwater communication system in the neutrino telescope project [18]. The neutrino telescope is an underwater structure located at the bottom of the Mediterranean Sea. This system allows the detection of cosmic particles, as neutrinos. It consists of thousands of optical detectors and photomultipliers, which must communicate with other system parts, located at distances, extremely small (practically in contact). The photodetectors are distributed in threes along umbilical cables of 450 meters high, designed to carry signals and power. Until now they are using cables and penetrators, to unite the different parts. These pieces have a high economic cost. Using wireless communications, we would be reducing the cost of this material and would avoid the critical connections that can propagate a fault (or leak) through the system. Finally, the fact that the distances between the devices are so small, makes the depth of this infrastructure is not a problem for the transmission of information. There are other applications such as, military applications, marine monitoring and even industrial applications such as marine fish farms [19], to reduce the deposition of organic waste on the seabed and to fight against environmental contamination.

We want to extend the applicability of this system. To do this, our next studies will be focused in two directions. The first will focus on gradually reducing the work frequency, trying to keep the values of transmission. The second line of research will be the design of an antenna optimized for underwater transmission of EM signals in the frequency band of 2.4 GHz and other inferior frequencies that we can prove in the study.

Acknowledgements. This work has been partially supported by the "Ministerio de Ciencia e Innovación", through the “Plan Nacional de I+D+i 2008-2011” in the “Subprograma de Proyectos de Investigación Fundamental”, project TEC2011-27516, and by the Polytechnic University of Valencia, though the PAID-15-11 multidisciplinary projects.

References

1. Mohsin, A.H., Bakar, K.A., Adekiigbe, A., Ghafoor, K.Z.: A Survey of Energy-aware Routing protocols in Mobile Ad-hoc Networks: Trends and Challenges. Network Protocols and Algorithms 3(4), 1–17 (2011)
2. Segal, M.: Improving Lifetime of Wireless Sensor Networks. Network Protocols and Algorithms 1(2), 48–60 (2009)
3. Garcia, M., Sendra, S., Atenas, M., Lloret, J.: Underwater Wireless Ad-hoc Networks: a Survey. In: Mobile Ad hoc Networks: Current Status and Future Trends, pp. 379–411. CRC Press (2011)

4. Chakraborty, U., Tewary, T., Chatterjee, R.P.: Exploiting the loss-frequency relationship using RF communication in Underwater communication networks. In: Proceedings of 4th International Conference on Computers and Devices for Communication, CODEC 2009, Kolkata, India, December 14-16 (2009)
5. Liebe, H.J., Hufford, G.A., Manabe, T.: A model for the complex permittivity of water at frequencies below 1 THz. International Journal of Infrared and Millimeter Waves 12(7), 659–675 (1991)
6. Somaraju, R., Trumpf, J.: Frequency, Temperature and Salinity Variation of the Permittivity of Seawater. IEEE Transactions on Antennas and Propagation 54(11), 3441–3448 (2006)
7. IEEE Std 802.11, IEEE Standard for Information technology—telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, New York, USA, pp.1–1184 (2007)
8. Chaitanya, D.E., Sridevi, C.V., Rao, G.S.B.: Path loss analysis of underwater communication systems. In: 2011 IEEE Students' Technology Symposium (TechSym 2011), Kharagpur, India, January 14-16, pp. 65–70 (2011)
9. Sehgal, A., Tumar, I., Schonwalder, J.: Variability of available capacity due to the effects of depth and temperature in the underwater acoustic communication channel. In: OCEANS 2009 – EUROPE, Bremen, Germany, May 11-14, pp. 1–6 (2009)
10. Arnon, S.: Underwater optical wireless communication network. Journal of Optical Engineering 49, 015001 (January 15, 2010), doi: 10.1117/1.3280288
11. Hunt, K.P., Niemeier, J.J., Kruger, A.: RF communications in underwater wireless sensor networks. In: IEEE International Conference on Electro/Information Technology 2010, Normal, Illinois, USA, May 20-22 (2010)
12. Jiang, S., Georgakopoulos, S.: Electromagnetic Wave Propagation into Fresh Water. Journal of Electromagnetic Analysis and Applications 3(07), 261–266 (2011)
13. Sendra, S., Lamparero, J.V., Lloret, J., Ardid, M.: Underwater Communications in Wireless Sensor Networks using WLAN at 2,4Ghz. In: International Workshop on Marine Sensors and Systems (MARSS), Valencia, Spain, October 17-22 (2011)
14. Martin, F., Gorday, P., Adams, J., Leeuwen, H.V.: IEEE 802.15.4 PHY Capabilities (May 2004) Doc.: 15-04-0227-04-004A, `https://mentor.ieee.org/802.15/file/04/15-04-0227-04-004a-ieee-802-15-4-phy-layer-and-implementation.ppt`
15. Chitode, J.S.: Digital Communications, 1st edn. Technical Publications Pune (2007-2008)
16. Andren, C., Webster, M.: CCK Modulation Delivers 11Mbps for High Rate 802.11 Extension. In: Proceedings of the Wireless Symposium/Portable By Design Conference, San Jose, CA, USA, February 22-26 (Spring 1999)
17. Lloret, J., López, J.J., Ramos, G.: Wireless LAN Deployment in Large Extension Areas: The Case of a University Campus. In: Proceedings of Communication Systems and Networks 2003, Benalmádena, Málaga, Spain, September 8-10 (2003)
18. Ardid, M.: ANTARES: An Underwater Network of Sensors for Neutrino Astronomy and Deep-Sea Research. Ad Hoc & Sensor Wireless Networks 8, 21–34 (2009)
19. Garcia, M., Sendra, S., Lloret, G., Lloret, J.: Monitoring and Control Sensor System for Fish Feeding in Marine Fish Farms. IET Communications 5(12), 1682–1690 (2011); The Institution of Engineering and Technology

A Parameter-Based Service Discovery Protocol for Mobile Ad-Hoc Networks

Unai Aguilera and Diego López-de-Ipiña

Deusto Institute of Technology - DeustoTech
University of Deusto
Avda. Universidades 24, 48007 Bilbao, Spain
{unai.aguilera,dipina}@deusto.es
http://www.morelab.deusto.es

Abstract. Application of traditional service discovery solutions to mobile ad-hoc networks is a challenging task due to their intrinsic dynamic nature and the absence of any central information manager. However, service discovery is a critical aspect of service oriented technologies, e.g. remote service execution or, particularly service composition. We propose a solution for service discovery in mobile ad-hoc networks which is based on the dissemination of information about services' parameters instead of service unique identifiers. Disseminated information is subsequently used during service search in order to reduce the number of propagated messages. In our solution, performed searches are maintained in the network until they are explicitly cancelled by source nodes. We also state that the usage of a shared taxonomy of parameter types reduces the number of propagated messages during dissemination and search. The proposed protocol has been fully implemented and tested using a network simulator.

Keywords: mobile ad-hoc networks, manet, service discovery, parameter-based, taxonomy.

1 Introduction

Mobile ad hot networks (MANET) are a type of wireless network which are characterized by the mobility of their nodes and the absence of a fixed communication infrastructure. Nodes in these networks communicate among each other using neighbour broadcasting and performing collaborative routing of transmitted messages. In service oriented architectures, an application which is running in a node could want to access services which are offered by other near or remote nodes in the network. Due to the dynamism of mobile ad-hoc networks the application will first need to discover the required service prior the usage of its functionality. Service discovery is a fundamental process for the application of any service oriented task in dynamic environments. For example, in service composition, when needed services are searched and no directly compatible ones exist, new service work-flows are created which provide the required functionality to the user. Service composition is performed by means of using the required

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 274–287, 2012.

service as a contract which could be fulfilled by executing other available services in the network in some defined order. Service composition uses discovery to search those services which could be used to create the composed service [1]. One of the many aspects which must be solved during service composition is input/output compatibility [11]. Currently proposed solutions for service discovery in mobile ad-hoc networks are based on searching using the service's identifier or type. However, for tasks such as service composition, we think that it will more suitable to search services based on their parameters' type.

This paper proposes a solution for service discovery in wireless ad-hoc networks which tries to reduce the overload produced by service discovery protocol messages while facing its dynamic nature. Our proposal is based on the dissemination of parameter information instead of information about services themselves. Nodes disseminate, to the network nodes, the information about the input and output parameters of the services they provide. We think that to propagate parameter information is more adequate for service discovery when facing problems such as service composition than searching services using its type or identifier. Furthermore, propagating service type information requires all nodes in the network to know a description for each possible available service in the network. On the other hand, when disseminating parameter information nodes only need to share knowledge about different data types enabling the specification of multiple service signatures by means of reusing them. In addition, applications which perform service composition in the network will benefit from this information when applying matching of available services.

The proposed protocol uses a proactive approach for information dissemination. Whenever a new neighbour appears nodes send their parameter information table which contains information about their own provided parameters and those provided by near nodes. Replication of whole services or individual parameters could exist in the network meaning that some nodes could be propagating redundant information. Furthermore, it is not only possible that two or more parameters, which are propagated in the same network area, are exactly equal but they could also be related by some generalization relationship. In the proposed solution, all nodes in the network share a common taxonomy which is used during dissemination to decide if the information currently being propagated adds some extra information to the previously disseminated one.

In our proposal, services are searches by means of the parameters they provide. This means that clients searching for a service send search messages containing the type of the input and output parameters of the needed services. After a search is started it is maintained in the network until the source node explicitly cancels it. Active searches are propagated to new neighbours when they appear and removed when a communication link breaks due to mobility. This means that new appearing services will be matched against active searches without the need of sending new search messages periodically. In addition, disseminated information is used during searching in order to reduce the scope of the messages.

The rest of the paper is organized as follows. The next section presents the parameter-based service discovery protocol in more detail. Section 3 presents the evaluation performed to test the proposed protocol, Section 4 summarizes the related work in the area of service discovery in mobile ad-hoc networks and, finally, Section 5 concludes the paper and presents some future work.

2 Parameter-Based Service Discovery

The proposed parameter-based service discovery protocol is the result of different layers and mechanisms working together: dissemination and search layers, neighbour detection and the application of reliable broadcast.

2.1 Parameter Dissemination

The parameter dissemination starts on those nodes which have registered services with input and/or output parameters. Each of these parameters must be categorized according to a predefined taxonomy of types which is shared among all the nodes of the network. The usage of a common taxonomy means that nodes can only interoperate if they have *a priori* knowledge of this taxonomy. However, the proposed solution does not include mechanism to propagate parameter taxonomies among nodes, so a previously shared taxonomy is supposed.

Each node in the network maintains a table with information about those parameters types which are located near him ([0-n] hops). The table contains an entry for each different parameter type associated with a value known as the *estimated distance* to some near node which provides the parameter. The distance to which parameter information is propagated is defined by the dissemination distance D_d. The *estimated distance* value for a parameter is reduced each time the information about a parameter type is propagated one-hop in the network. The higher the *estimated distance* value is for a parameter type the nearer that a parameter of that type will be to the current node. Those parameters which are local to a node, that is, which belong to a service registered in the own node, have the maximum value, equal to D_d in their entry of the parameter table. For example, see Figure 1, where $D_d = 3$ and the dotted node provides some services with parameters, those nodes located at n-hops will have a value for the *estimated distance* of $D_d - n$ until the the value reaches 1 and the information is not further propagated.

The parameter table is defined as a table of P, L pairs where P is a parameter type and L is the list of *estimated distances* to one or more parameters of that type. The list contains elements of the form (D, N) where D is the *estimated distance* and N is the unique identifier of the neighbour which supplied that information. The list can contain more than one element, however, the *effective distance indicator* is the element of the list with the greatest value for the *estimated distance*. The list cannot contain two different *estimated distance* values provided from the same neighbour node. Therefore, new values for the *estimated distance* of a parameter coming from the same neighbour will modify

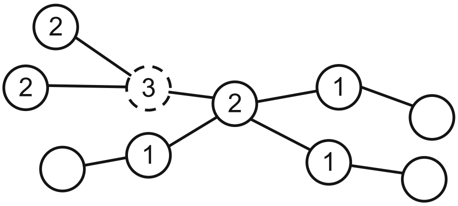


Fig. 1. Dissemination of parameters type information through the network

the old value. Storing the source neighbour of the information, N, enables to avoid back-propagation of disseminated information. When a node receives information about the *estimated distance* to a parameter from a neighbour node the received information will contain the previous source of that information. Therefore, if the information was originally provided by the current node it will mean that back-propagation is occurring and the received information will be discarded.

Nodes disseminate parameter type information by sending a *UpdateTable* messages to neighbours. These messages can contain their whole table or only those changes which where produced since last update. Because sometimes table differences are propagated, nodes rely in the fact that previous information was correctly propagated. Therefore, there is a need for reliable broadcasting during message propagation, see Section 2.5.

UpdateTable messages are sent in response to one of the following events: neighbour appearance will cause the current node to broadcast its current whole parameter table, if it is not empty. The disappearance of a neighbour causes the current node to remove information which came from the disappeared neighbour, producing changes in its table and propagating that changes to neighbours if needed. Every time a node modifies its local parameter table, including the addition or removal of local services, the update might cause, in turn, the propagation of the update to neighbour nodes. Each *UpdateTable* message contains the changes to apply to the neighbour tables. These changes are specified as a list of deletions and a list of additions. Deletions, which are processed first by receiving nodes, contain a list of entries which must be removed from the receivers' tables. On the other hand, the addition list contains new information which must be added by receivers. The detailed algorithm for processing *UpdateTable* messages is represented in Figure 2 and works as follows:

For each parameter type contained in the update message, if a deletion exist, the *estimated distance* which was previously provided by the neighbour sending the update is removed. If the removed *estimated distance* is greater than 1 the deletion is propagated to neighbours. Also, if after deleting the element the new *effective distance indicator* is greater than one an addition containing the new *estimated distance* is propagated to neighbours. Additions are processed next, and only if the added information was not originally provided by the current node. After addition, if the current *effective distance indicator* (the head of distance list) has not changed, the addition must not be further propagated. Furthermore,

```
function processUpdateTable(updateTable, neigh, paramTable, node)
    $newNeighTable \leftarrow EmptyTable$
    for each Parameter p in updateTable do
        $eDistanceList \leftarrow getEstimatedDistanceList(paramTable, p)$
        if $containstDelete(updateTable, p)$ then
            $\{D, N\} \leftarrow removeEntryFrom(eDistanceList, neigh)$
            if $D > 1$ then
                $insertDelete(newNeighTable, p)$
                $\{D_{new}, N\} \leftarrow getEstimatedDistance(p)$
                if $D_{new} > 1$ then
                    $insertAddition(newNeighTable, p, D_{new})$
                end if
            end if
        end if
        if $containsInsert(updateTable, p)$ then
            $\{D, N\} \leftarrow getInsert(updateTable, p)$
            if $N \neq node$ then
                $D_{previous} \leftarrow getEstimatedDistance(eDistanceList, p)$
                $D_{new} \leftarrow addEstimatedDistance(eDistanceList, D, N)$
                if $D_{new} > D_{previous}$ and $D_{new} > 1$ then
                    $insertAddition(newNeighTable, p, D_{new})$
                end if
            end if
        end if
    end for
    if $notEmpty(newNeighTable)$ then
        $propagateTable(newNeighTable)$
    end if
end function
```

Fig. 2. Algorithm for parameter information dissemination

only additions whose *estimated distance* is greater than 1 are propagated in the resulting updated message. All the resulting deletions and additions are include in the same update message in order to reduce the number of transmissions.

2.2 Using Taxonomy Information

Nodes contain a taxonomy about available parameter types meaning that they are able to known if one of the following relationships exists between two parameters. Let A and B be two parameters, *equality* occurs when A and B have the same exact type, while *subsumption* means that parameter A has a type which is more general than parameter B, according to the taxonomy. Finally, parameters are *not-related* if none of the previous conditions occurs. The taxonomy of parameter types can be expressed using languages such as XMLSchema. RDF, OWL [10] or any other ad-hoc implementation which could be supported by the network devices.

In order to use taxonomic information the algorithm presented in Section 2.1 is modified in the following way. The parameter table works with parameter groups instead of simple parameter types. Each group represents all those parameters which are related through *equality* or *subsumption* relationships. Each group is represented by the more general type of all those parameters which are contained within the group. The representative type changes when more general parameters than those already registered in the group are added. However, when parameters are eliminated from a group its representative type does not change and its maintained until the group is deleted. Now, the parameter table maintains an entry for each parameter group instead of parameter type. During the dissemination of parameter information the representative type of the corresponding updated parameter group is propagated. Whenever a new parameter is added to its corresponding group, due to an addition contained in an update message, and the *estimated distance* for that group is greater that 1, an update message is propagated. This way the generalization of parameter types is disseminated across the network. Entries are now removed when all the parameters which are within the same parameter group are eliminated.

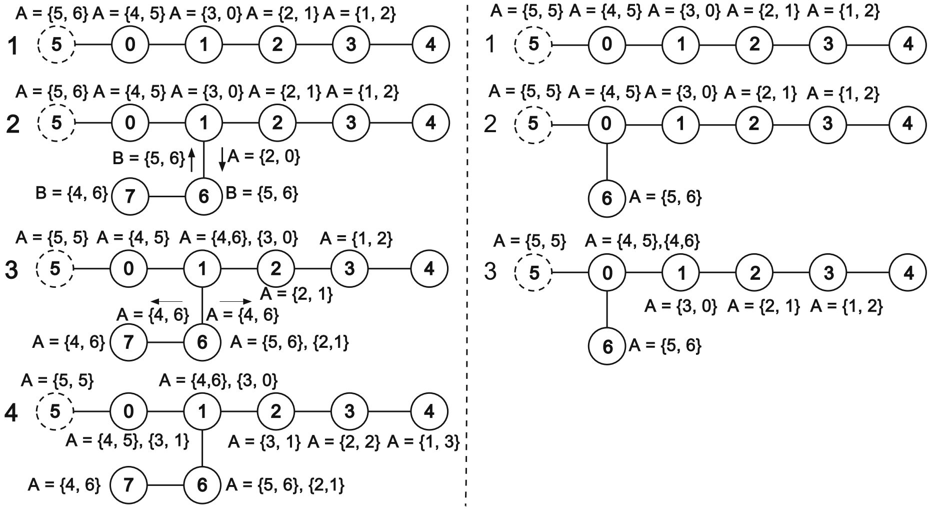

A = {5, 6} A = {4, 5} A = {3, 0} A = {2, 1} A = {1, 2}
1
5
0
1
2
3
4
A = {5, 6} A = {4, 5} A = {3, 0} A = {2, 1} A = {1, 2}
2
B = {5, 6}
A = {2, 0}
B = {4, 6}
7
6
B = {5, 6}
A = {5, 5} A = {4, 5} A = {4,6}, {3, 0} A = {1, 2}
3
A = {2, 1}
A = {4, 6}
A = {4, 6}
A = {4, 6}
A = {5, 6}, {2,1}
A = {5, 5} A = {4,6}, {3, 0}
4
A = {4, 5}, {3, 1}
A = {3, 1} A = {2, 2} A = {1, 3}
A = {4, 6}
A = {5, 6}, {2,1}
A = {5, 5} A = {4, 5} A = {3, 0} A = {2, 1} A = {1, 2}
1
A = {5, 5} A = {4, 5} A = {3, 0} A = {2, 1} A = {1, 2}
2
A = {5, 6}
A = {5, 5} A = {4, 5},{4,6}
3
A = {3, 0} A = {2, 1} A = {1, 2}
A = {5, 6}

Fig. 3. Parameters are propagated using taxonomic information

Figure 3 shows, in the left side, what happens when the propagation is produced using taxonomic information. The taxonomy used in the example has two parameters A and B, with A being more general than B. Node 5 has propagated its local parameter information through the network with a $D_d = 5$. Then, node 6 appears containing information about parameter B. Nodes 1 and 6 interchange their whole parameter table information and the parameter groups of each node's table are updated. Node 6 updates the representative type of the stored parameter because it has added parameter A which has a more general type than the previously included one. In this case, it also causes the notification of neighbour nodes with the information of the more general parameter A meaning that node 7 receives this information. Finally, node 1 will propagate the *estimated distance* information because the new *estimated distance* is greater than the previous one.

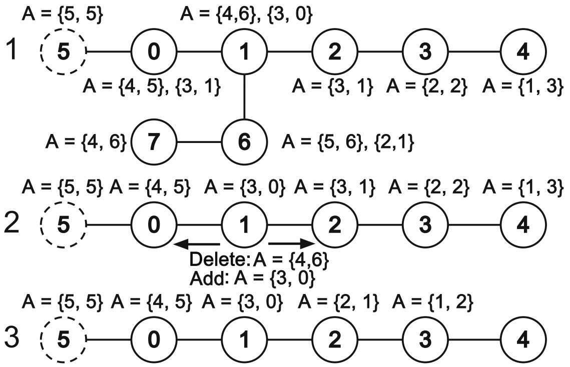


Fig. 4. Network breaks producing information propagation

The right side of Figure 3 shows what happens when related information was already propagated. When the node 6 appears in the network its disseminated information is stopped by node 0 because the *estimated distance* list did not change after the update. Finally, Figure 4 shows what happens when the network breaks and the *estimated distance* list changes in node 1 after removing the information from node 6. Removal is notified to neighbour nodes and the new *estimated distance* information is propagated across the network.

2.3 Parameter-Based Search

The parameter-based service discovery is performed by means of propagating search messages through the network. Nodes start searches by sending a *SearchMessage* which contains those parameter types whose associated services want to be discovered. Search messages have a *TTL* which is used to control the area where the search is propagated. Searches are flooded through the neighbour and, in order to avoid duplicated propagation, they contain a unique identifier (i.e. source node and message number) which enables to drop those searches which are received more then once. Whenever a search is received by a node it is checked against the local parameters, that is, those parameters which belong to services registered in the receiving node. Two types of searches are supported: *Exact* and *Generic*. In the first case, the search is accepted if the node contains parameters whose types is exactly the same than the searched one. In the second case, not only exact matches but also those parameter which are subsumed by the searched taxonomy concept result in search being accepted.

When a search message is accepted by a node, a *SearchResponse* message is sent to the searching node. Responses are sent using unicast routes created during the propagation of search messages. Response messages also triggers the creation of unicast routes, in this case, which enable to communicate with the node offering the found parameters.

Searches do not expire until they are explicitly cancelled using a *SearchCancel* message. This means that searching nodes do not need to re-propagate searches periodically in order to locate new services. When a node receives a search it is stored as an *active search* meaning it will be propagated to any neighbour appearing later. Furthermore, these also means, that if new service is registered in those nodes which have an active search the source node will be notified and ad communication route will be created.

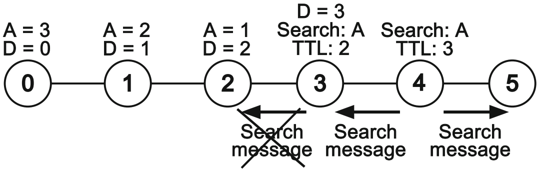


Fig. 5. Search messages are pruned using disseminated information

In order to reduce the number of messages during search a message prune is applied. Searches are only propagated to neighbours if its TTL, which is decremented after each hop, is greater or equal than the distance for the searched parameters according to the current neighbour's parameter table. The distance to a parameter is calculated as $D_d - estimatedDistance$. If the previous condition does not hold, it means that the search message is in a node which, although it can contain information about the searched parameter, the source of the parameter cannot be reached with the current message TTL. For example, in Figure 5 the search message is not propagated by node 3 because its TTL is lower than the distance calculated to the searched parameter and it could no be reached with the current TTL. If a parameter type does not have an entry in a table its distance is calculated as D_d, the maximum possible value.

The parameter search layer reacts to node mobility in the following ways. When a new neighbour appears active searches are propagated to new neighbours according to the previously explained pruning rules. It also possible that new parameters appear in the network causing that active searches are checked against that new information added to the parameter table. If matches exist active searches are propagated according to the pruning rules explained before. On the other hand, the disappearance of a neighbour produces that searches coming from the disappeared neighbour are invalid and must be removed. A *RemoveSearch* message, containing the disappeared searched, is propagated to neighbours. Created unicast routes are also cleared when a link breaks using a *RemoveRoute* message.

2.4 Neighbour Detection

The protocol being proposed is proactive while performing parameter dissemination meaning that it disseminates and maintains information about parameters during the mobility of the network. Searches, on the other hand, are only propagated when a node wants to search for services offering a set of parameters. However, once a search has been initiated it is maintained in the network meaning that any new found compatible parameter will be notified to the searching node. In both cases, information must be added or eliminated when nodes appear or disappear.

The reaction to the network mobility is performed thanks to the usage of beacon messages. Using beaconing for neighbour detection has its advantages and disadvantages, which have been studied in [4], [3]. Nodes send beacon messages with a period of T_b seconds and each node maintains a table with all the currently discovered neighbours. Every time a message from a unknown neighbour is heard by a node the source of the message is added to the node's neighbour table. If a node does not hear a message during a period of time $T \geq 2 \cdot T_b$ from a neighbour, it is considered to be inaccessible and, therefore, removed from the current node's neighbour table.

In order to reduce the number of messages, two improvements are made to this basic beaconing algorithm. Firstly, any message sent by the node can act as a beacon message. This means that beacon messages are only sent by a node if no other messages were previously sent in the defined period of time. Secondly, changes in the neighbour table, i.e. additions or removals, are notified to the node with a small delay. This avoids producing multiple events and, therefore, sending multiple messages due to multiple simultaneous neighbours appearance or disappearance. For example, multiple detected neighbours detected in a small amount of time when a node joins to a network.

2.5 Reliable Broadcast

Despite medium access control performed by *IEEE802.11*, message collision yet occurs due to the *hidden terminal* problem, reducing the network capacity [9]. The parameter-based service discovery protocol needs the usage of reliable broadcast [16] due to the fact that information messages are not periodically refreshed among nodes. Information is sent only when changes occur and are applied to the maintained tables (e.g when neighbours appear/disappear). Once the information was reliable propagated it is considered to be known by all neighbours nodes in the current state of the neighbour.

In reliable broadcast 1-hop neighbours which receive a message from a node must reply with an ACK message. In order to reduce medium usage if various acknowledgement messages are pending they are bundled together in a single transmission. Furthermore, nodes delay the transmission of their information messages during a fixed period of time T_w in order to minimize the medium usage by bundling may messages into a single transmission. Each transmitted message contains a list of all the expected destinations nodes. Only nodes which are contained in this list will respond with ACK messages. A message is considered as *delivered* if the transmitter node receives acknowledgements from all the expected destinations. The time a node waits for ACK responses is calculated as $T_t \cdot (destinations + 1) + T_w$, where T_t is the estimated transmission time in a medium without collisions or interferences. The wait time T_t accounts for the initial transmission time, the transmission of all ACK responses and the fixed delay added by each node. Messages are re-broadcast until all expected destinations receive the message.

As a transmission can contain multiple messages, each one with a different set of destination neighbours, those contained messages which were already delivered, are removed from further rebroadcasting. Furthermore, every time a node disappears it is removed from the expected destinations of the current message, as it could not be ever reached. The node applies a back-off time after each rebroadcasting to reduce network congestion. The back-off increases with the following equation $T_{backoff} = T_w \cdot t^2$ where t is the current rebroadcasting try number. This enables to reduce the number of rebroadcasting on those cases when a neighbour of the transmitting node has disappeared.

3 Evaluation

The proposed protocol has been fully implemented and evaluated using the NS-2 network simulator[1] extended with AgentJ[2]. This extension enables to execute Java applications directly on each node. The implementation has been performed using Java SE 1.6 and the default communication libraries (i.e. plain Java UDP sockets). This means that the resulting implementation could be directly deployed in Java enabled devices (e.g Android smartphones).

Evaluation scenarios has been constructed with the following characteristics: 100 nodes with a uniform speed distribution ranging from 0-5 m/s over an area of 700 x 700 meters. According to [7] this scenario has an Average Network Partition $< 5\%$ and an Average Shortest Path = 4.15 hops. The ns-2 simulator has been configured to use a *TwoRayGround* reflection model. *IEEE802.11* is the only used underlying protocol configured with a data-rate of 11 Mb/s and a maximum packet size of 1500 bytes. Experiments were configured with a TTL = 10 hops for dissemination and search propagation. In all experiments searches were performed in simultaneous sets with a period of 5 seconds between each set of searches. Each time a set of searches is triggered a group of 5 randomly selected nodes start simultaneous searches of available parameters. Searches were performed during 100 seconds of the simulation and each one was maintained during 10 seconds until it was explicitly cancelled by its source node. Each simulation has been repeated 10 times and the obtained results were averaged. Experiments have been repeated for nodes configured with a pause time of 100 and 50 seconds among each random movement.

[1] The Network Simulator - ns-2 - http://www.isi.edu/nsnam/ns/

[2] AgentJ Java Network Simulations in NS-2 - http://cs.itd.nrl.navy.mil/work/agentj/

3.1 Dissemination Overhead Reduction

The following experiment was conducted in order to measure the reduction in dissemination messages produced by the existence of duplicated parameters on the network. First, we performed a simulation where a variable number (ranging from 2 to 20) of services where randomly distribute on different nodes in the network with each service having a total of 6 different I/O parameters. These results are compared with a configuration where the same number of services is used but those services are obtained by replicating two initial services. Figure 6 shows, for two different nodes pause times (100 and 50 seconds) that the number of *Sent Table Messages* is effectively reduced due to service replication. Figure also shows that the time needed to discover the first occurrence of a parameter is reduced, as expected due the fact that the probability of finding a nearer compatible parameter increases with replication.

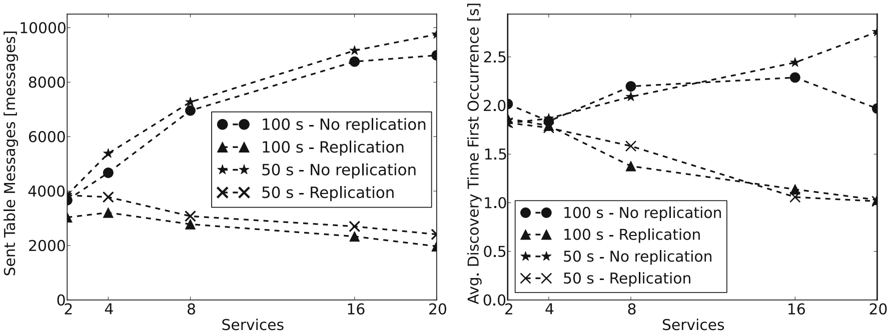


Fig. 6. Table messages reduction due to parameter replication

3.2 Search Message Pruning

The aim of this experiment was to show how search messages are pruned according to the rules explained in the previous section. The experiment was configured by selecting 30 nodes in the network which were configured with 6 different parameters each one. Search pruning has been simulated using searches which are trying to locate parameters which do not exist on the network and increasing the ratio of those non available parameters in each configuration. Figure 7 shows different metrics for the performed simulation. Plot a) shows the average ratio of discovered parameters calculated taking into account the number of really existing parameters on the network. The discovery ratio is not perfect because of the existence of disconnected nodes in the network and the used TTL. Obviously, when the ratio of non-existing searched parameters is 1.0 the average discovered parameters decays to 0. Plot b) shows the average discovery time and, as shown, it gets reduced as the congestion of the network is reduced due to the increase of invalid searches. Finally, plots c) and d) show that the traffic overhead gets reduced thanks to the applied pruning. *Multicast Traffic Overhead* is the average

KB/s obtained by taking into account the size of those messages related with parameter dissemination and search messages. Finally, *Sent Search Messages* is the total number of search messages sent during the simulation. The application of search pruning effectively reduces the traffic overhead due to the reduction of the number of sent search messages while there are not necessary.

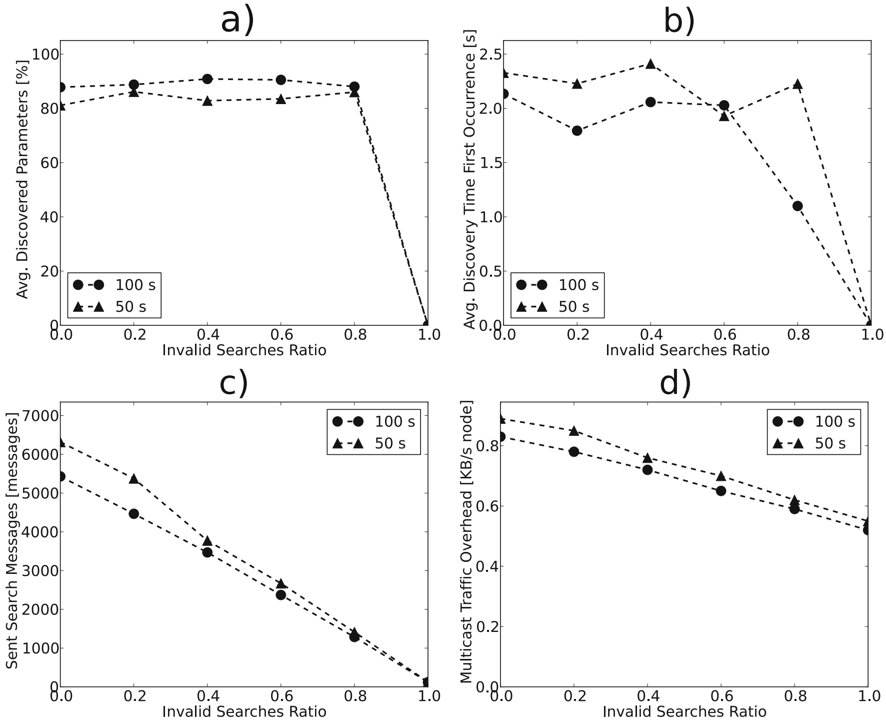


Fig. 7. Search messages are pruned using disseminated information

4 Related Work

Traditional approaches for service discovery, such as Jini, Salutation, UPnP or Bonjour, have been compared in [15] showing their inappropriateness for their application to mobile ad hoc networks. In [14] the authors propose a directory-based architecture which uses a group of selected nodes as a back-bone which maintain information about services. Our proposal, on the other hand, can be categorized as directory-less architecture which uses controlled flooding by means of reducing the scope of propagated messages, thus, avoiding the need to maintain the backbone and reducing the possible points of failure. In [8] the authors propose a service discovery protocol were electrostatic field theory is applied. Services are modelled as electric charges and request messages are directed using a mechanism similar to an attracting force. Our proposal has some resemblances to this work in the way that parameter information is disseminated through the network. However, our proposal enables to find all those services which are offering a particular parameter in some area and not only those with the highest attracting factor.

The usage of taxonomies and, in a more general way semantic information, for service discovery in mobile ad hoc networks has been previously explored. For example, GSD [2] and Konark [5] use OWL to describe services and service requests. However, these two approaches treat the services as a whole and do not enable to search for services using the parameters they provide. In [6] the authors also propose a service discovery mechanism which uses ontology information to search for services in the network. This solution uses a service registry to maintain the information of all available services meaning that information is partially centralized on some nodes of the network. Finally, the dissemination of service information with the usage of semantic overlay networks based on DHT information distribution has been explored in [13].

Our work could also have some similarities to a publish/subscribe system were service providers publish their parameter information and clients subscribe to any service provider with the required characteristics, in our case I/O parameter types. Application of pub/sub systems to mobile ad hoc networks has been recently studied in [12]. However, our contribution includes the usage of taxonomic information into the dissemination and search processes enabling to reduce the number of propagated messages and to perform richer searches.

5 Conclusions and Future Work

This paper has presented a service discovery protocol for mobile ad-hoc networks where search is performed through services' parameters instead of service themselves. The protocol disseminates information about parameters provided by each service located on different nodes. The dissemination leverages on the use of a shared parameter taxonomy in order to reduce the number of propagated messages. Searches are maintained in the network until they are explicitly cancelled by source nodes. Two kinds of search messages are supported: *exact* searches and *generic* searches. The former search locates those parameters with a type equal to the searched one, while the latter will locate equal and more specific parameters according to the taxonomy. The proposed protocol has been fully implemented and evaluated using a network simulator. The results show that the proposed protocol effectively reduces the network messages by means of usage of a parameter taxonomy.

For the future, we plan to study if the proposed protocol could be generalized to propagate any kind of information and not only parameter related one. We think that the dissemination and discovery protocol could be used in any situation where node discovery must be performed using nodes' particular information and a taxonomy of concepts exists. Also, we are interested in examine how it could be possible to propagate different concept taxonomies through the mobile network avoiding the need to share a fixed common taxonomy by all the participant nodes.

References

1. Brønsted, J., Hansen, K.M., Ingstrup, M.: Service composition issues in pervasive computing. IEEE Pervasive Computing 9(1), 62–70 (2010)
2. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward distributed service discovery in pervasive computing environments. IEEE Transactions on Mobile Computing 5(2), 97–112 (2006)
3. Giruka, V.C., Singhal, M.: Hello protocols for ad-hoc networks: Overhead and accuracy tradeoffs. In: Sixth IEEE Intl. Symposium on a World of Wireless Mobile and Multimedia Networks, pp. 354–361. IEEE (2005)
4. Heissenbüttel, M., Braun, T., Wälchli, M., Bernoulli, T.: Evaluating the limitations of and alternatives in beaconing. Ad Hoc Networks 5, 558–578 (2007)
5. Helal, S., Desai, N., Verma, V., Lee, C.: Konark-a service discovery and delivery protocol for ad-hoc networks. In: Wireless Communications and Networking, vol. 3, pp. 2107–2113. IEEE (2003)
6. Islam, N., Shaikh, Z.A.: Towards a robust and scalable semantic service discovery scheme for mobile ad hoc network. Pak. J. Engg. & Appl. Sci. 10, 68–88 (2012)
7. Kurkowski, S., Navidi, W., Camp, T.: Constructing MANET simulation scenarios that meet standards. In: IEEE Intl. Conf. on Mobile Adhoc and Sensor Systems, pp. 1–9 (2007)
8. Lenders, V., May, M., Plattner, B.: Service discovery in mobile ad hoc networks: A field theoretic approach. Pervasive and Mobile Computing 1(3), 343–370 (2005)
9. Li, J., Blake, C., De Couto, D.S., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: Proc. of the 7th Intl. Conf. on Mobile Computing and Networking, pp. 61–69. ACM (2001)
10. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C Recommendation 10, 2004–03 (2004)
11. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic matching of web services capabilities, pp. 333–347 (2002)
12. Paridel, K., Vanrompay, Y., Berbers, Y.: Fadip: Lightweight publish/Subscribe for mobile ad hoc networks. On the Move to Meaningful Internet Systems, 798–810 (2010)
13. Pirrò, G., Talia, D., Trunfio, P.: A DHT-based semantic overlay network for service discovery. Future Generation Computer Systems (2011)
14. Sailhan, F., Issarny, V.: Scalable service discovery for MANET. In: Proc. of the Third IEEE Intl. Conf. on Pervasive Computing and Communications, pp. 235–244. IEEE Computer Society (2005)
15. Ververidis, C.N., Polyzos, G.C.: Service discovery for mobile ad hoc networks: A survey of issues and techniques. IEEE Communications Surveys & Tutorials 10(3), 30–45 (2008)
16. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proc. of the 3rd ACM Intl. Symposium on Mobile Ad Hoc Networking and Computing, pp. 194–205 (2002)

RCDP: A Novel Content Delivery Solution for Wireless Networks Based on Raptor Codes

Miguel Báguena, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni

Department of Computer Engineering
Universitat Politécnica de Valéncia
Camino de Vera S/N, 46022, Spain
mibaal@upvnet.upv.es, {calafate,jucano,pmanzoni}@disca.upv.es

Abstract. The growth of research on Forward Error Correction (FEC) coding has boosted the usage of FEC strategies when addressing the challenges of multicast and broadcast delivery. However, FEC approaches can also be used for unicast content delivery to avoid known TCP issues in wireless network environments. In this paper we exploit the error resilience properties of Raptor codes by proposing RCDP, a novel solution for reliable and bidirectional unicast communication in lossy links that can improve content delivery in situations where the network becomes the bottleneck. Since the implementation of RCDP in real systems involves important technical challenges, we also focus on the design, implementation, and optimization issues, proposing different architectural and design alternatives for RCDP. Our goal is to find the best trade-off between complexity and efficiency in order to maximize the throughput achieved under different conditions. Experimental results show that RCDP is a highly efficient solution for environments characterized by high delays and packet losses (e.g. ad-hoc networks), achieving significant performance improvements compared to traditional transport-layer protocols.

Keywords: Application-layer FEC, Raptor codes, design and implementation, testbed.

1 Introduction

In recent years, quite sophisticated Forward Error Correction (FEC) algorithms have been proposed, being Raptor codes one of the most remarkable solutions for application-layer FEC (AL-FEC). In the literature, most authors adopting such FEC strategies have mostly addressed the challenges associated with multicasting/broadcasting to a high number of users, being video delivery the application of choice. However, the FEC approach is also applicable to unicast content delivery, where wireless transmission channels reduce the throughput achieved by conventional TCP-based delivery protocols. Most of the TCP issues in wireless networks are described in [1] and an analytical model is presented in[2]. However, there are many specific-purpose systems which can sacrifice the TCP

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 288–301, 2012.

compatibility in order to achieve a higher performance level, like sensor networks, point-to-point satellite connections or vehicular ad-hoc networks.

These systems attempt to use efficiently the network resources, no matter whether it is wired, wireless or a combination of both. To achieve this goal, the protocols involved have throughput maximization as a primary objective. Nevertheless, protocol design must also take into account other characteristics such as flexibility, scalability and reliability, seeking a trade-off between them, in order to allow a better network experience and an efficient operation in all sorts of devices [3]. Besides protocol design itself, the implementation of such protocols is also a complex task which must address several issues to achieve the best performance in all environments.

In this paper, we propose a novel protocol for end-to-end content delivery in wireless networks that adopts the aforementioned characteristics. Our solution, named Raptor-based Content Delivery Protocol (RCDP), uses a Forward Error Correction (FEC) scheme based on Raptor codes [4] to achieve efficient data transmission in loss-prone environments. In particular, RCDP is oriented to wireless networks and wired/wireless mixed environments. In terms of implementation, we propose different architectural and design alternatives at both client and server that seek an optimal trade-off between throughput and resource consumption. Experimental testbed results show that an application-layer implementation of RCDP is able to achieve significant performance improvements compared to existing transport layer protocols.

The paper is organized as follows: section 2 reviews different protocols available in the literature that attempt to optimize content delivery in wireless environments. Section 3 explains how the proposed RCDP protocol works, including its main tasks and components, and identifies possible improvement strategies. Section 4 details the different improvements that were implemented, discussing the complexity of these changes, and how they could affect the overall performance. In section 5, we thoroughly evaluate the different implementation alternatives to assess their performance benefits; afterwards, we compare our solution against other existing transport protocols. Finally, section 6 concludes the paper.

2 Related Work

Developing efficient content delivery protocols for wireless environments is an issue that has received much attention from the research community. Several performance studies have been done to evaluate such solutions in terms of both design and implementation. The works available in the literature can be split into four different groups [5]: link-layer solutions, split-connection solutions, TCP-enhancements, and FEC based solutions.

In terms of link layer solutions, protocols like Snoop [6] and Tulip [7] attempt to improve the performance of higher layer protocols by making the link-layer aware of on-going connections. Split-connection approaches, like Mobile TCP [8] and Wireless-TCP [9], attempt to improve TCP performance in wireless environments by dividing the TCP connection into two separated connections. Both

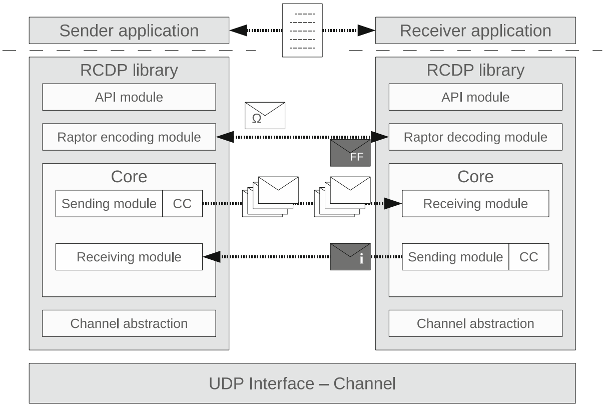


Fig. 1. Simple RCDP implementation diagram

link layer and split-connection approaches require performing changes in some of the network's intermediate elements, which can be a drawback when attempting to deploy them.

Concerning those solutions that enhance the TCP protocol, TCP Westwood [10] offers substantial performance improvements in wireless networks with lossy links compared to legacy protocols such as TCP SACK and TCP Reno. Mo-TCP [11] is another solution addressing heterogeneous wireless networks by adjusting its operations according to underlying link and network conditions. Both are end-to-end solutions, avoiding any changes at intermediate network elements.

Concerning content delivery solutions that explicitly adopt Raptor codes FEC, Luby et al. [12] propose a solution for reliable file delivery over 3GPP mobile broadcast networks, using Raptor codes to offer Multimedia Broadcast and Multicast Services (MBMS). Chiao et al. [13] describe the experience of using the FLUTE protocol for file delivery over a WiMAX unicast network. However, notice that these Raptor based approaches rely on unidirectional file pushing to clients, being applicable to multicast and broadcast scenarios without bidirectional communication requirements.

Our proposal differs from the previous ones by taking a completely novel approach. In particular, we rely on Raptor-based FEC to provide a unicast content delivery solution that offers reliable bidirectional communications (like TCP) while completely avoiding packet retransmissions (unlike TCP). Thus, no windowing or retransmission control has to be performed. Instead, the proposed solution relies on bandwidth estimations at the endpoints to perform rate control based on the end-to-end congestion state.

3 The RCDP Protocol

RCDP is a content delivery protocol that was developed focusing on wireless network scenarios such as ad-hoc networks. Thus, it focuses on environments characterized by low end-to-end throughput, high error rates and high retransmission delays. To avoid poor performance under these conditions, it relies on a Forward Error Correction strategy, known as Raptor encoding [4], to ensure that any piece of information sent can be successfully recovered at the final destination, even when part of that information is lost. The method to send encoded data is the following: data is split into blocks, typically bigger than 1 MByte, which are coded separately. Each block is split into symbols whose size is typically made equal to the maximum size that fits into a single packet; these symbols are then used as input to a two stage encoding process. In the first stage, where input data are referred to as *source symbols*, pre-coded symbols are generated. Then, through an arithmetic combination of these pre-coded symbols, an infinite number of encoded symbols can be generated. To recover information at the destination, any combination of the original symbols and the recovery symbols allows retrieving the original information. In fact, for the most recent version of the Raptor libraries [14], the probability of successfully decoding a total of r symbols received is:

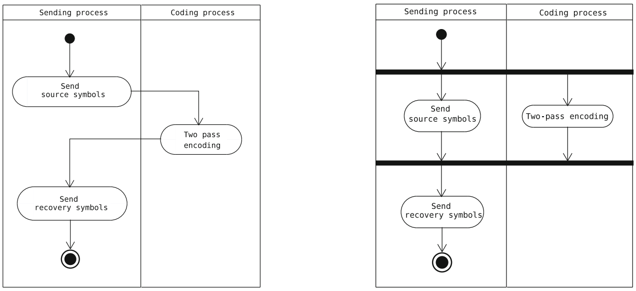
$$P_{dec} > 1 - 10^{-2(r-k+1)},\ r \geq k \tag{1}$$

This means that, to recover a source block with symbol size k, the probability of a successful decoding is greater than 99% if k encoded symbols are received, greater than 99.99% if $k + 1$ encoded symbols are received, and greater than 99.9999% if $k + 2$ encoded symbols are received.

The RCDP protocol is full-duplex, encompassing both sending and receiving processes. At the sender side, the contents requested are partitioned and encoded as stated above. Afterward, each symbol is placed in a single packet and sent to the receiver. Notice that, in order to aid the receiver in the decoding tasks, all the information required to recover a block is available in the header of each packet, namely the number of source symbols, the symbol identifier, the block identifier, and the real block size. Symbols are delivered in a continuous flow until the receiver is able to recover the original information; such event is notified to the sender with a control packet. When this control packet is received by the sender, delivery of the next data block begins. Since control packets are generated periodically, the loss of such a packet will be recovered by the next control packet successfully received.

The sequence of actions taken by the receiver are complementary to those of the sender: the receiver is continuously listening to incoming symbols, storing them in memory upon arrival. When enough symbols to recover a block have been received, the receiver sends back a *successful block recovery* notification and starts the decoding procedures; this strategy allows the sender to switch to the following block as soon as possible.

Since the proposed strategy does not adopt the *sliding windows* approach, flow and rate control tasks can easily be made independent. In our solution, flow control operates on a block basis, as described above, while rate control is made on a packet basis. In particular, rate control relies on end-to-end estimations of available bandwidth based on packet arrival patterns. The sender continuously



(a) Sequential process. (b) Parallel-like process.

Fig. 2. Proposed coding process enhancements

adjusts the transmission rate according to these available bandwidth estimations, generating packet trains [15] towards the receiver. This requires all packets to be marked with a train identifier, as well as recording the arrival time for each packet at the receiver. Bandwidth estimations are made based on information about arrival times for the first and the last packet of a train, along with the number of packets received per train. This estimation is sent back to the sender for every train ID, allowing it to dynamically adapt its transmission rate. Notice that this technique seamlessly adapts to network congestion since congestion will cause the time between consecutive packets in a train to increase, thereby decreasing the bandwidth value in the estimations made.

Packet trains are generated as follows: starting from the bandwidth estimation described above, the sender applies a β correction to this value, obtaining a target data rate ($\bar{R}$) that is slightly lower than the estimated bandwidth (BW_e):

$$\bar{R} = \beta \times BW_e, \; 0 < \beta \leq 1 \tag{2}$$

The available bandwidth value must be mantained to avoid saturating the channel. Notice that higher β values imply that the network will be working close to saturation. In order to detect when the bandwidth available increases, packets are grouped in a train, and the train rate (T_r) is defined as:

$$T_r = \frac{1}{\alpha} \times \bar{R}, \; 0 < \alpha \leq 1 \tag{3}$$

In this equation, parameter α allows adjusting the degree of burstiness. In particular, lower values for α are synonym of lower inter-packet times. Notice that T_r will be always higher than, or equal to, the target data rate ($\bar{R}$), being the latter the average data rate for each train period. To make sure that $\bar{R}$ is mantained (on average), all packet trains are followed by a pause period (inter-train time).

4 Implementation Issues

One of the main problems of Application Layer FEC techniques (AL-FEC) is the high computational cost of the coding and decoding process. RCDP uses Raptor codes which, although offering a linear cost in their coding algorithms, still requires a very efficient implementation for the proposed solution to be efficient as a whole. Moreover, since we adopt a user-level development approach, there are additional delays associated with the switching between kernel and user modes that do not appear in kernel level approaches, and whose effects should be mitigated.

4.1 RCDP Design: Initial Approach

In a first step, we have designed a solution offering the basic functionality required for the RCDP protocol to be operative. Using the UDT library [16] as a starting point, RCDP was created following the architecture shown in figure 1. The first module (API) acts as an interface between top level applications and the services offered by the library. The second module acts as a data encoder and decoder. The sending and receiving modules are responsible for rate control purposes, acting as described in section 3. Finally, RCDP includes a channel abstraction module which simply sends and receives packets to and from an UDP pipe.

To optimize the performance of the solution presented above, we have to tackle several issues. The first one is related to the coding module. Since we are using systematic Raptor codes [4], the first output symbols from the encoder are the source symbols themselves, and so no pre-processing for this first set of symbols is required, meaning that they can be sent without actually requiring any encoding to take place. However, the Raptor encoding process is mandatory to create the recovery symbols. Therefore, a delay between the first (source) and the second (recovery) set of symbols is introduced. To optimize this sequence of tasks, we reconfigure the coding process so as to eliminate the delay between the two sets of symbols; this strategy avoids introducing periods when no packets are sent. The diagrams shown in figure 2 depict the evolution from the original approach, which relies on sequential sending and coding processes (see figure 2a), towards a parallelized solution (see figure 2b), where partitions indicate that both sending and coding processes are performed in parallel.

The first approach to implement this optimization is to split the coding process into smaller slices, interleaving them with the delivery of source symbols. However, this approach could introduce additional problems related to the regularity with which the system is able to deliver packets due to the high CPU usage and low granularity level at this point. Another approach is to optimize the buffer's size. If we consider the encoding process as an irregular injection of symbols to be sent, instead of using two periods of symbol generation followed by an intermediate pause, we can use a buffer to regulate symbol generation to the lower layers. In this case, the only parameter that must be correctly tuned is the buffer's size. We must ensure that the transmission time of the buffered packets will be greater than the coding time to avoid starvation at the queue level, as shown in equation 4:

$$B_s \geq \frac{T_c \cdot BW}{P_s} \tag{4}$$

where B_s is the minimum size that the queue buffer should have, in number of packets, T_c is the block coding time, BW is the maximum bandwidth that the channel can achieve (in bits per second), and P_s is the packet size (in bits).

A second issue that must be considered is related to the timing accuracy for the packet generation process. The proposed solution to this problem relies on a two phase approach, where, in the first phase, a timed wait corresponding to a fraction of the sleeping time takes place. In a second phase, the last part of the idle period is a busy waiting.

A third element prone to optimization is the instant when the source is warned about the correct decoding of the current block. By default, this occurs only when the block decoding procedure is successfully completed. However, since the Raptor libraries provide feedback about the viability of the decoding process even before this process starts, the receiver can warn the sender about it much earlier, thus avoiding wasting time and network resources by preventing the generation of additional recovery symbols when they are no longer required.

In summary, the baseline optimizations introduced are the following: (i) buffer size tuning to regulate symbol generation, (ii) increased packet injection time accuracy, and (iii) early feedback from the receiver about successful decoding. A prototype of RCDP encompassing these optimizations, tagged as *RCDP+BO*, will be compared against other alternative solutions in section 5.

4.2 Multithreading Support

In this section we propose a performance improvement strategy that exploits parallel processing.

We will use two independent threads to code data blocks in parallel to overlap two different coding processes, thereby avoiding idle periods in the network. To achieve this goal, when a first block is being sent, the next block starts being coded. Due to this early load of the next block, the sending process will be improved by parallelizing all the management structures. The ability of decoding up to two different block is also introduced and this can be used, as parallel coding, to get a decoder ready to work without delay while the previous block is being decoded.

When several threads are working cooperatively, as in the aforementioned cases, processing overhead associated with thread management can become a problem. As in all processes whose complexity is incremented, additional software overhead must be included. This introduces processing delays, which could downgrade performance compared to simpler, sequential implementations. Therefore, it is important to determine the optimal trade-off between parallelization and overhead to achieve maximum performance. These issues are evaluated in detail in section 5.

4.3 Design Optimizations

The parallel coding design introduced in the previous section allows generating symbols from two different data blocks, if needed. Such a combination of symbols could be particularly interesting for the period that begins when all source symbols are sent, and recovery symbols start being generated, since we do not know exactly how many recovery symbols are actually required. Thus, to prevent occupying the whole transmission medium with redundant information for a period equivalent to one round-trip time, we propose mixing symbols of two consecutive data blocks in order to perform a gradual transition from the first to the second block. Such a strategy should be particularly effective in the presence of high network delay since, in such environments, too much time is wasted sending unnecessary recovery symbols. Thus, by mixing recovery symbols from the current block with source symbols from the next block, we ensure that at least part of the information sent during such period is useful to the receiver.

For evaluation purposes we developed the RCDP Mixed Blocks implementation (RCDP+PED+MB), a solution able to mix recovery symbols of the current block and source symbols of the next block following the strategy described above. In particular, the recovery vs. source symbols ratio is set to an arbitrary value (X:Y), where X is the number of source symbols and Y is the number of recovery symbols. The ratio can be optimized to different packet loss ratios, being that the ratio should decrease when channel losses are higher. Finally, in the unlikely event that transmission of source symbols of the second block is completed before the first block is recovered, only recovery symbols for the first blocked are generated to promote an ordered recovery of transmitted blocks.

5 Performance Evaluation

In this section we will study the performance trade-offs offered by the different protocol implementation strategies defined in section 4. The three RCDP enhancements under analysis are the following:

- RCDP+BO: This implementation contains a buffer size large enough to avoid interruptions during the sending process, as explained in section 4.1. This implementation also includes the two phased process described in that section, and anticipates the delivery of *successfully recovered block* notifications.
- RCDP+PE+BO: A prototype which combines the previous optimization and the parallel encoding described in section 4.2.
- RCDP+PED+MB: Includes the enhancements described for the previous solution, as well as the parallel decoding optimization and the ability of sending, receiving, and storing mixed symbols from two different blocks, as described in section 4.3. The selected ratio of source vs. recovery blocks is 1:4.

To test the effectiveness of the different RCDP optimizations, we created a network black box that is able to emulate different wireless network conditions by

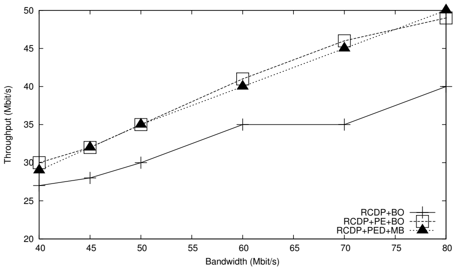


Fig. 3. Throughput vs. available bandwidth in a network with a 10 ms end-to-end delay (null error rate)

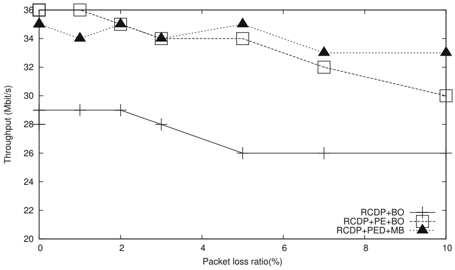


Fig. 4. Throughput vs. packet loss rate in a 50 Mbps channel with a 10 ms delay

varying available bandwidth, end-to-end delay, and packet loss conditions. Also, for each RCDP version, we implemented both server and client applications, where the latter requests a very large file to the former. We took ten samples for all the configurations.

5.1 Performance under Different Channel Conditions

Figure 3 shows the throughput achieved when varying the available bandwidth. We observe that, in general, the different RCDP versions tested behave as expected, experiencing an almost linear throughput increase as the available bandwidth increases. Second and third evaluated algorithms are able to achieve a higher degree of productivity compared to the former one. In particular, when compared to the first RCDP implementation, combining baseline optimizations with parallel encoding allows increasing throughput by about 45% in the best case, which is a very substantial improvement. Besides throughput enhancements, the block pre-charge (PE) technique described in section 4.2 enables the generation of a continuous symbol flow which contributes to a more regular transmission rate compared to RCDP+BO.

Figure 4 we shows the overall throughput when varying the packet loss rate. The desired behavior would show a linear throughput decrease for increasing packet loss ratios. We find that, in general, all the RCDP versions approximately

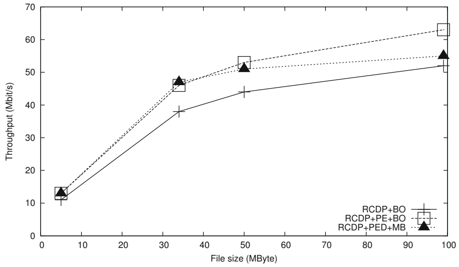


Fig. 5. Throughput vs. file size in a network environment with 50 Mbps bandwidth and a 10 ms delay (null error rate)

follow this trend. Similarly to the previous results, two well defined groups can be identified. Results show that, in general, error immunity remains similar to the original RCDP implementation, although actual throughput values basically depend on the different enhancements proposed, as explained above.

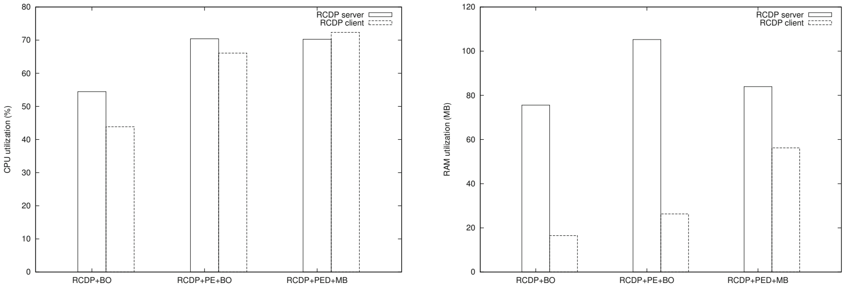
Figure 5 shows the throughput when varying the size of the delivered content. Notice that, when the file size is small, the average throughput is low because the initial startup time overhead is similar to the transmission time. A similar effect occurs with TCP as well. For greater files sizes, the impact of the startup times on throughput become negligible.

A different trend is observed due to the software complexity of the different solutions. In particular, the block management strategy for first implementation is more lightweight (fewer threads, fewer program instructions to execute, lower memory usage) than for the other implementations; therefore, in a high memory and CPU demanding software such as Raptor coding, it has an impact in terms of achieved throughput.

5.2 Resource Consumption Analysis

We now focus on the computational resources required by RCDP and the proposed optimizations. The results were obtained using the Linux “ps” tool as a background process, which was in charge of periodically measuring the CPU utilization (CPU time used divided by the time the process has been running) and RAM utilization at both client and server. Notice that, despite the software running on both client and server is similar, and despite communication is bidirectional, data is being transferred from server to client, meaning that the server will be mostly performance data encoding tasks, while the client will be performing decoding tasks instead. Thus, for the baseline RCDP implementation, the server CPU load is about 40% greater than the load at the client.

Figure 6a.) shows the additional CPU overhead of the different RCDP enhancements. At the server side, notice that the CPU utilization increases significantly when parallel coding is adopted due to the threads sinchronization overhead (see section 4.2). At the client side, although the CPU load is usually quite lower than the server load in absolute terms, the differential analysis



(a) CPU utilization. (b) RAM utilization.

Fig. 6. Resource utilization for RCDP and the different optimizations proposed

of the different RCDP versions shows that the CPU overhead increases significantly when some of the proposed improvements are adopted. In particular, load becomes higher for the solution adopting parallel coding. This is because enhancements are mostly oriented to alleviate CPU stress at the server side.

Focusing on RAM usage, Figure 6b.) show that the data structures required to support parallel coding cause memory consumption to increase. In particular, the last algorithm present the highest memory usage since both parallel coding and decoding are supported. Notice that, no matter which of the endpoints we focus on (client or server), both must perform coding and decoding tasks to fully support bidirectional communication.

Overall, results show that, despite both client and server share the same protocol architecture, the emphasis on either encoding or decoding tasks results in different behaviors. In all cases, the requirements and complexity of Raptor encoding impose more overhead on the server compared to the client: about 40% in terms of CPU, and between 25 and 70 MB in terms of RAM usage.

5.3 Performance Comparison against Different Transport Layer Solutions

To complete our analysis, in this section we assess the effectiveness of the proposed design and implementation optimizations for RCDP against different versions of TCP.

In our tests we use well-known TCP variants (TCP Reno [17], TCP Vegas [18], TCP SACK [19], TCP FACK [20]) for reference, as well as other solutions like Mobile TCP [11] and TCP Westwood [10], which specifically address wireless channels by discriminating channel-related losses from congestion-related losses.

Figure 7 shows the results obtained in our testbed. The available network bandwidth is of 50 Mbps, and delay is set to 10ms. The α and β parameters for all RCDP versions are set to 0.7 and 0.9, respectively. A set of experiments were performed in order to select these α and β values, which lead the system to show the best behaviour.

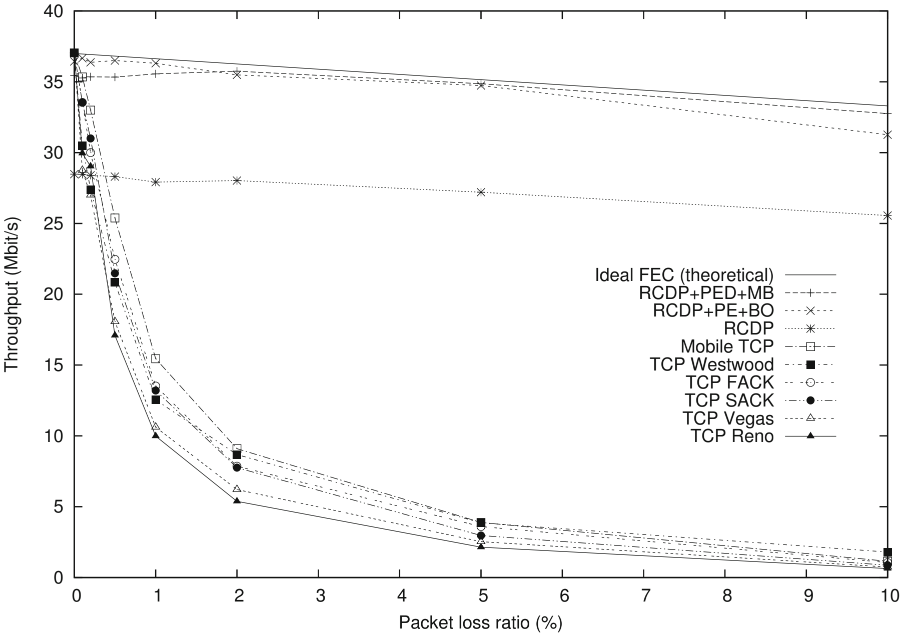


Fig. 7. Throughput performance of RCDP compared to different TCP-based solutions

From figure 7 we observe that, although offering different levels of error resilience, the different TCP-based solutions are quite sensitive to packet losses, mostly due to (i) the need of retransmitting lost packets, and (ii) using sliding windows of variable size to adjust the data rate in the presence of loss. Thus, when packet loss increases, the throughput associated with the different TCP variants drops quickly, experiencing a significant decrease compared to the *no loss* situation. Notice that the most noticeable differences between the different TCP versions occur in the loss range from 0% to 1%, as expected. Contrarily to TCP, the throughput values remain mostly immune to loss for all RCDP versions, being sustained near the maximum value, and only decreasing by a 15% in the worst case. In fact, we find that both RCDP+PED+MB and RCDP+PE+BO present performance levels that are comparable to those of an ideal (theoretical) FEC solution. The figure also shows that, compared to the original RCDP implementation, the chosen RCDP optimizations allow improving throughput by about 25%.

6 Conclusions

The delivery of large contents in wireless environments, such as ad-hoc networks, can be a complex task due to the different sources of loss inherent to these networks. In this paper we proposed RCDP, a novel solution offering reliable unicast content delivery that operates at the application layer. To achieve high reliability and efficiency while avoiding packet retransmissions, RCDP relies on Raptor codes, an AL-FEC solution that is highly effective at recovering missing data and that allows generating as many recovery packets as required.

When compared to transport layer solutions, we find that RCDP achieves much more stable throughput values in the presence of loss, approaching the performance of an ideal solution. In particular we find that, when the packet loss rate increases up to 10%, RCDP only experiences a throughput decrease of 15% in the worst case, while TCP based alternatives experience a throughput decrease of up to 95%.

Experimental results showed that parallelization of the encoding and decoding stages, along with the fine tuning of buffer sizes and other optimizations of the RCDP approach represents different design alternatives, which tackle the different trade-offs between complexity and efficiency in order to maximize end-to-end throughput.

Overall, we consider that RCDP introduces a new paradigm in the field of wireless communications, being a very attractive solution for reliable content delivery in environments such as ad-hoc networks due to its highly efficient use of available bandwidth resources. Additionally, since it operates in an end-to-end basis, no changes to the network infrastructure are required.

As future work we plan to develop a version of RCDP for the OMNeT++ simulator in order to test RCDP's performance in vehicular network environments.

Acknowledgments. This work was partially supported by the *Ministerio de Ciencia e Innovación*, Spain, under Grant TIN2011-27543-C03-01 and by the *Ministerio de Educación*, Spain, under the FPU program, AP2010-4397.

References

1. Fu, Z., Zerfos, P., Luo, H., Lu, S., Zhang, L., Gerla, M.: The impact of multihop wireless channel on TCP throughput and loss. In: INFOCOM 2003: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies, vol. 3, pp. 1744–1753. IEEE (2003)
2. Parvez, N., Mahanti, A., Williamson, C.: An analytic throughput model for TCP NewReno. IEEE/ACM Transactions on Networking 18(2), 448–461 (2010)
3. Bhoedjang, R., Ruhl, T., Bal, H.: User-level network interface protocols. Computer 31(11), 53–60 (1998)
4. Shokrollahi, A.: Raptor codes. IEEE Transactions on Information Theory 52(6), 2551–2567 (2006)
5. Balakrishnan, H., Padmanabhan, V., Seshan, S., Katz, R.: A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Transactions on Networking 5(6), 756–769 (1997)
6. Balakrishnan, H., Seshan, S., Amir, E., Katz, R.: Improving TCP/IP performance over wireless networks. In: Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, pp. 2–11. ACM (1995)
7. Parsa, C.: TULIP: A link-level protocol for improving TCP over wireless links. In: IEEE Wireless Communications and Networking Conference, WCNC 1999, pp. 1253–1257. IEEE (2002)

8. Brown, K., Singh, S.: M-TCP: TCP for mobile cellular networks. ACM SIGCOMM Computer Communication Review 27(5), 19–43 (1997)
9. Sinha, P., Nandagopal, T., Venkitaraman, N., Sivakumar, R., Bharghavan, V.: WTCP: A reliable transport protocol for wireless wide-area networks. Wireless Networks 8(2/3), 301–316 (2002)
10. Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M.Y., Wang, R.: TCP westwood: end-to-end congestion control for wired/wireless networks. Wireless Networks 8, 467–479 (2002)
11. Akbar, M.S., Ahmed, S.Z., Qadir, M.A.: Performance Optimization of Transmission Control Protocol in Heterogeneous Wireless Network during Mobility. IJCSNS International Journal of Computer Science and Network Security 8(8), 70–80 (2008)
12. Luby, M., Watson, M., Gasiba, T., Stockhammer, T., Xu, W.: Raptor codes for reliable download delivery in wireless broadcast systems. In: 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006, vol. 1, pp. 192–197 (January 2006)
13. Chiao, H.-T., Li, K.-M., Sun, H.-M., Chang, S.-Y., Hou, H.-A.: Application-Layer FEC for file delivery over the WiMAX unicast networks. In: 2010 12th IEEE International Conference on Communication Technology (ICCT), pp. 685–688 (November 2010)
14. Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T.: RaptorQ Forward Error Correction Scheme for Object Delivery, Internet Engineering Task Force, Internet Draft draft-ietf-rmt-bb-fec-raptorq-00, Work in progress (January 2010)
15. Jain, R., Routhier, S.A.: Packet trains - measurement and a new model for computer network traffic. IEEE Journal on Selected Areas in Communications 4, 986–995 (1986)
16. Gu, Y., Grossman, R.: UDT: UDP-based data transfer for high-speed wide area networks. Computer Networks 51(7), 1777–1799 (2007)
17. Stevens, W.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. Internet Engineering Task Force, RFC 2001 (January 1997)
18. Brakmo, L., O'Malley, S., Peterson, L.: TCP Vegas: New technique for congestion detection and avoidance. In: Proceedings of ACM SIGCOMM 1994 (August 1994)
19. Mathis, M., Mahdavi, J., Floyd, S., Romanow, A.: RFC2018: TCP Selective Acknowledgement Options. RFC Editor United States (1996)
20. Mathis, M., Mahdavi, J.: Forward acknowledgement: refining tcp congestion control. SIGCOMM Comput. Commun. Rev. 26, 281–291 (1996)

Impact of Different Content Placement and Delivery Strategies on Content Delivery Capacity of the Wireless Mesh Networks

Milenko Tošić[1], Mirko Ćirilović[1], Ognjen Iković[1], Daniel Kesler[2],
Staniša Dautović[2], and Dragan Boscovic[12]

[1] La Citadelle Inzenjering Novi Sad Serbia
[2] Department of Power, Electronics and Communication Engineering, Faculty of Technical Sciences, University of Novi Sad, Serbia
{milenko.tosic,mirko.cirilovic,ognjen.ikovic,
dragan.boskovic}@lacitadelleing.com,
{keslerd,dautovic}@uns.ac.rs

Abstract. Wireless mesh networks are introduced as a way for providing cheap, easily deployable and maintainable access to core network's services. The content delivery capacity of these networks directly depends on the number of installed wireless mesh network gateways. In this paper, we present analysis of impact of content placement on nodes of wireless mesh network on the total streaming capacity of that network. Our analysis shows that network's streaming capacity, with respect to given topology and cached content, can be equally increased with content placement on access points of the network, as it would be increased with installation of additional gateways.

Keywords: wireless mesh network, content delivery network, content placement, total streaming capacity.

1 Introduction

Internet has evolved to be the most significant information exchange medium. Its widespread usage and utility is not limited to the business related applications only, but is equally used for facilitating social and human interactions. The achieved success has burdened Internet with even higher expectations in terms of performance and support requirements for new applications and services. Equally, technology advances in video capturing and conditioning, for consumption on personal and portable devices, have made it become the most dominant traffic type on modern internet in support of "richer immersive experiences" [1]. In order to enable service providers to cope with the new challenges and limitations of the current Internet, Content Delivery (Distribution) Networks (CDNs) were introduced [2]. CDN networks deploy a constellations of dedicated servers strategically placed across the Internet geographical footprint. The burden of hosting the produced content is offloaded from content providers to CDN service providers. Content is distributed across CDN servers called Replica Servers (RS). The distribution is conducted in a way which tends to maximize the cache hit ratio at replica servers (based on distribution of users' requests). Distributing the content closer

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 302–315, 2012.

to the end users (edge of the network) will result in decreased load on content origin servers, offload of traffic from the core segments of the network (better load balancing and resource utilization) and increase in QoS provided to the end users through reduced delay and jitter. Typical CDN system is shown in Fig. 1. First, different Content Delivery Regions (CDRs) are defined based on identified profiles of the users and their request distribution for available content. Next, dedicated replica servers are installed at every CDR. Based on the dominant users' profiles in different CDRs, different content will be distributed from the source server to available replica servers [2]. CDN management system is responsible for organization of distributed replicas and content distribution strategies.

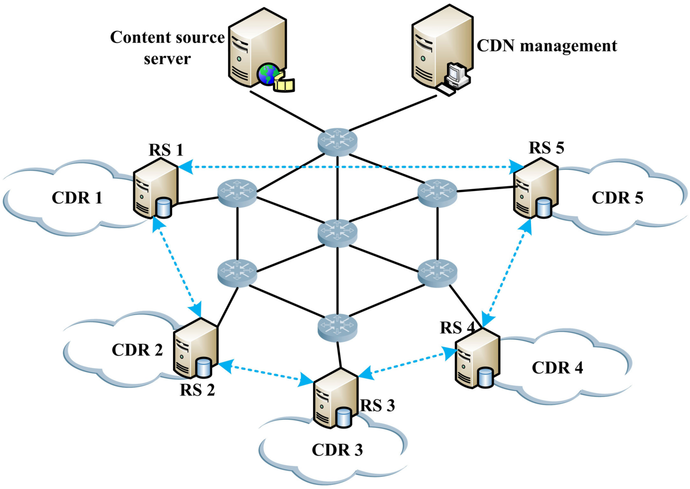


Fig. 1. Example of a CDN system

The CDNs have provided the service providers with temporal solutions for improvements of streaming capacity over the Internet. Now, mobile users are becoming dominant consumers of the content provided over the Internet. Therefore, wireless internet access is the new challenge for the content delivery systems. These access networks introduce new limitations and challenges regarding streaming capacity of the infrastructure.

Wireless Mesh Networks (WMNs) are introduced as a way for providing cheap, easily deployable, maintainable and scalable Internet access. Nevertheless, high transfer throughput is a must for WMNs in order to successfully compete with other last-mile wireless technologies. Maintaining a high transfer throughput in WMNs is difficult because these networks, as all other wireless networks, are subject to interference due to open nature of the wireless medium. Radio interference can affect WMN performance in several ways [3-4]:

- Higher hops count from source to destination is likely to lower the overall path's throughput since hops may need to share radio resources.
- IP packet loss on a given path is proportional to the number of hops and consequently lowers the TCP throughput.

- Many mesh paths congregate on a single WMN gateway (GW) thus increasing probability of interference in vicinity of the gateways.

One of the ideas behind future CDN systems is to continue bringing the content closer to the end users. WMN Access Points (APs) are infrastructure nodes which are closest to the end users. Therefore, content placement on these network nodes will bring the content as close to the end users as possible, thus providing the highest QoS to them. This content distribution approach can also result in increased level of WMN bandwidth resource utilization, which, on the other hand, increases the overall streaming capacity of the WMN. There are several approaches for content placement on WMN nodes and content delivery from these nodes to the requesting users. In this paper we will analyze impacts of these different approaches on total content delivery capacity of the WMN and Average Delivery Tree Length (ADTL).

Rest of the paper is organized as follows. First, work related to CDNs, content centric networking and content delivery over WMNs will be presented. Then, three different approaches for content placement and delivery over WMNs will be introduced. Finally we will present detailed analysis of these three approaches.

2 Related Work

CDNs are introduced as a collection of servers with full or partial replicas of content stored on origin server. When local request for the content cannot be fulfilled from the dedicated server, it will be forwarded to the origin server. Next step was introduction of hierarchical replica servers [2]. Several hierarchical layers are formed in CDN system, and user's request for content is forwarded from lower layers to the upper ones and all the way to the origin server, in case of the cache miss for particular content. These content delivery approaches resulted in poor storage resource utilization, because, in order to achieve lower number of cache misses, the same content had to be replicated over as many replica servers as possible [2]. The next evolutionary step was introduction of peer to peer (p2p) communication among CDN replica servers. This allowed CDN servers to address each other's cache misses (dashed lines in Fig. 1). As the users' requests became more demanding, CDN systems had to include more and more dedicated replica servers and bring the content closer to the end users. This trend led to proposal of hybrid CDN solutions, incorporating p2p content delivery among end users into the standard CDN framework. There are many systems proposing this type of CDN solutions [5-9]. However, the main challenge in these solutions is management of end user's resources and content stored on their devices. The security risks, guaranteed resource availability, users' incentives and highly distributed nature of the system are still limiting the practical implementation of these solutions. The next step is introduction of highly distributed CDN clouds which will completely squeeze out the dedicated servers [10]. However, problems related to hybrid CDNs will need to be addressed.

The other idea, which promises to solve the majority of the future internet related problems, is the concept of content centric networking (CCN) [11-12]. Its governing philosophy argues that a communication network should support user's focus on the content needed without the need to specify physical location from where to get the requested content. Solutions proposed by this approach will provide mechanisms for efficient utilization of all networking nodes capable of content caching (not only servers and end user devices, but also routers, switches and access points). Enabling WMN APs to store and stream multimedia content will lay down a solid basis for implementation of CCN services on top of these networks. Including the WMN access points as replica servers into the content delivery system can provide the following benefits:

- Content will be brought to the end point of the infrastructure, thus providing the best level of QoS (decrease in start-up delay, decreased probability of jitter effects and increased throughput) to the end users.
- Content delivery capacity of the WMNs will be increased without the need for additional GWs and topology changes.
- Service and network providers will have the full control over the caching equipment and content placed on them, which will significantly lighten the CDN management system when compared to the approaches proposing content placement on end users' devices.
- Delivering multimedia content from networking devices (APs), which work all the time (whether or not they are included into the content delivery process), will result in significant decrease in power consumption of the system [13-14] when compared with standard CDN approach, where dedicated servers are installed only for the purpose of content delivery.

There are previous research efforts towards proposing the content placement on WMN nodes. Work described in [4] addresses the problem of content placement on APs of a WMN, and shows how this approach can result in significant improvements of WMN's bandwidth resources utilization, which increases the overall content delivery capacity of the underlying WMN. However, authors in [4] have focused their work on proposing and evaluating the new content caching techniques in WMN nodes, rather than investigating how different content placement strategies and WMN topologies/configurations impact the CDN system. In this paper we will provide in depth analysis on how different content placement/delivery approaches, combined with different WMN configurations, impact the overall streaming capacity of the WMN.

3 Different Content Placement/Streaming Techniques in WMNs

In this section we will define three different techniques of content placement and delivery over WMNs. First, we will present a content delivery method based on a standard CDN approach, where all of the users' requests are fulfilled from a dedicated content server located in the core network (see Fig. 2a). Analysis of this method will show why the WMNs are considered as bottlenecks for content delivery. Next, two methods for streaming of content cached on WMN nodes will be analyzed. By placing content on WMN nodes, we are following a CCN paradigm for including the networking nodes into the caching and streaming processes. We will analyze approach when content, which is stored on WMN APs, can only be delivered to the requesting users who are directly connected to these WMN APs (see Fig. 2b). Finally, we will introduce and analyze content streaming solution where content cached on one WMN AP can be streamed to users connected to other APs in the WMN (see Fig. 2c). In this approach, a collaborative (p2p based) content delivery among WMN nodes is enabled.

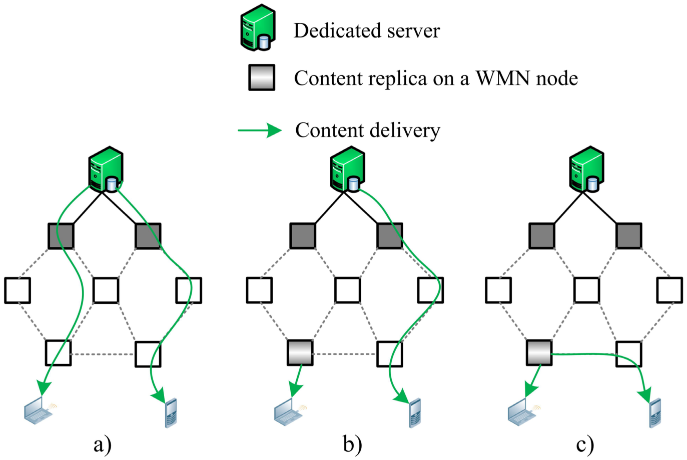


Fig. 2. Different content placement/delivery approaches

For every of the three aforementioned content delivery approaches in WMNs, we have conducted a detailed analysis. Different WMN topology setups and different selections of nodes for content placement/delivery are investigated. This in depth analysis has shown the impact of different system setups on overall content delivery/streaming capacity of the WMNs. The streaming capacity of a WMN is defined as a maximal number of end users who can be served with the selected load at the same time. Analytical analysis is conducted in MathWorks MatLab and Wolfram Mathematica software packages with custom built simulator and mathematical model. Mathematical model is built for solving the problem of optimal context aware content placement on WMN nodes in light of Average Delivery Tree Length (ADTL) minimization [13]. Different system setups for different WMN graphs are analyzed. Besides investigating the impact of these system setups on streaming capacity of WMNs, analysis is also conducted in order to show how these streaming techniques and system setups impact the ADTL. WMNs in our experiments are presented as network graphs. All experiments are conducted on network graphs corresponding to well designed WMN networks. This means that network graph is connected, nodes have degrees less than some threshold (in our experiments threshold was 6), the number of leaf nodes (with degree equal to one) is limited to 5 percent of total WMN node count and there is maximally one edge between any two nodes in the network graph. For the sake of clarity of the paper's presentation we will focus description of our analytical approach on one fairly simple WMN network graph depicted in Fig. 3.

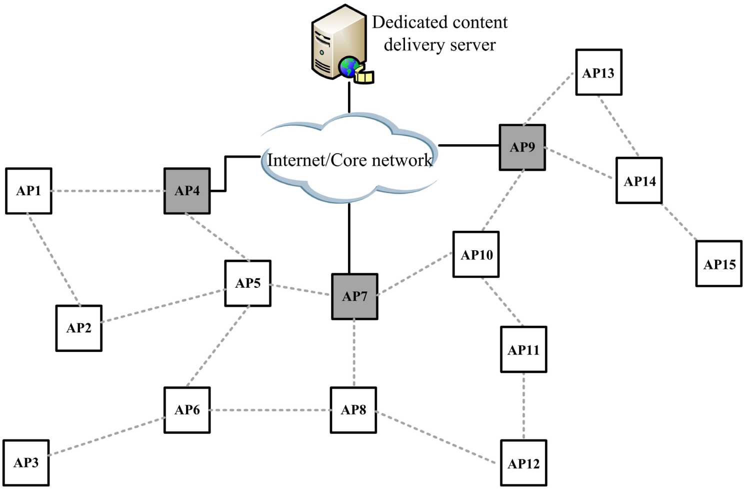


Fig. 3. WMN network for detailed analysis (gray nodes- WMN GWs)

Shaded nodes in the Fig. 3 are WMN GWs. Different selection of gateways will be shown in the analysis. Gray dashed lines represent all possible connections among WMN nodes. It is considered that every WMN node has two radio interfaces for communication with other WMN nodes (backhaul connections) and one interface for providing access to the end users. A standard 802.11 based WMN is considered. Backhaul communication is done by using the IEEE 802.11a protocol which can provide around 30Mbit/s of user's traffic. Access to the WMN APs is provided over the 802.11g protocol which also provides 30Mbit/s for users' data. If more than two backhaul connections are shown for one WMN node in the network graph, these wireless links will share the total bandwidth provided by two radio interfaces (2x30Mbit/s). We will not address the problems of channel interference and its impact on achievable throughput. 30Mbit/s is considered on every established link between two radio interfaces (if the radio interfaces are not shared with other active links). We will also consider that cable backhaul has an infinite capacity when compared to the capacity of the wireless links (backhaul cable network will not be considered as bottleneck for content delivery). Although our conclusions are made under ideal conditions (no TCP related problems, routing and packet forwarding works perfectly and no interference), the logic behind them is valid for realistic WMN deployments as well. This is because we have focused on impact of WMN topology and delivery node selection changes, which will have the same effect in realistic WMNs as in the simulator's environment.

Now, we will present all three content delivery solutions and their detailed analysis on the WMN topology presented in Fig. 3.

3.1 Content Delivery from a Dedicated Server

The typical WMNs have a specific networking principle where all of the traffic to and from end users has to pass through limited number of WMN GWs [2-3]. The number of these GWs greatly dictates the capacity of the network. Less GWs means more economical deployment of the WMNs, because less network cable installation needs to be provided. However, the smaller the number of WMN GWs is, the capacity of the network will be limited to the value below that which can be provided by the underlying communication technology (i.e. IEEE802.11). If the number of WMN GWs increases, the network will come closer to Wireless Local Area Network (WLAN) in light of capacity, but in the light of installation and maintenance costs as well. Therefore, a proper WMN engineering requires careful selection of GW nodes within the WMN topology.

When the content streaming for all end users is done from a dedicated server located in the core side of the network, the entire load needs to be transferred over the WMN GWs. These nodes will become a bottleneck for content delivery service over the WMN. Therefore, for this content streaming approach, the number and position of GWs within the WMN's topology will have the greatest impact on the total content delivery capacity of the system. In Fig. 4a is shown how the content can be delivered over three WMN GWs depicted as gray shaded rectangles. The topology (AP numbering and location) from Fig. 3 is valid for WMNs presented in Fig.4.

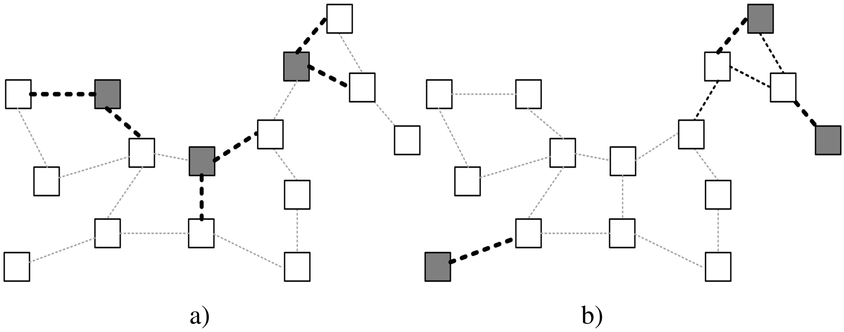


Fig. 4. Content delivery over WMN with three GWs with: a) proper GW selection; b) improper GW selection

Since every GW has two radio interfaces for communicating with its neighbors, streaming capacity which can be achieved over three GWs presented in Fig. 4a equals the sum of available throughput over six backhaul links. Since every GW is also a WMN AP, users directly connected to GWs can be served with the requested content without burdening the backhaul links of the WMN. Therefore, Total Streaming Capacity (*TSC*) of the WMN in Fig. 4a can be calculated as:

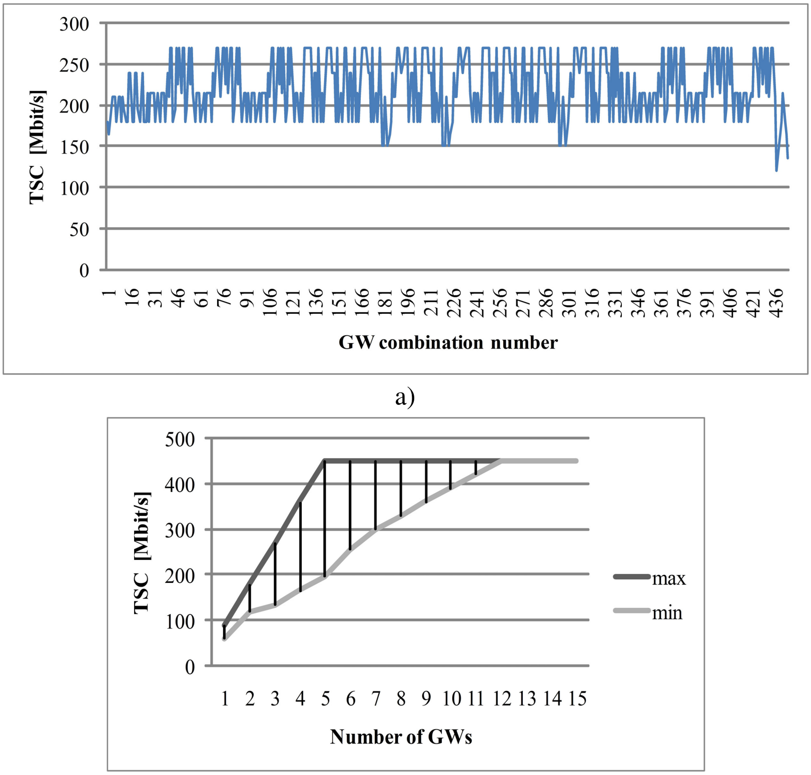
$$TSC = N_{GW} * (2*C_B + C_A) \quad (1)$$

where N_{GW} is the number of GWs, C_B is the throughput capacity of backhaul links and C_A is capacity of access interface on WMN nodes. For the system parameters introduced in section 3, *TSC* of the WMN shown in Fig. 4a, according to (1), equals 270Mbit/s. If every user requests 2Mbit/s, then 135 users can be served with the requested throughput. However, (1) is not universal in this type of content delivery over WMNs. For example, presented equation cannot be applied when one WMN

GW node has only one neighbor. In Fig. 4b is shown a WMN topology configuration where two GWs have only one neighbor. These GWs cannot use both of theirs backhaul radio interfaces and therefore available backhaul capacity is drastically affected. Also, AP13 and AP15 as GWs (see shaded rectangles in Fig. 4b and node numbering in Fig. 3) are too close one to another and the local backhaul link topology becomes congested before backhaul radio interfaces of both of these nodes are maximally utilized. AP14 is receiving 30Mbit/s over the backhaul link AP15-AP14. The other backhaul radio interface can be used to transfer 15Mbit/s of throughput from AP13 to AP10 over AP9. This is because all of the AP14's users can be satisfied from AP15, therefore additional deliveries from AP13 will not be used by users in the coverage area of AP15, AP14 or AP9 (users of this AP are satisfied from AP13). Therefore, the remaining backhaul radio interface of the AP14 will be shared among links AP13-AP14 and AP14-AP9, which means that average of 15Mbit/s can be achieved over these two links. The *TSC* of the WMN in Fig.4b is 195Mbit/s. These results show how different GW locations can impact the *TSC* of the WMN. There are 455 different combinations of 3 GW nodes in the WMN with 15 nodes. The minimum achievable *TSC* with 3 GWs for the WMN shown in Fig. 4 is 135Mbit/s for i.e. GW combination: AP13, AP14 and AP15. When the optimization goal for GW selection is maximization of *TSC*, equation (1) is valid for all optimal GW placements. Other GW placements which are not resulting in the maximal achievable value for *TSC* are considered sub-optimal. The value of *TSC* for all combinations of three GWs for the WMN topology shown in Fig. 3 is shown in Fig. 5a.

In Fig. 5b is shown how the *TSC* value changes with the number of WMN gateways (both minimal and maximal *TSC* value). For lower numbers of GWs, the maximal achievable *TSC* with respect to the increasing number of GWs, is following a linear dependency (in line with the equation (1)). However, after certain threshold in the number of GWs (for the WMN shown in Fig.4 it is five GWs), the maximal *TSC* cannot be increased with introduction of additional GWs. This effect is due to WMN topology and the fact that wireless backhaul links are used to transfer traffic to and from GWs. With certain number and combination of GWs in the WMN topology, capacity of their backhaul interfaces will be enough to satisfy access capacity of all remaining APs, making introduction of additional GWs unnecessary. Therefore, the same maximal *TSC* can be achieved with i.e. 5 GWs as in the case when all WMN nodes are configured as GWs (equivalent of WLAN). If the values of minimal *TSC*s are considered, than 12 GWs in the WMN showed in Fig. 4 can achieve the same *TSC* as equivalent WLAN for all possible GW combinations.

Different GW selections will have different ADTLs. This parameter will directly impact QoS provided to the end user (increased ADTL is directly proportional to increased delay). For example, if there is only one GW in the WMN showed in Fig. 3, then optimal GW placement (with respect to minimization of the ADTL) is AP7 (ADTL equal to 3.07), while AP15 achieves ADTL equal to 5.2 (uniform distribution of users' requests among all WMN nodes is considered). Therefore, optimal GW placement can be subject of two optimization criteria, one being maximization of *TSC* while the other is minimization of ADTL. As the number of GWs in the WMN rises, ADTL will decrease and when every WMN node is configured as GW, the ADTL



a)

b)

Fig. 5. TSC values a) for all combinations of three GWs, b) as function of the number of GWs

will have the value of 1. We haven't considered cable links for the ADTL, since delay over these links can be neglected when compared to delay of the wireless links. Fig. 6 depicts ADTL value decrease variations when an additional GW is added into the WMN topology. ADTL values are given for every additional GW (from AP1 to AP15 – AP7 excluded as it is the ultimate GW). The results clearly demonstrate dependency of the ADTL value relative to the GW location within a given WMN's constellation. If AP12 is reconfigured into a new GW (in addition to the existing GW-AP7) ADTL decreases by 0.2. However, if AP14 is configured as a new GW, this will result in an ADTL reduction by 0.6.

3.2 Delivering of Content Which Is Placed on WMN Nodes

In this analysis we are considering a content streaming approach where content can be placed on WMN nodes and streamed only to the requesting users directly connected to these nodes. Content is still delivered from the dedicated server in the core side of the network (as shown in Fig. 2b). WMN APs with the stored content will be able to deliver that content only to their direct users and thus the *TSC* of the WMN will be

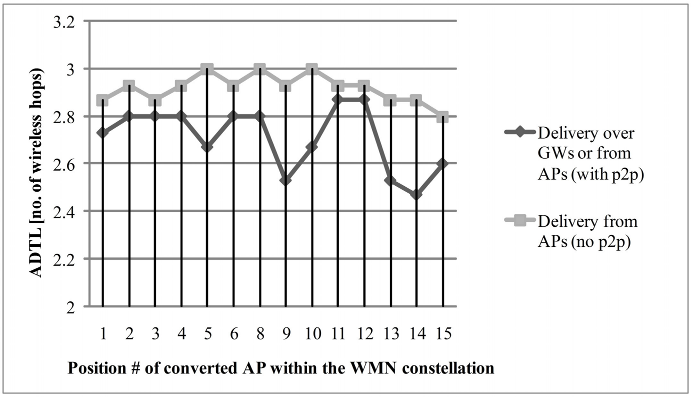


Fig. 6. ADTL variations when a specific AP within the analyzed constellation is reconfigured to act as an additional GW or a streaming server

increased. In this analysis we will consider a WMN with three GW nodes and we will show how content placement on different number and combination of WMN APs impacts the *TSC*.

We will start with the WMN configuration as shown in Fig. 7a. AP4, AP7 and AP9 are WMN GWs (shaded grey) and AP1, AP6, AP8 and AP14 (node labeling from Fig. 3) have the content of interest cached in their memory (depicted with patterned gray shade). The thicker dashed lines in Fig. 7a depict content delivery. When compared to Fig. 4a it is clear that streaming of the content from WMN APs increases the *TSC* for the amount of maximal throughput achievable in the access of the WMN AP. Therefore, if we are placing the content on either of the WMN APs, apart from the currently selected GWs, we can expect the increase in the *TSC* which is equal to the $N_{AP}*C_A$, where N_{AP} is the number of WMN APs with content placed on them, and C_A is the throughput capacity of the technology used for providing network access to the end users.

Fig. 8 shows how *TSC* of the WMN showed in Fig. 7a (for this GW combination we have showed that *TSC* equals 270Mbit/s) changes with the increasing number of APs on which the content is placed. For the WMN GW configuration showed in Fig. 7a, when we place content on six APs, we can achieve the *TSC* equal to that of the equivalent WLAN. In the previous content delivery approach, for achieving the same effect we have needed at least two, carefully selected, additional GWs. Installation of new GWs into the existing WMN deployments can require significant expenditures. Therefore, a much more economical solution would be to place a required content on WMN APs which will, as shown, increase the *TSC* of the WMN and, in addition, bring the content closer to the end users, thus decreasing ADTL and providing higher level of QoS. ADTL value variations as a function of file copy placement on particular AP are shown in Fig. 6. Placing content on any of the APs within the WMN would have less impact on ADTL decrease than converting that particular AP into an additional GW. Under assumption that users' requests are uniformly distributed among all APs, the ADTL will drop more significantly if the content is placed on an AP farther away from the existing GWs. For example, placing copy of a file on AP15 will result in most noticeable ADTL reduction relative to the situation when every request is fulfilled over the ultimate GW (AP7).

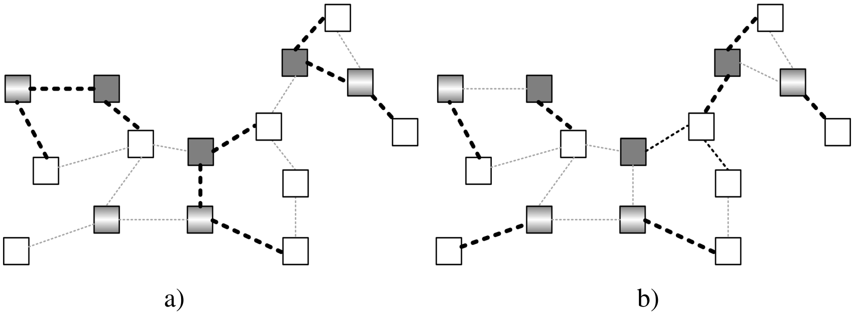


Fig. 7. Content streaming from WMN APs: a) APs can stream the content only to the directly connected users; b) p2p between APs is enabled

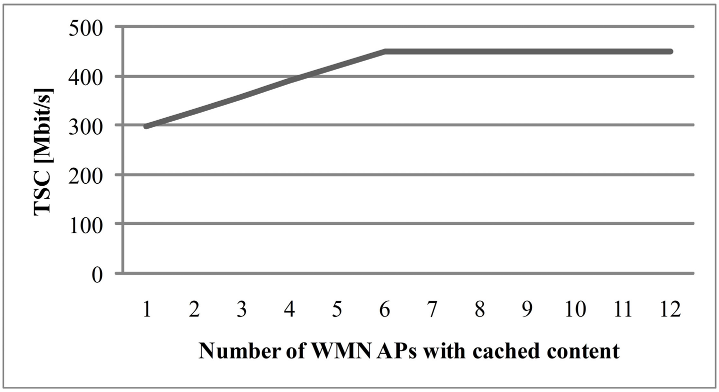


Fig. 8. Maximal *TSC* values for different number of WMN APs with cached content

If the content placement on APs follows a context aware strategy, which takes into account the number and spatial distribution of users' requests, then delivery from WMN APs to directly connected end users will ensure significant increase in *TSC* and decrease in ADTL in the real deployments of WMNs. However, with additional management and modifications to WMN APs it is possible to achieve content delivery strategy presented in the next sub-section.

3.3 Content Placement on WMN APs with Collaborative Content Delivery Enabled

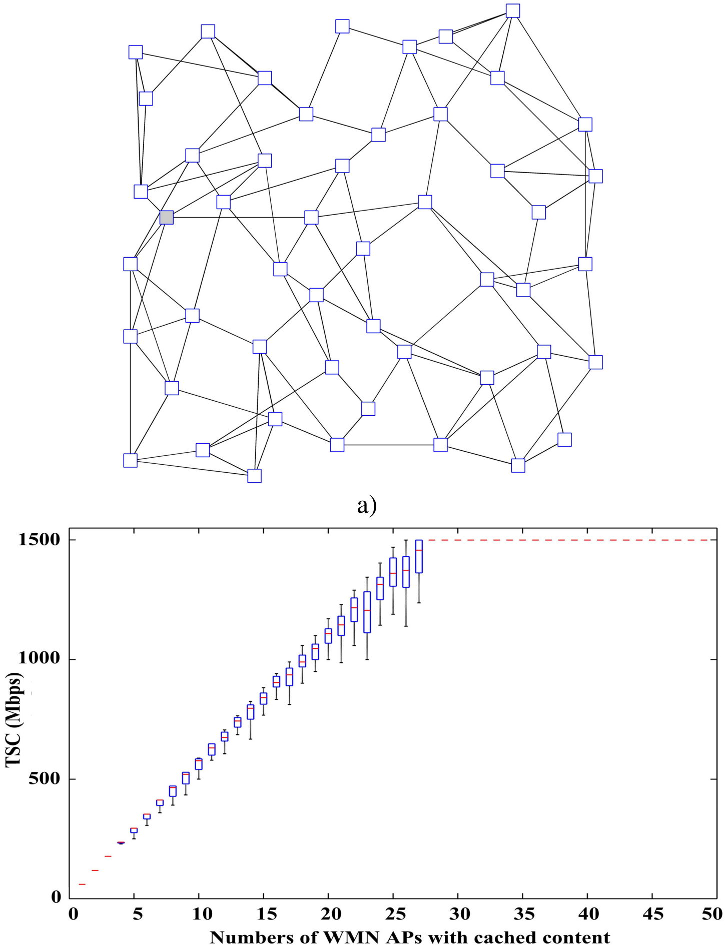
Collaborative content delivery among WMN nodes is defined as the ability of the involved WMN nodes with stored content to collectively address the cache misses for the content requests originated from users connected to the other WMN nodes. When the requested content is not stored on the WMN AP to which the requesting user is connected, CDN management system will instruct the optimal WMN AP, which has the content of interest in its cache, to deliver the content to the requesting user. The impact of this content delivery approach on *TSC* will be illustrated on the WMN configuration presented in Fig. 7b.

The GW placement is the same as in the previous approach. This GW placement is one of the optimal placements (as shown in sub-section 3.1) with respect to *TSC*. In the Fig. 7b, the content is placed on AP1, AP6, AP8 and AP14 (the same as in sub-section 3.2). However, these nodes are able to deliver the content to the end users directly connected to them, as well as to the end users connected to other APs. Therefore, these APs have exactly the same impact on the *TSC* as adding the additional GWs. The results from the chart in Fig. 5b are valid for this content delivery approach as well. In this case, the minimal and maximal *TSC* values for four GWs can be achieved in WMN from Fig. 7b with three GWs and content placed on one AP. The optimal AP selection will result in maximal value of *TSC* and other APs can provide lower *TSC*. Now, we have achieved *TSC* increase equivalent of that achieved with additional GWs. Since introduction of new GWs into the existing WMN is a very demanding job (with respect to costs and time consumptions), this content delivery approach presents the best solution for providing CDN over existing WMN. Also, content placement on WMN nodes represents more scalable solution for *TSC* increase than adding new GWs. By different content placement strategies on WMN APs, streaming capacity increase can be provided where and when needed.

Presented content delivery solution has the same impact on ADTL of the cached content (content cached on WMN APs), as introduction of additional GWs into the system (see Fig. 6).

In order to confirm the conclusion presented in this section, we present the same analysis for the WMN consisting of 50 APs. In this topology (see Fig. 9a), there is only one WMN GW. All WMN APs support collaborative content delivery. Content placement on different number of WMN APs and their combinations is evaluated. Number of APs with stored content is increased by one in every experiment until all APs have content stored on them. For every selected number of APs, 40 different combinations of content placement are inspected. This gives the total of 2000 (50 nodes x 40 combinations) results for *TSC* of the WMN presented in Fig. 9a. Box-plot diagram for *TSC* results is shown in Fig. 9b. In this diagram, we can see that maximal and minimal *TSC* values follow the same rule as in case where the number of WMN GWs and their combinations are gradually changed (results in Fig. 5b). The same results, as those shown in Fig. 9b, are achieved for 50 nodes WMN (see Fig. 9a) when the number and combination of GWs is changed, thus confirming our conclusion that content placement on WMN AP, with enabled collaborative (p2p) delivery, has the same potential to impact *TSC* increase (with respect to given topology and content which is cached on WMN APs) as reconfiguration of that AP into the new GW.

When compared with the second content delivery strategy (where content stored on WMN APs can only be delivered to directly connected end users), the ability of WMN APs to collaborate in order to address each other's cache misses, will provide maximal *TSC* with less copies of the content locally stored on WMN APs. This will provide better utilization of storage resources on WMN APs enabling local caching of broader spectrum of different content.



a)

b)

Fig. 9. A bigger instance of the *TSC* determination problem: a) WMN topology composed of 1GW and 49 APs (50 nodes in total) and b) box-plot diagram of *TSC* results for different number and combinations of WMN APs with stored content and p2p delivery enabled (beginning with 27 APs, calculated *TSC* values for all experiments are constant and the same, resulting with the absence of *box plots*)

4 Conclusion

In this paper we have provided detailed analysis of three different content placement and delivery strategies relative to WMN centric media streaming. First we have analyzed content delivery from a dedicated server in a core side of the network. We have shown how different number and combinations of WMN GWs impacts total streaming capacity for this delivery strategy. Next, we have analyzed approach where content is placed on individual APs and can be delivered only to the end users directly connected to these specific APs. This approach contributes less to the overall *TSC* than the one which can be achieved with addition of new GWs into the WMN topology. Finally, we have presented the content placement/delivery approach in which content is pre-placed across WMN APs and collaborative delivery among APs is enabled. Analysis of this approach demonstrates the same gains in terms of the *TSC* increase as the first one, in which APs are reconfigured into the GWs. Therefore, the same *TSC* can be achieved with no need for additional GWs, which consequently means lower delivery cost and better scalability (collaborative delivery increases *TSC* when and where needed).

Acknowledgment. Research presented in this paper is partially funded by the EU grant no. FP7-ICT-257385 - OneFIT (www.ict-onefit.eu). La Citadelle Inzenjering is a member of the project's consortium. The views expressed in this document do not necessarily represent the views of the complete consortium. The Community is not liable for any use that may be made of the information contained herein.

References

1. Future Media Internet Research Challenges and the Road Ahead. Whitepaper of Future Media Internet-Task Force (April 2010)
2. Khan Pathan, A.M., Buyya, R.: A Taxonomy and Survey of Content Delivery Networks. Technical Report GRIDS-TR, University of Melbourne (April 2010)
3. Wu, X., Liu, J., Chen, G.: Analysis of Bottleneck Delay and Throughput in Wireless Mesh Networks. IEEE Mobile Ad-hoc and Sensor Systems, 765–770 (2006)
4. Dogar, F., Phanishayee, A., Pucha, H., Ruwase, O., Andresen, D.: Ditto- A System for opportunistic Caching in Multi-Hop Wireless Networks. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom 2008). ACM, New York (2008)
5. Coppens, J., Wauters, T., De Turck, F., Dhoedt, B., Demeester, P.: Design and Performance of a Self-Organizing Adaptive Content Distribution Network. In: 10th IEEE/IFIP Network Operations and Management Symposium, Vancouver, pp. 534–545 (2006)
6. Skevik, K.A., Goebel, V., Plagemann, T.: Evaluation of a Comprehensive P2P Video-on-Demand Streaming System. Computer Networks 53, 434–455 (2009)
7. Lee, C.N., Kao, Y.C., Tsai, M.T.: A vEB-Tree-Based Architecture for Interactive Video on Demand Services in Peer-to-Peer networks. Journal of Network and Computer Applications (33), 353–362 (2010)
8. Wah Yim, A.K., Buyya, R.: Decentralized Media Streaming Infrastructure (DeMSI): An Adaptive and High-Performance Peer-to-Peer Content Delivery Network. Journal of Systems Architecture (52), 737–772 (2006)
9. EU grant no. FP7-ICT-216217: Next Generation Peer-to-Peer Content Delivery Platform (P2P-NEXT), `http://www.p2p-next.org/` (last visited on February 24, 2012)
10. Pathan, M., Broberg, J., Buyya, R.: Maximizing Utility for Content Delivery Clouds. In: Vossen, G., Long, D.D.E., Yu, J.X. (eds.) WISE 2009. LNCS, vol. 5802, pp. 13–28. Springer, Heidelberg (2009)
11. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N., Braynard, R.: Networking named content. In: Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies, pp. 1–12. ACM, Rome (2009)
12. Borcoci, E., Negru, D., Timmerer, C.: A Novel Architecture for Multimedia Distribution Based on Content-Aware Networking. In: 3rd International Conference on Communication Theory, Reliability, and Quality of Service, pp. 162–168 (2010)
13. Boskovic, D., Vakil, F., Dautovic, S., Tosic, M.: Greening of Video Streaming to Mobile Devices by Pervasive Wireless CDN. Journal of Green Engineering (2), 1–27 (2011)
14. Valancius, V., Laoutaris, N., Massoulie, L., Diot, C., Rodriguez, P.: Greening the Internet with Nano Data Centers. In: Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies, pp. 37–48. ACM, Rome (2009)

Bonjour Contiki: A Case Study of a DNS-Based Discovery Service for the Internet of Things

Ronny Klauck[1] and Michael Kirsche[2]

[1] Innovations for High Performance Microelectronics (IHP)
Leibniz-Institute for Innovative Microelectronics, Germany
klauck@ihp-microelectronics.com

[2] Computer Networks and Communication Systems Group
Brandenburg University of Technology Cottbus, Germany
michael.kirsche@tu-cottbus.de

Abstract. With the integration of everyday objects and sensors into the Internet, users gain new possibilities to directly interact with their environment. This integration is facilitated by the development of tiny IP stacks that enable a direct Internet connection for resource constrained devices. To provide users with the same level of usability that is predominant in the current Internet infrastructure, a self-configured discovery service for sensors and objects is needed. We thus present a use case of a discovery service based on Multicast DNS and DNS Service Discovery, which we adopt for resource constrained devices and operating systems. Applications using this service can realize direct connections between resource constrained devices following the end-to-end principle of the IP-based Internet, allowing for a seamless integration of potentially millions of objects and sensors into the current Internet and facilitating the pervasive infrastructure that is envisioned by the Internet of Things.

Keywords: Internet of Things, Discovery, mDNS/DNS-SD, Contiki.

1 Introduction

The Internet is rapidly reaching over into the physical world with the integration of sensor networks and the embedding of communication technology in everyday objects. This technological progress was named *"Internet of Things"* by Kevin Ashton [1]. The realization of the IoT vision goes hand in hand with the development of Internet Protocol (IP) solutions for embedded devices. With the development of small IP stacks like uIP(v6) [2], embedded and resource constrained devices can be connected directly to the Internet, while at the same time omitting approaches that break with the Internet's end-to-end principle.

With an integration of everyday objects and sensors into the Internet, a new kind of pervasive and ubiquitous infrastructure will be available, leading to new types of applications and services for the interaction of users with their environment. To facilitate a seamless integration, appropriate solutions must be compliant with the current Internet infrastructure. There are two prerequisites to achieve this: a standardized discovery scheme complying with the IP standard

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 316–329, 2012.

of the conventional Internet domain and the self-configuration ability to handle the possibly large number of objects emphasized by the IoT vision [3]. Discovery, in our work, covers the topics of finding and addressing objects as well as issues of interoperability between different device classes. Self-configuration, as an underlying paradigm of device management, covers the problems of scalability and bootstrapping. Both aspects are vital to provide the same level of service quality that users and developers are accustomed to from the conventional Internet. We therefore want to facilitate solutions that provide autonomous bootstrapping and service discovery instead of complex manual setups.

A main problem for a seamless integration lies in the restrained nature of embedded devices. Typical limitations are: communication over short ranges and failure prone wireless links, limited power supply, and limited memory sizes. Another problem is the multitude of device types, whereas each type has different characteristics, limitations, and physical communication abilities, leading to the need for bridging solutions to enable a device-overlapping interconnection. Next to different communication technologies, embedded devices often use specialized and proprietary protocols on the higher layers of their communication stack. Examples are the Constrained Application Protocol (CoAP) [4] and the Message Queue Telemetry Transport (MQTT) [5] protocol, both rely on gateways and protocol translators to connect embedded devices to Internet-based systems. This leads to a loss of flexibility and end-to-end functionality because messages need to be translated [3], which is typically a time-consuming, failure-prone, and complex task. So instead of trying to introduce new protocols for the discovery of applications and services offered by everyday objects, we propose to adapt established protocols that work in compliance with the current architecture for the integration of *Things* into the *Internet of Things*.

In this work, we introduce a lightweight discovery service implemented for the Contiki [6] operating system for embedded devices. Our approach is based on the combination of *Multicast DNS* (mDNS) [7] and *DNS Service Discovery* (DNS-SD) [8] in a lightweight implementation with adjustments for embedded devices (e.g., small code footprint, minimized overhead), while enabling interoperability and service discovery at the application layer through DNS messages. mDNS and DNS-SD are combined with *Zeroconf* [9] under the term *Bonjour* [10], an established and widely used standard in current IP-based networks. While Bonjour enables computers to communicate ad hoc without a complex setup or manual bootstrapping, our approach will be a first step in this direction by integrating everyday objects and sensor devices in the current IP-based Internet architecture in order to enable the discovery of devices and services of resource constrained devices in compliance with the current infrastructure as a basic discovery service for the Internet of Things vision.

The remainder of this work is structured as follows: Section 2 introduces our case study together with a discussion of related problems. Section 3 presents a service-oriented solution to discover and address sensor devices. A verification through a prototypical implementation is presented in Section 4. Related work is discussed in Section 5 and concluding remarks are presented in Section 6.

2 Case Study for Locating Sensors over IP

An autonomous discovery of devices and services inside a network domain is a central usability criterion. Applications and users need mechanisms to identify new devices and gather relevant information to access offered services. For Internet of Things scenarios, this means that devices need to be discovered in wired and wireless networks, with support through pre-configured infrastructure as well as possibly without (ad hoc) any infrastructure support. Next to these requirements, a discovery solution should also work in compliance with current systems. We thus favor using IP-based solutions to facilitate an easy integration into current networks and systems.

We choose our own working environment as a testbed for a typical IoT scenario. In our "smarter workplace" scenario, different devices (sensors as well as actuators) are distributed to support workers in their daily routines, as Figure 1 illustrates. This leads to a case comparable to common "smart home" scenarios. Smart homes are widely covered in research, although typically used technologies are proprietary and not IP-enabled [11], thus requiring gateways and protocol adapters. As we omit protocol gateways in favor of an end-to-end connection with common Internet protocols, we focus on setting up a testbed of different devices connected over IP links. We connect various stationary computers and workstations, mobile devices (e.g., smartphones, netbooks), as well as embedded devices, sensors, and actuators over their respective communication technologies. Figure 1 depicts the architecture of the system. It is important to note that we cannot omit bridges to interconnect different technologies physically. We assume that one of the following bridging methods is provided to interconnect diverse communication technologies: interconnection via USB/serial port, support for multiple radio technologies within a single device, or embedding the sensor/device in power outlets for a connection over power line communication.

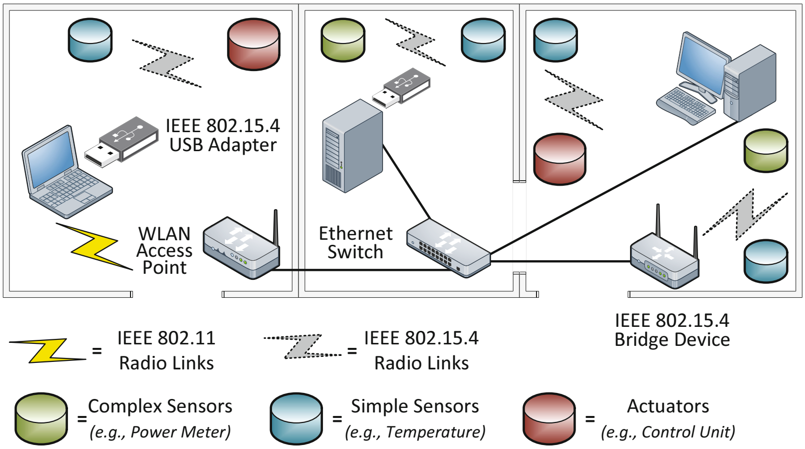


Fig. 1. System Architecture and Use Case

The discovery of non-constrained devices (e.g., workstations, smartphones, netbooks) and their respective services over IP links is provided through today's standards like *Bonjour*, which we adopt for the depicted resource constrained sensors and actuators. For the bridging of different communication technologies, we currently work with USB adapters to link IEEE 802.15.4 and Ethernet over intermediate devices like notebooks. A next step will be the practical deployment of a IEEE 802.15.4 bridging device in form of a small power outlet connected access point for IEEE 802.15.4, comparable to the *Sheevaplug* [12].

Autonomous discovery of new devices and services increases usability and user autonomy. We eventually want to reach a point where sensor devices are just plugged in (in power outlets) or started (if battery operated) and users can "search" the network for new services and then "subscribe" to them, gaining access to information and new ways of interacting with their environment. This operational scheme is directly connected to the concept of Service-oriented Architectures (SOA) [13]. SOA enables a seamless integration and interaction of different device types while specific devices are abstracted as services according to their offered functions. This abstraction of functionality into services is what we ultimately aim for with the discovery service for constrained devices.

Several problems arise due to the resource constrained nature of typical embedded sensor and actuator hardware as well as due to the non-permanent attachment of such devices to the network infrastructure. The following list is a selection of problems that we explicitly address in this work:

– Resource constrained devices operate on tiny 8-bit microcontrollers with limited memory (typically 8-16 KB of RAM and 64 - 128 KB of ROM);
– Network connection is often ad hoc and non-permanent in contrast to non-constrained devices due to a sensor's limited battery-driven power supply;
– Embedded communication standards (e.g., IEEE 802.15.4) support only small Maximum Transmission Units (MTUs) when compared to normal IP-based infrastructure (cp. 802.15.4 with 127 Bytes to IPv6 with 1280 Bytes),

We cover these problems in our solution design with several adaptations and optimizations. The following section presents the standards that we use and our adaptations while Section 4 presents a practical verification that specifically addresses the problems of embedded microcontrollers and limited memory.

3 mDNS / DNS-SD-Based IoT Discovery Service

The presented IoT discovery service is based on mDNS and DNS-SD. Both protocols were implemented for Contiki, an operating system for resource constrained hardware with a small code footprint and integrated IPv6 support. Our combination of Contiki, mDNS, and DNS-SD is named *uBonjour*. uBonjour enables a high interoperability between non-constrained and resource constrained sensor devices at the application layer through the use of standardized DNS messages. Our implementation is based on the *Ethernet Bonjour* [14] project. We extended and reimplemented it in an optimized way for Contiki with IPv6 support and tested it with *Avahi* as described in Section 4.

3.1 Background Information

This subsection presents an overview of Contiki, mDNS, DNS-SD, and the SOA concept to substantiate the subsequent descriptions of the discovery service for IP-based embedded sensor devices.

Contiki [6] is an open source operating system, running on embedded hardware with constrained memory and computing resources. IP connectivity is provided by the integrated *uIP* stack, which features ARP, IPv4/v6, SLIP (Serial Line IP), ICMP echo, UDP, and TCP protocol support. Contiki's core system is based on an event-driven kernel with on-demand preemptive multithreading to effectively share marginal memory resources between all processes. To provide concurrency, processes are implemented as event handlers that run till completion and return back to the kernel when finished. A process is implemented either as an application program or as a service. Services provide functions that can be used by application programs. Inter-process communication is provided through the kernel by posting events.

Multicast DNS (mDNS) [7] is one part of a group of standards that is used to enable computers to view or find other devices and to share their services with each other in network environments. mDNS's functionality is to resolve domain names without the help of any server by delivering messages to reserved multicast addresses `224.0.0.251` (IPv4) and `ff02::fb` (IPv6) and the UDP port `5353`. Devices inquire network addresses with requests to a multicast group, while the corresponding device responds with its list of DNS resource records (refer to *DNS-SD*). mDNS is often implemented together with DNS-SD. Both are available for various platforms, for example for Mac OS, iOS and Windows with Bonjour [10], and for Linux, BSD, OpenWRT and Android with Avahi [9].

DNS Service Discovery (DNS-SD) [8] is another part of the standards used to discover devices and share their services. It is combined with mDNS and also provided by Bonjour [10] and Avahi [9]. DNS Service Discovery enables the location and announcement of services of entities in a network domain. DNS resource records are again used to provide information about services. A device usually offers its service by propagating the following DNS records: a service name of the service offering device via the `SRV` record, a hostname/domain name mapped to an IPv4 address via the `A` record and optionally for IPv6 with the `AAAA` record, a user-defined text with the `TXT` record, and an assignment of service instances to a service with the `PTR` record.

Network configuration and management is simplified with mDNS and DNS-SD because entities can detect each other inside a network without previously distributed configuration or prior mutual acknowledgment [15]. Extending a network with additional devices is simplified due to the fact that all devices are able to explore their network vicinity for available services and running applications. An extension of this discovery functionality beyond local network boundaries is enabled by Wide-Area DNS-SD [16] using the same DNS-SD APIs. This can ultimately boost the integration of resource constrained devices into the current Internet infrastructure [17], thus supporting the IoT vision.

Service-oriented Architecture (SOA) is an approach to provide transparency through service abstraction for specific device functions as well as seamless interaction with different device types through self-organization and autonomous connection establishment and management. Transparency through SOA enables devices to browse their network domain for neighbor devices and newly published services [18]. SOA also enables an easier and failure-resistant network bootstrapping [19] because hard-coded start-up addresses and broken pre-configurations can be avoided. Devices perform look-ups for specific services after booting and request necessary information to perform their context-based actions [20]. SOA eventually enables a service-oriented network where devices can act and collaborate spontaneously and autonomously.

3.2 Standard Discovery Service

uBonjour is a lightweight service to discover and address devices and available services in network environments. Application protocols can register their availability as services in the network as well as discover other devices that use the same application protocol. uBonjour helps to fulfill two main goals of the IoT vision: standardized discovery and self-configuration. Figure 2 depicts the architecture of a Contiki-based sensor device including uBonjour as a general service for announcing available application protocols.

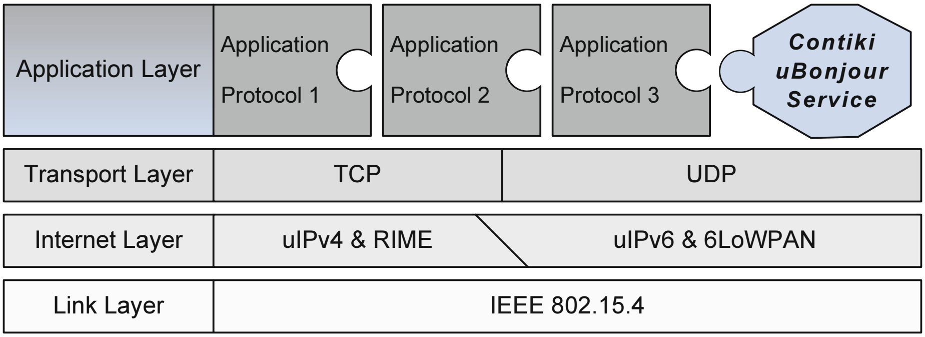


Fig. 2. uBonjour running as a Discovery Service on Embedded Devices

Standardized Discovery defines an application- and device-independent announcement for service offering entities in the network environment. It implements the standardized behavior of mDNS according to [7] and DNS-SD [8] to ensure a compliant message exchange with different kinds of computer systems using either Bonjour or Avahi. This enables computer systems to discover and address sensor devices in an easy-to-use and transparent way without application protocol gateways.

Self-Configuration is an essential aspect of the Internet of Things, as a large number of devices are going to be connected to the Internet. An automatic setup during the bootstrapping phase is necessary to support the diversity of sensor technology as well as configuration possibilities and to facilitate an easy start-up for the subsequent interaction between devices [21]. uBonjour therefore supports self-configuration instead of hard-coded addresses so that devices can scan their network environment and share results without the need to know the exact network topology. Adding a new sensor is performed as easy as starting and requesting information from surrounding devices, while a coordinated exit of a device is enabled with "service unavailable" messages.

3.3 Resolving Hostnames

To discover the address of another device in the network, a device needs to send a DNS query for the domain name to the multicast group. The device with the corresponding domain name replies with an A record including its network address. If a hostname could be resolved, uBonjour will broadcast an event to all listening processes with the IP address as data, or else a timeout for the query will be triggered. Only one name can be resolved at the same time. Sending a new query at the same time will stop the ongoing search for a hostname and will start a new search with the currently submitted hostname.

3.4 Discovering Services

A device initiates the service discovery by sending a `PTR` record to the multicast group containing the name of the searched service. If a service query is resolved, uBonjour posts an event to all processes with `PROCESS_BROADCAST`, containing the resolved IP address and port as data. If the query cannot be resolved, a timeout for the searched service is triggered. Only one service query is supported simultaneously. Sending a new query at the same time stops any ongoing service search and starts a new search with the currently submitted service name.

3.5 Registering, Removing, and Updating Services

To publish an available service, a sensor device has to send four DNS records as described in Section 3.1. Each application running on a device has to register a service with its service name, IP address (provided by Contiki), and port, if it wants to be found in the network by other devices. If a `PTR` query arrives, the corresponding device replies with one `SRV`, `TXT`, `A` or `AAAA`, and `PTR` record. To remove a service from a network, the device needs to send a `PTR` record with the Time-To-Live (TTL) set to zero. Our uBonjour API also supports updating an already published service by resending the four DNS records with changed data. uBonjour can handle up to eight service registrations per device by default. This value can be adjusted to the memory size of the specific device.

3.6 Memory Management Optimization

As constrained devices only support limited memory resources, reducing the code size and the number of used variables and buffers becomes very important.

A large quantity (about 60%) of uBonjour's source code size is consumed by the handling of received DNS records and by the generation of DNS responses. We thus optimized the memory management for this code part to minimize the memory consumption. The buffer size of the parser is reduced as the handling is now done directly inside the uIP buffer. The generation of DNS responses requires only a small buffer of the size of a DNS header while the rest of the message generator directly uses the uIP buffer. This *in-place processing strategy* facilitates a memory-efficient discovery service for Contiki.

3.7 Message Size and Flow Optimizations

Contiki's uIP stack uses lower layers (e.g., Rime for IPv4, 6LoWPAN for IPv6) and their provided features (e.g., fragmentation) to efficiently route IP packets in a network. An IP packet relies on the lower layer fragmentation skills, which again depend on the sensor device and its built-in radio transceiver. This limits the supported IP packet size of Contiki, as the following Table 1 shows.

Table 1. Supported IP Payload Sizes of Contiki 2.5 in Byte

Sensor Device & Radio Module	IPv4	IPv6
AVR Raven / Redbee Econotag	1300	1300
Tmote Sky / AVR ZigBit	108 / 240	240
MEMSIC IRIS / MICAz	128	240
STM32	140	140
MSB430	116	116
ESB	110	110
Zolertia Z1	108	140

Table 1 summarizes the maximum available payload sizes of an IP packet for each supported sensor device and radio module. If these values are exceeded, DNS records will not fit into a single IP packet and IP packet reassembly must be enabled, which will cost an additional amount of RAM and 700 Bytes of code size. Experiments for performing lower layer fragmentation have shown that this mechanism is energy-efficient for request/response cycles as there is no need to optimize the number of fragments [22, Sec. VI]. Devices like the AVR Raven or the Redbee Econotag can handle the lower layer fragmentation very well and thus support a higher IP payload size when compared to other devices (e.g., Tmote Sky, Zolertia Z1). We therefore decided that each DNS record of uBonjour must fit into a single IP packet to avoid the use of IP packet reassembly. The TTL flag in the DNS header needs to be set for each DNS record with a time in seconds to specify how long a published service will be available. A normal value in this case is 120 seconds, which should be increased for further optimization. Larger TTL values minimize the number of sent messages between devices and they do not interfere with the joining of new devices, because those can explicitly ask for available services in the network. Evaluations of the impact of increased TTL values on the data traffic in larger networks are future work.

Further optimizations for uBonjour are possible by implementing compression methods to keep the data traffic to a minimum, especially in multi-hop networks. Two different methods are currently available: the *Known-Answer Suppression* [7, Sec. 7.1] and the *Duplicate Question Suppression* [7, Sec. 7.3] method. The known-answer suppression method reduces the total number of answers while a device sends a response for a group of devices (including himself), thus reducing the number of necessary responses to gather information about the whole network. For this each device needs to cache published service offerings in the network and wait a randomly chosen time before it answers a request. If a device detects a cached answer to a request, then this answer will be added to its own. If other devices recognize this they will refrain from sending their own answers. The duplicate question suppression method, in contrast, reduces the total number of requests. A device will assume a request as its own when it sees a request that matches its own. This prevents the sending of redundant DNS responses because less `PTR` query messages are sent. Again, each device has to wait a random period before it can send its request.

At the moment we decided to refrain from choosing either one of these two methods for uBonjour because both optimizations need to store a bundle of message related data for proper functionality. This results in an increase of needed buffers and code size, therefore increasing the use of RAM and ROM for the discovery service. To avoid this, we developed our own optimization approach for uBonjour, which is introduced in the following section.

3.8 One-Way Traffic Optimization Approach

uBonjour should assist devices in finding available services and being discovered by other classes of computational devices inside a network. The implementation therefore must be as slim as possible to allow other applications to reside in the device's limited memory as well. Since the *Known-Answer Suppression* and the *Duplicate Question Suppression* method would consume too much memory (as described in the previous section), we developed an economical optimization of mDNS and DNS-SD for resource constrained devices called *One-Way Traffic* (OWT). The OWT optimization is built-in and can be activated during the compilation of uBonjour. This optimization puts a sensor device into a passive mode in which the device only publishes its services periodically (via TTL) and responds only to incoming name and service requests. Passive mode disables the active resolving of hostnames and the ability to parse service query responses from other devices in the network. This also avoids ping-pong effects of DNS responses, while service query responses are targeted only on non-constrained devices. The activation of OWT and the subsequent disabling of hostname resolving and service query response parsing reduces the used code size significantly and also saves energy because message parsing and network traffic are minimized overall. The OWT optimization leads to a reduction of lines of code too, since the parser handling for incoming service query responses can be skipped, which frees around 400 lines of code. We do not lose much of the core functionality of uBonjour because sensor devices are still able to actively register services and to react to requested services from desktop and mobile systems inside the network environment. Overall, this behavior facilitates the lightweight aspect of the discovery service by coupling non-constrained devices with resource constrained hardware. Desktop and mobile systems can scan their environment for sensor devices with a pre-installed mDNS and DNS-SD service, while nearby sensor devices can directly answer to them with DNS records, without the need of installing additional protocols or using application protocol gateways either. This establishes an easy-to-use discovery mechanism for consumers and offers a simple integration strategy for system administrators.

4 Evaluation

This section presents a verification and prototypical evaluation of the performance of our uBonjour solution in relation to our use case for both IPv4 and IPv6 in terms of memory footprint, message size and response time.

4.1 Experimental Setup

All experiments were performed with Contiki version 2.5 on Zolertia Z1 sensor hardware, which is based on a low-power MSP430F2617 microcontroller with 92 kB of ROM and 8 kB of RAM. The Z1 also provides an IEEE 802.15.4-compliant Chipcon 2420 RF transceiver. The IPv4 test setup consists of devices running uBonjour that are directly connected via the Serial Line Internet Protocol (SLIP) to a computer running Linux. The test setup for IPv6 uses a one-hop network with static routes. One Zolertia Z1 runs the 6LoWPAN border router (shipped with Contiki) connected again to a computer running Linux. The border router converts 802.15.4 / 6LoWPAN frames to Ethernet / IPv6 frames. Two additional Z1 complete the IPv6 setup by running uBonjour. The forwarding of mDNS messages to the Ethernet interface of the Linux PC is done by Avahi (pre-configured parameters were `enable-reflector=yes` and `allow-point-to-point=yes`). We monitor incoming mDNS packets with Wireshark[1] for both cases in order to verify the correctness of generated DNS records from the devices, to measure the DNS record sizes, and to monitor the interaction between the computer and the service offering devices. The response time was measured by sending a `PTR` record to the multicast group and stopping the time between this request and all four received DNS responses sent from the corresponding node.

4.2 Message Size

Avoiding the use of any kind of additional buffer was mandatory for a slim and memory-efficient implementation of uBonjour. The Zolertia Z1 device has one

[1] Wireshark Network Protocol Analyzer [Online] `http://www.wireshark.org/`

of the lowest IP payload sizes as depicted in Table 1. It is therefore a good reference platform to test if typical DNS messages (refer to [23, Sec. V-C]) fit into a single IP packet of the uIP stack. The DNS record length combined for all four kinds depends on the sum of the length of the submitted service name, the length of the text information in the TXT record, and the length of the used domain name. The total sum of these freely selectable parameters is *36* for IPv4 and *68* for IPv6 on the Zolertia Z1. The rest is reserved for the DNS header and the DNS message structure. If these measured values are exceeded for the Z1, DNS records will subsequently not fit into a single IP packet and IP packet reassembly must be enabled, which we want to omit as explained in Section 3.7.

4.3 Memory Footprint

uBonjour is realized in only 1450 lines of code. Table 2 shows the detailed memory footprint of uBonjour. The code is compiled with msp430-gcc (GCC) 4.4.5 for the Zolertia Z1. uBonjour with one service that may be registered requires 3.82 kB of ROM / 0.3 kB of RAM for IPv4 and 3.89 kB of ROM / 0.3 kB of RAM for IPv6. As mentioned in Section 3.8, the *OWT* optimization reduces the amount of used memory significantly, in our case it will be cut into half while the lines of code are reduced to around 1050. Each additional service registration for uBonjour will cost around 0.14 kB (IPv4) and 0.23 kB (IPv6) of RAM. These two values are both calculated from the total sum of freely selectable parameters for the DNS records as described in Section 4.2.

Table 2. Memory Footprint of uBonjour with / without uIP stack and OWT

uBonjour	**ROM in kB**	**RAM in kB**
IPv4 / IPv6	7.12 / 7.69	0.4
IPv4 / IPv6 OWT enabled	3.82 / 3.89	0.3
IPv4 / IPv6 with uIP stack	16.9 / 27.24	1.62 / 3.38
IPv4 / IPv6 OWT with uIP stack	13.6 / 23.44	1.46 / 3.22

The difference in memory consumption of uBounjour between IPv4 and IPv6 is only small because both variations just differ in the used IP address length (16 Byte for IPv6 versus 4 Byte for IPv4 addresses). Minimal larger buffers for sending and storing registered services are therefore needed with IPv6. Unfortunately, this behavior is not adopted by the uIP stack in general: the uIPv6 stack is three times larger in RAM and twice as large in ROM consumption when compared to its IPv4 counterpart. This means that for the Zolertia Z1 nearly half of the memory is allocated by the uIPv6 stack alone. A slim and memory-efficient implementation is therefore even more important for IPv6 then for IPv4.

4.4 Response Time

Discovering services with uBonjour takes 71 *ms* for directly connected sensor devices over SLIP (IPv4). In IPv6 scenarios a 6LoWPAN border router is needed for our use case, hence packets will always be delayed via one-hop. The measured response times for multi-hop are: 1233 *ms* for one-hop, 1954 *ms* for two-hop and 2324 *ms* for three-hop scenarios. No optimizations nor an active forwarding of DNS responses was implemented in uBonjour. Multi-hop routing is handled by Contiki's IP stack and depends on the performance of its used lower layers [24].

5 Related Work

Existing research related to our work can be divided into generic work in the area of Internet of Things architectures and IoT integration strategies as well as existing approaches that use either mDNS or DNS-SD for embedded devices. An approach to use web services over IP links to integrate sensor networks into the current IT infrastructure is presented in [25]. We share their idea of using IP links and Contiki but refrain from using web services. Bardin et al. [20] proposed a service-oriented component framework for the integration of devices and subsequent discovery of services inside heterogeneous networks with the help of a residential gateway. Our approach distributes discovery tasks directly to the devices, making residential and centralized gateways obsolete. The authors of [21] provide a general discussion and an overview of the idea of facilitating a service-based Internet of Things. They resort to a service-oriented yet complex middleware to enable the discovery of services.

Examples for practical mDNS and DNS-SD implementations are Bonjour [10] and Avahi [9], both widely used on desktop and mobile systems. Both are open source and written in C/C++, but too big to fit into the memory of constrained devices. A smaller implementation is Liaison [26], which is around 100 kB in size and written in C++. Porting one of these three implementations for resource constrained devices would be an extensive and time consuming task, because a complex refactoring with subsequent design restructuring is necessary to adapt the implementations for the requirements of constrained devices. [27] stated that they implemented and tested mDNS for Contiki with a memory footprint of only 1.0 kB of ROM and 0.5 kB of RAM, but there is no code proof available. A direct integration of mDNS for Contiki can be found online.[2] It offers an advanced version of the uIP hostname resolver functions and supports IPv4 and IPv6, but there is no plan to extend it with DNS-SD. The most promising mDNS and DNS-SD implementation with only 14 kB is Ethernet Bonjour [14] for the Arduino platform. It was written in C++ for the WIZnet chipset on the Ethernet shield by Georg Kaindl and supports only IPv4 for Ethernet frames. We reused the parser / message generator and its functions stub for uBonjour. C++ parts were ported to C and the WIZnet chipset related code was rewritten to use the uIP stack of Contiki. These measures ensure that uBonjour runs properly on Contiki with a minimized memory consumption.

[2] *darkdeep* Contiki Branch [Online] http://svn.deepdarc.com/code/contiki/trunk

6 Final Remarks

This work promotes a discovery service for the Internet of Things vision based on mDNS and DNS-SD. We showed that an existing standard can be used economically on resource constrained devices. Furthermore, uBonjour enables self-configured and autonomously acting sensors in a network environment while it avoids the need of hard-coded bootstrap parameters. Sensor devices can react more precisely on topology changes and on joining or leaving devices. uBonjour will help to integrate sensor devices seamlessly into the Internet infrastructure and also enable an easy access from commodity computer systems.

Through the implementation and evaluation of mDNS and DNS-SD for Contiki, we gathered new insights on implementing available and established protocols, which were originally designed for desktop systems, with a low memory and small code footprint for constrained devices. As a result of the implementation process, we showed that uBonjour can be used for self-configuration and standardized discovery of sensor devices. With uBonjour, we enable a transparent service discovery over the Internet or local networks and offer a standardized integration into current infrastructure for the Internet of Things vision.

In future work, we want to perform simulations of uBonjour to identify bottlenecks, the bootstrapping scaling factor, and further implementation optimizations. We also plan to deploy more devices and extend the current state of our small testbed to facilitate larger practical tests.

References

1. Ashton, K.: That 'Internet of Things' Thing. Online RFID Journal (2009), http://www.rfidjournal.com/article/view/4986/ (accessed February 10, 2012)
2. Durvy, M., Abeille, J., Wetterwald, P., O'Flynn, C., Leverett, B., Gnoske, E., Vidales, M., Mulligan, G., Tsiftes, N., Finne, N., Dunkels, A.: Making Sensor Networks IPv6 Ready. In: Proceedings of the 6th ACM Conference on Networked Embedded Sensor Systems, SenSys 2008 - Poster session (2008)
3. Mattern, F., Floerkemeier, C.: From the Internet of Computers to the Internet of Things. In: Sachs, K., Petrov, I., Guerrero, P. (eds.) Buchmann Festschrift. LNCS, vol. 6462, pp. 242–259. Springer, Heidelberg (2010)
4. Shelby, Z., Hartke, K., Bormann, C., Frank, B.: Constrained Application Protocol (CoAP). Internet Draft, IETF (2011)
5. IBM: MQ Telemetry Transport (2012), http://mqtt.org/ (accessed February 20, 2012)
6. Dunkels, A., Gronvall, B., Voigt, T.: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In: Proceedings of the 29th Annual IEEE Conference on Local Computer Networks, LCN 2004, pp. 455–462. IEEE (2004)
7. Cheshire, S., Krochmal, M.: Multicast DNS. Internet Draft, IETF (2011)
8. Cheshire, S., Krochmal, M.: DNS-based Service Discovery. Internet Draft, IETF (2011)
9. The Avahi Team: More About Avahi - Details about mDNS, DS-DNS and Zeroconf (2011), http://avahi.org/wiki/AboutAvahi (accessed February 23, 2012)
10. Apple Inc.: mDNSResponder (2011), http://www.opensource.apple.com/tarballs/mDNSResponder/ (accessed February 20, 2012)

11. Cheng, J., Kunz, T.: A Survey on Smart Home Networking. Technical Report SCE-09-10, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada (September 2009)
12. Marvell Technology Group Ltd.: PlugComputer Community (2012), http://www.plugcomputer.org/ (accessed February 28, 2012)
13. Zender, R., Lucke, U., Tavangarian, D.: SOA Interoperability for Large-Scale Pervasive Environments. In: Proceedings of the 24th IEEE Conference on Advanced Information Networking and Applications (WAINA 2010), pp. 545–550. IEEE (2010)
14. Kaindl, G.: Bonjour/Zeroconf with Arduino (2012),http://gkaindl.com/software/arduino-ethernet/bonjour/ (accessed February 23, 2012)
15. Edwards, W.K.: Discovery Systems in Ubiquitous Computing. IEEE Pervasive Computing 5(2), 70–77 (2006)
16. Cheshire, S.: Setting up DNS to Allow Clients to Advertise their own Wide-Area Services) (2012), http://www.dns-sd.org/#WA (accessed February 20, 2012)
17. Pohlsen, S., Buschmann, C., Werner, C.: Integrating a Decentralized Web Service Discovery System into the Internet Infrastructure. In: Proceedings of the IEEE 6th European Conference on Web Services (ECOWS 2008), pp. 13–20. IEEE (2008)
18. Chen, D.K.: Systematic Review of Applying Service Oriented Architecture in Networking. In: Proceedings of the 6th Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010), pp. 167–170. IEEE (2010)
19. Hammoudeh, M., Mount, S., Aldabbas, O., Stanton, M.: Clinic: A Service Oriented Approach for Fault Tolerance in Wireless Sensor Networks. In: Proceedings of the 4th Conference on Sensor Technologies and Applications (SENSORCOMM 2010), pp. 625–631. IEEE (2010)
20. Bardin, J., Lalanda, P., Escoffier, C.: Towards an Automatic Integration of Heterogeneous Services and Devices. In: Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC 2010), pp. 171–178. IEEE (2010)
21. Teixeira, T., Hachem, S., Issarny, V., Georgantas, N.: Service Oriented Middleware for the Internet of Things: A Perspective. In: Abramowicz, W., Llorente, I., Surridge, M., Zisman, A., Vayssière, J. (eds.) ServiceWave 2011. LNCS, vol. 6994, pp. 220–229. Springer, Heidelberg (2011)
22. Kovatsch, M., Duquennoy, S., Dunkels, A.: A Low-Power CoAP for Contiki. In: Proceedings of the 8th Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2011), pp. 855–860. IEEE (2011)
23. Klauck, R., Gaebler, J., Kirsche, M., Schoepke, S.: Mobile XMPP and Cloud Service Collaboration: An Alliance for Flexible Disaster Management. In: Proceedings of the 7th Conference on Collaborative Computing: Networking, Applications & Worksharing (CollaborateCom 2011), pp. 201–210. IEEE (2011)
24. Silva, J., Camilo, T., Pinto, P., Ruivo, R., Rodrigues, A., Gaudêncio, F., Boavida, F.: Multicast and IP Multicast support in Wireless Sensor Networks. Journal of Networks (JNW) 3(3), 19–26 (2008)
25. Yazar, D., Dunkels, A.: Efficient Application Integration in IP-based Sensor Networks. In: Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2009), pp. 43–48. ACM (2009)
26. Sledz, D.: Liaison - because with Liaison, the client finds you (2003), http://www.acm.uiuc.edu/signet/liaison/ (accessed February 20, 2012)
27. Schönwälder, J., Tsou, T., Sarikaya, B.: Protocol Profiles for Constrained Devices. In: Proceedings of the IAB Workshop on Interconnecting Smart Objects with the Internet (February 2011)

Application-Level Operations Latency Control in Networked WSAN

Pedro Furtado and Jose Cecilio

Universidade de Coimbra, Coimbra 3030, PT
pnf@dei.uc.pt
http://eden.dei.uc.pt/~pnf

Abstract. The utilization of wireless sensor networks in industrial environments poses issues related to performance control. We consider a networked system composed of multiple wireless sensor networks (WSN) that are plugged into a cabled Networked Control System (NCS) made of middleware computers and control stations. The challenge that we faced was to provide predictable end-to-end latency expectations in those settings, as opposed to the problem of predicting latencies within a WSN only. We consider a deployment with multiple small-sized schedule-based WSN sub-networks and the NCS. The approach accounts for details such as de-synch issues that are to be expected in the heterogeneous context. Experimental results show actual latencies and confront them with predictions in a testbed deployment.

Keywords: distributed middleware, sensor networks, wireless sensor networks.

1 Introduction

Wireless sensor nodes are small devices with only limited amounts of memory, processing power and energy, but featuring a radio and wireless communication capabilities (with a short or medium range) that allows them to be autonomous computation-capable nodes in wireless sensor networks (WSN). The deployment ease and the possibility of saving enormous amounts of money by not requiring totally cabled systems have been the driver for applying them in industrial environments, at least as a complement to other technologies. WSN can be arranged as networks of nodes sensing and relaying data until it reaches a cabled receptor. Configuration and actuation commands can also travel the cabled and wireless sensor networks to a recipient node. In industrial premises, those wireless nodes can be arranged in multiple networks with star or tree topologies, each featuring a small number of nodes (e.g. ten to fifty). Those will be sub-networks of a larger cabled NCS. The industrial applications on that NCS monitor physical variables, but they are also capable of controlling valves and other devices all over the factory. Their functionality includes sensing, actuating (e.g. closing or opening a valve, or emergency shutdowns) and implementing closed-loop control over physical variables. Since WSN nodes are computation-capable, the system

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 330–343, 2012.

must also allow remote configuration of operation through simple configuration commands. All these control and data messages travel and are routed through gateways, middleware nodes and control stations in the NCS and through wireless sensor nodes in the WSN.

The networked control system built with wireless sensor sub-networks is depicted in Figure 1. In that figure WSN sub-networks with a few nodes are connected to a sink in a gateway, and gateways are connected to middleware machines and servers that aggregate data and connect multiple gateways. The control room, where alarms are shown and configuration thresholds can be defined, is connected to all servers. Closed loop control decisions may be triggered within a WSN (simple thresholds), or in servers outside the WSN. It is also possible to have a control loop that senses from one WSN and actuates through another one. We need to have some form of latency predictability, even though the system is not a single entity but rather a set of disparate sub-networks without global synch.

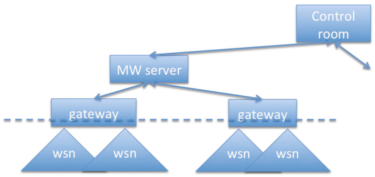


Fig. 1. Control System with wsn Components

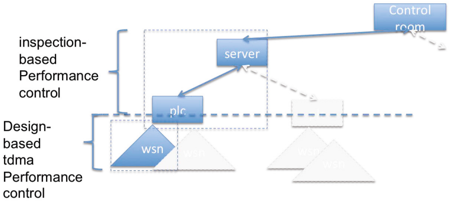

Control room
inspection-based Performance control
server
plc
Design-based tdma Performance control
wsn

Fig. 2. Performance Control Parts

The challenges for which we provide a solution are: how to provide latency expectations in the network of sub-networks, e.g. latency of a sensing or actuation command.

Many previous works have dealt with network-level performance control within a single WSN. They miss the set of independent sub-systems perspective, which are important issues in practical deployments. For instance, when a workstation commands an actuation to close a valve in a WSN, what is the latency expectation all the way down?

On the other hand, industry-strength approaches to performance control in the factory also do not deal with these issues, they typically concentrate on providing the appropriate infrastructure for cabled devices to work in a real-time manner. This is based on the use of technologies such as individual per sensor analog 4-20 mA communication links to and from controllers, or fieldbus (profibus) industrial networked systems for real-time distributed control. These allow multiple analog and digital points to be connected at the same time in a real-time manner. Profibus, for instance, defines master and slave devices and implements a token ring access control over the master devices (e.g. workstations and servers).

2 Related Work

Previous works on performance control involving WSNs are concerned with achieving control of performance within a single wireless sensor network. Their limitation when compared with our work is that they are concerned with the wsn, while we are concerned with performance monitoring and control over a networked control system that includes multiple WSN components. Nevertheless, we review works on that issue, as well as works on monitoring performance in wsns and on performance of networked control systems in general. Schedule-based approaches are used generically in performance controlled, real time wireless sensor networks. The RT-Link protocol [1] is an example of a performance-controlled approach. In it time-slot assignment is accomplished in a centralized way at the gateway node, based on the global topology in the form of neighbor lists provided by the WSN nodes. It supports different kinds of slot assignment depending on whether the objective function is to maximize throughput or to minimize end-to-end delay. Interference-free slot assignment is achieved by means of a 2-hop neighborhood heuristic, coupled with worst-case interference range assumptions. WirelessHART [2] is another performance controlled, reliable design to support industrial process and automation applications. In addition, WirelessHART uses at its core a synchronous MAC protocol called TSMP [3], which combines TDMA and Frequency Division Multiple Access (FDMA). A central entity called Network Manager is used to assign collision free transmission slots and to select redundant routing paths through a mesh network. Thus, the protocol guarantees an upper delay bound within the wireless system, while ensuring high transport reliability. GinMAC [5] is a TDMA protocol that incorporates topology control mechanisms to ensure timely data delivery and reliability control mechanisms to deal with inherently fluctuating wireless links. The authors show that under high traffic load, the protocol delivers 100 percent of data in time using a maximum node duty cycle as little as 2.48 percent. This proposed protocol is energy efficient solution for time-critical data delivery with neglected losses. PEDAMACS [6] is another TDMA scheme including topology control and routing mechanisms. The sink centrally calculates a transmission schedule for each node, taking interference patterns into account and, thus, an upper bound for the message transfer delay can be determined. PEDAMACS is restricted by the requirement of a high-power sink to reach all nodes in the field in a single hop. PEDAMACS is analyzed using simulations, but a real-world implementation and corresponding measurements are not reported. Finally, we also looked at work on networked control systems (NCS), since we are dealing with application level issues. The study of NCS is mostly concerned with either the perspective of the supervision processes involved being delay-tolerant, or with the performance or design of totally wired or wireless architectures for implementing NCS. We briefly review some of those works in the rest of this section. [7,8] has studied networked control systems over wireless networks. [7] shows that distributed random channel access schemes lead to significant performance degradation compared with TDMA and Polling. In [8] the authors propose a cross-layer framework for design of distributed control over wireless networks. The network design goal was to optimize the control performance, an implicit function of the network performance. They conclude that optimized control is achieved with a cross-layer solution. These works were focused on a totally wireless networked control system. In [10] the authors argue that communication network in the feedback control loop makes the analysis and design of an NCS complex. According to the authors, conventional control theories with many ideal assumptions, such as synchronized control and non-delayed sensing and actuation, must be reevaluated before they can be applied to NCS. The main issue from the perspective of design of the supervision processes is how to design delay-tolerant supervision for closed loop control. The work in [9] studies synchronous versus asynchronous actuation in NCS where delays are expected, concluding that asynchronous approaches can improve the performance of control loops in such contexts. Knowledge and correct estimation of delays is pointed out as an important factor to improve the approaches. The work in [11] discusses design issues for network architectures in a type of distributed control system where sensors, actuators, and controllers are interconnected by a common-bus network. The authors discuss the impact of network architecture on control performance and provide design considerations related to controlling performance as well as network quality of service. Design considerations include network parameters, control parameters and networked control system performance as the design guideline. The work is clearly focused on planning the NCS, not on monitoring performance on architecture with wireless sensor networks and the remaining cabled NCS.

3 Layout and Performance Control

The architecture is shown in Figure 1, with cabled network servers and WSN sub-networks connected to the cabled servers through gateways. The WSN sub-networks are tree-organized hierarchies with tens of nodes each, connecting to the physical systems for sensing and actuating all over the factory. One WSN tree typically spans a region in the factory premises. Our experimental testbed contains two such WSN sub-networks, totaling three dozens of sensor nodes. The cabled part of the system is independent of the WSN technologies involved, so that we can easily plug in any WSN sub-network through a gateway that must be provided. The gateway is a small software component in a gateway server that interfaces with the WSN sink node, using IP to connect to the cabled servers. The key to performance guarantees in these settings is to use schedule-based operation and a precise schedule in the WSNs. The control knob concerning the WSN networks is to dimension the schedules appropriately for meeting target performance objectives. The ethernet-based cabled network is assumed to have no performance limitations and a large bandwidth. A simple-enough middleware implements end-to-end communication, messages, remote configuration and operation of the nodes. End-to-end timing performance of the whole system is monitored and controlled by the middleware, through timing statistics and inspection during both deployment and runtime. This way it becomes possible to verify whether performance targets are met as required. Figure 2 shows a diagram of the system with performance control parts.

3.1 WSN Tdma Schedules

The WSN components use a TDMA schedule (our testbed applies GinMac [5]). The TDMA schedules of each WSN are cyclic round-robin operation that provides each WSN node with a slot to send its data to the parent node. By default, the whole tree is covered in a depth-first fashion, and parent nodes are provisioned with their own slot plus one slot per child to send their children data up. There are n branches in the tree (three in our testbed and examples: left, middle and right), feeding their data to the sink node one branch at a time starting by the left branch. There are also three levels in the example tree. Level 1 is composed of the three nodes that are closest to the sink node, then level 2 are the nodes that are children of level 1 and finally level 3 nodes are the leaf nodes. Nodes go to sleep when they are not in their schedule slot, which saves a significant amount of energy. They wake up in the previous slot and transmit only in their slot (all nodes also wake up simultaneously in a specific slot in the schedule to hear downstream message broadcasts). Figure 3 shows a diagram of the tree, together with slot numbers for one of the tree branches. In this tree there are a total of 33 slots upstream.

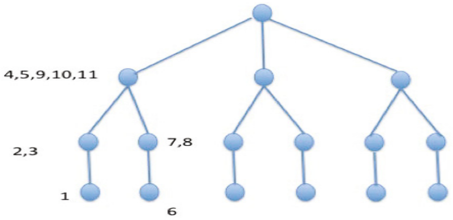


Fig. 3. Tree with tdma slots for one branch

Timing constraints dictate the epoch size (and maximum number of nodes) of the component WSNs. If there is a constraint that monitoring latency should be less than 1 second, the epoch size can be set to a value lower than that latency, so that the maximum operation latency it at most 1 second - the time to go all around the tree, plus the time to transmit and process in the cabled part of the system. After the 33 slots necessary to transmit data from all nodes are inserted within a 950 msecs epoch size (each slot taking 10 msecs), the rest of the epoch is spent sleeping (which conserves battery). To that schedule we must also add downstream slots for sending both configuration commands and actuations to nodes in the tree (plus slots for network management). Three downstream broadcast slots with all nodes awaken are sufficient to reach all nodes in three levels of the tree in the example, and this is 3x2=6 if we include retry slots. We consider multiples of these 6 slots depending on how many actuation, configuration commands and timing synch transmissions are expected per each epoch.

For instance, 2 x 6 slots would be able to handle one actuation command per second (e.g. closed loop control with a period of 1 second) and one configuration command. Configuration commands identify one or more recipient nodes within a WSN with small numeric single byte ids (0 to 255).

3.2 Performance Control of the WSN Components

The epoch length (eL) provides an upper bound for the time taken by a sensed value to arrive at the sink node, from an operations perspective. But lower bounds can also be deduced. For instance, if the sensing instant is synchronized with the slot time (e.g. the sensing is triggered 50 msecs before the node gets its slot), then the upper bound on latency can be that triggering time plus the time to reach the sink node. Since the WSN tree is divided into branches that forward all their sensed values into the sink before the next branch is processed, a simple upper bound for sensing would be the triggering time plus the sum of the slot times for all the nodes of the branch, or it can be defined for each specific node as the triggering time plus the slots necessary for the value from that specific node to arrive at the sink. For instance, Figure 4 shows the order of sending data for branch b1.

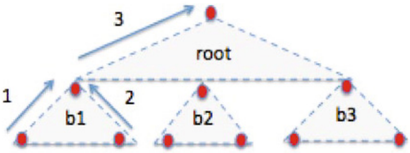


Fig. 4. Slot order example in branch b1

In Figure 3 we have placed slot numbers into the picture. The leaf node with slot 1 sends its data first to its parent, which then uses slot 2 to send the child data to the parent and then slot 3 to send its own data. The parent himself then forwards all children data from that sub-branch to the sink node, starting by the data from the first node. Next, the leaf node of the other sub-branch uses slot 6 to send its data up, and so on. In this case leaf node data will take 5 slots to arrive at the sink, plus the triggering time. Sensing can also be synchronized with the beginning of an epoch size all nodes do sensing at the start of an epoch. In this case the timing for each node to start sending its sensed data is the number of slots counted from the beginning of the epoch (the sensing instant). To this number we must add the number of slots for the data to reach the sink node.

3.3 Performance Prediction of the End-to-End System

With a few assumptions, it is possible to determine expected bounds and expected average latencies that the system will exhibit. The latency perspectives concern: sensing, sending configuration commands to sensor nodes, actuating some physical device and closed loop. We discuss those in the next sub-sections. We will use as an example our testbed settings of the tree shown in Figure 5, with the corresponding schedule of Figure 6.

Sensing. The end-to-end (application-level) latency of sensing (ls) is the time interval from the instant the node senses a quantity to the moment the sensed value is delivered within a server that operates on the sensed value (the definition could also include for instance the time to render the alarm sign in the display). Latency has the following components shown in equation 1: latency from the sensing instant to the transmission slot, plus latency to traverse the wsn, plus latency introduced by the gateway, plus latency for the cabled part of the network, and finally latency within the middleware.

$$ls = l_{slot} + l_{wsn} + l_{gateway} + (l_{cabled} + l_{middleware}) \tag{1}$$

The gateway latency is expected to be small, we assume an upper bound of 5 msecs. The sum (lcabled + lmiddleware) or NCS latency (latency of the networked control system) includes cabled network and middleware overheads. It depends on how the gateway and middleware communication and processing logic handles the data. This can be subject to unexpected delays due to disturbances of the non-real-time environment. For instance, a garbage collector may delay execution in certain instants; other operating system processes may also delay processing. Nevertheless, this is a cabled component, sockets are open and ready and the cabled part is designed to avoid congestion. A reasonable upper bound can be used for this part, and inspection (simple testing) can be used to set this upper bound. In our experimental prototype this latency was consistently less than 100 msecs, and the average was about 20 msecs. The sensing synch approach also determines other possible variations. It is possible to start the sensing procedure some fixed amount of time before the sending slot arrives, resulting in a bound for lslot (e.g. 40 msecs). If, on the other hand, sensing starts at the beginning of the epoch schedule (every sensor senses simultaneously), then the latency is:

$$l_{slot} = l_{branch} + l_{nodeInBranch} == \sum_{untilbranch} slot + \sum_{inbranchuntilnodeslot} slot \tag{2}$$

Equation (2) means that lslot is made of the latency from the start of the epoch to the slot starting the wsn tree branch, plus the latency from the start of the branch to the slot of the node. This latency can also be looked-up from the slot number in the epoch schedule. The quantity lwsn is the latency for data to arrive at the sink. Considering a leaf level sensor node in the example, this latency includes the sending slot, slots for each node in its right of the branch to which the node belongs, then slots for the parent to send the data up from each node in its left and of the node. This means that every leaf node will take 4 slots to reach the sink (4 x 10 msecs). Since we consider a retry slot for each data slot, we will account for 4x20 = 80 msecs, less one slot if the data arrives correctly. The lgateway latency is given as a bound (in our testbed it is 5 msecs). The NCS latency is also given as a bound (100 msecs in our testbed). This way, a time bound for a leaf node with sensing synchronized with the slot is expected to be 40 + 70 + 5 + 100 = 215 msecs. When sensing of all nodes is synchronized with the start of each epoch, then lslot varies per node according to formula (2). The corresponding lslot latencies can be looked up from the epoch schedule. The remaining end-to-end application level latencies remain the same as discussed above.

Configuration and Actuation Commands. Configuration commands go downstream from the commanding workstation into the wsn, and then until they reach the node. There is a particularly significant latency involved in this path, which results from the command submission not being synchronized with the downstream slots of the wsn: when a command arrives at the sink node of the wsn it must wait for the downstream transmission slots. The command latency is:

$$ls = (l_{cabled} + l_{middleware}) + l_{gateway} + l_{slot} + l_{wsn} \qquad (3)$$

We assume similar latencies as before for the quantities: (lcabled + lmiddleware, lgateway). However, the lslot latency is now a probabilistic value, since the command can be sent at any moment and there is no synchronization with the wsn epoch schedule. There is an epoch (950 msecs in our testbed), and within that epoch there is a downstream slot that must be waited for. If there was a single downstream slot in the whole epoch, the command would have to wait a full epoch in the worst case (e.g. 950 msecs) and half of that (475 msecs) in average. The quantity lwsn depends on the level of the node receiving the command. It is 30 msecs for a leaf node in a 3-level tree and no retries (3 consecutive downstream slots for sending the data down three levels). An expected end-to-end average latency using (3) would therefore be (475 + 100 + 5 + 30 =610 msecs) (this is not exactly an average, since we are considering an upper bound on the NCS latency) and maximum latency of (950 + 100 + 5 + 30 =1085 msecs) is expected. Adding more downstream slots in the schedule can decrease these latencies. For instance, if there are two equally-spaced sink-to-children downstream slots per epoch, then the lslot latency is halved. Actuation refers to sending an actuation command from a middleware machine to a wsn node that controls some actuation device. This latency is assumed the same as the one for a configuration command, which means that we should expect an average of 610 msecs and a maximum of 1085 msecs.

Closed Loops. Closed loops are data paths, which involve sensing, decision logic and actuation based on the decision. The simplest decision logic is a threshold, more complex ones can be PID controllers or other computational mechanisms. The closed loop may be sending an actuation command every time a sensor value arrives or with a period decided in the decision logic. When there is fixed actuation period, the latency predictions provide a lower bound on the possible period. Depending on the position of the sensed and the actuated physical equipment, the lower bound requirements and the complexity of the decision logic, the whole closed loop can be done in a sensor/actuator node itself, through the sink node of a tree or through the middleware outside of the wsn trees. In the first case a single mote senses, applies decision logic and actuates. In this case the time taken is the sum of the time to sense, to compute and to actuate through a DAC, typically this would be less than 50 msecs, but depends on the sensor and actuator devices. Closed loops can also go through the sink node. In this case one or more sensor nodes send data to the sink, which evaluates a threshold and issues an actuation value to some other actuation node. The corresponding latencies are represented in equation 4.

$$ls = l_{sense} + l_{slotU} + l_{wsnU} + l_{compute} + l_{slotD} + l_{wsnD} + l_{act} \tag{4}$$

As in (1), if sensing is synchronized with the slot, (lsense + lslotU) will be the defined sensing interval (e.g. 40 msecs) and lwsnU will be 80 msecs for a leaf node of the example. The latency lcompute is typically insignificant. In order to have a small value for lslotD, a downstream slot should be placed only a few slots after the slot that delivers the sensed data to the sink node. This way the sink is able to send the actuation command soon after decision. In our experimental testbed this interval was (lslotD=180 msecs). The downstream latency (lwsnD) for a leaf node in a 3 level tree would be (3x(10+10) = 60 msecs). This assumes a retry slot for each transmission, which is important since actuation values should be delivered. The full closed loop path will then take 40 + 70 + 180 + 60 = 350 msecs. Since the largest latency in this computation is lslotD, simply placing the downstream actuation slots nearer to the upstream sensing slots can decrease this value. However, the sink node needs to be given enough time to process the decision condition, besides handling the sensor signals that may be constantly arriving from the other nodes. The end-to-end latency for closed loops through a middleware machine (in the cabled part of the system) is determined by summing the latencies from equation (1) for the upstream sensing part, plus equation (3) for the downstream actuation part. In the example, (3) is in average 610 msecs and at most 1085 msecs, and (1) would be 215 if sensing is synchronized with sending. Summing both, the worst case closed loop through the middleware with sensor and actuator in leaf nodes would be 1085 + 215 = 1300 msecs in the example, and the average value would be 825 msecs. If we want a lower worst case, we need to add downstream slots to decrease the (3) component.

4 Experimental Results

These experiments were done in the context of an experimental testbed for EU project Ginseng, the testbed being located in a factory floor (a refinery). We will present observed latency results concerning that setup and will show that expected latencies computed as in the previous section agree with the observed results. The testbed prototype consisted of two WSN networks both built with

TelosB nodes and running operation code on top of the GinMac tdma protocol. The gateways connecting the WSNs to the cabled network were connected through an Ethernet to three middleware machines containing backend middleware, which included our system configuration tool (which sends commands and actuations to nodes) and our performance monitoring tool. The WSN networks were running Contiki OS and sensing measures such as pressure in the pipes, fluid levels, temperature or gas levels. The trees were organized hierarchically, one had 13 nodes in a 1-1-3 structure (5), the other one had 16 nodes organized in a 1-1-2-1 tree (Figure 3). A static WSN topology was used with 950 msecs of epoch length and slots of 10 msecs per node, with an extra retry slot for each transmission, except for downstream config messages (these are retried at application level if the end-to-end ack message is not received within a specific timeout).

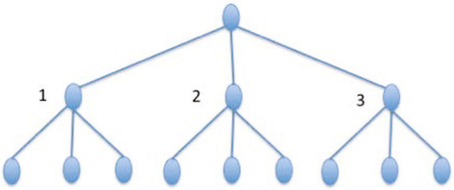


Fig. 5. 1-1-3 Tree Configuration

The slot assignments used in the refinery testbed are shown in Figure 6. Node 0 is the sink node, then the three tree branches are headed by nodes 1, 2 and 3 respectively. Looking from left to right in the schedule of Figure 6, there are: Two processing slots for the sink node; Two control sensing slots for the sensor used in closed loop control (node 1) to send a sample to the sink node; Upstream slots for all nodes from branch 1 to send their data to their head; Upstream slots from branch 1 head to send the data of its children and then its data to the sink node; Timing synchronization slots; Downstream slots for configuration commands to reach every node (no retry, broadcast); Downstream slots for actuation values (in this case the actuation is done in node 1, so the actuation values go only one link down from sink to node 1; Upstream slots for the remaining two tree branches; The remaining epoch time is spent sleeping.

4.1 Monitoring Latency - Results and Analysis

Figure 7 shows the observed monitoring latencies (average and standard deviation) for the 1-1-3 tree, after running for 14 hours with every node sensing every second. Figure 8 shows the latencies within the WSN.

In order to verify that the latencies agree with the values we would expect using the calculations of the previous section, we have marked in the schedule of Figure 6 the slot numbers and the instants when each node sends its signal up to the parent node. Since each slot takes 10 msecs, each node sends its data

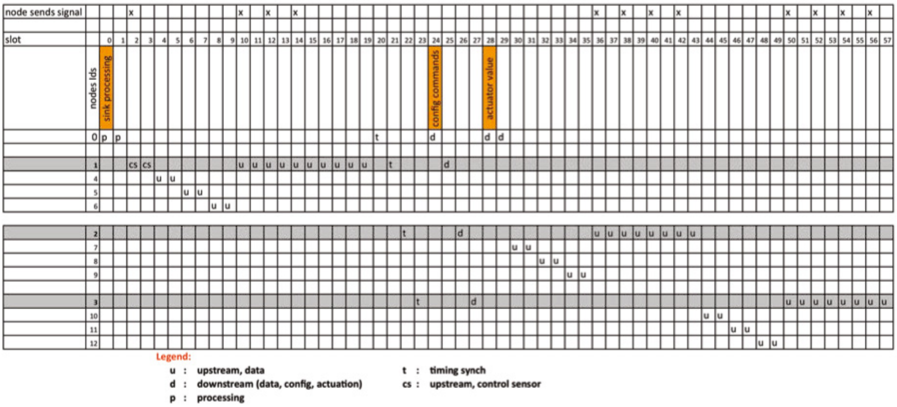


Fig. 6. 1-1-3 Tree Slots Assignment

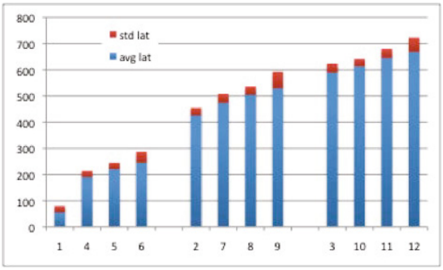


Fig. 7. Per-node end-to-end operation latencies (average and stdev)

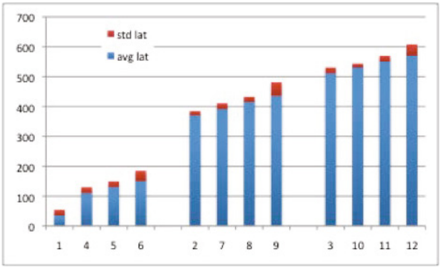


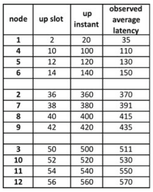
Fig. 8. Per-node average and stdev latencies - WSN

up at an instant t = 10 x slot. The observed latencies of Figure 8 agree with the schedules of Figure 6, and Figure 9 shows this. It lists the slot numbers, up instant (instant when the node starts sending up its data) and observed latency (average). The Figure shows that the average observed latencies within the WSN are as expected by looking at the schedule, with only 10 to 15 msecs of maximum difference to expected values.

This accounts for the part (lslot + lwsn) of the latencies formula. While Figure 8 shows the latencies of the wsn part, Figure 7 shows the total end-to-end application-level operation latencies. Comparing Figure 7 with Figure 8 we can see that the remaining part (lgateway+ lcabled + lmiddleware) in these experiments was below 100 msecs for al nodes, as expected by the NCS latencies.

4.2 Config Commands - Results and Analysis

Figure 10 shows the average and maximum latencies of a configuration command sent to a leaf (level 3) node. As predicted in the previous section, most of the latency is spent waiting for a downstream slot in the schedule. The average



node	up slot	up instant	observed average latency
1	2	20	35
4	10	100	110
5	12	120	130
6	14	140	150
2	36	360	370
7	38	380	391
8	40	400	415
9	42	420	435
3	50	500	511
10	52	520	530
11	54	540	550
12	56	560	570

Fig. 9. Schedule slot instant and observed latencies

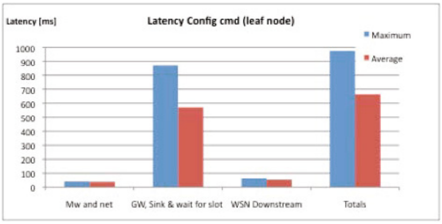


Fig. 10. Config Command latency

and maximum gateway, Sink and wait for slot time was 580 and 880 msecs respectively. This agrees with the predicted values of half the epoch size (450 msecs) for the average slot waiting time and the maximum interval between downstream slots (950-80 msecs, since commands can be sent in the timing synchronization or actuation slots) for the maximum slot waiting time. To those latencies we must add middleware and network latencies (100 msecs maximum), and also 50 to 60 msecs for the config command to go down in the wsn (3 broadcast slots with retries), resulting in a prediction close to the actual observed values.

4.3 Closed Loop - Results and Analysis

We have setup two simple closed loop actuation scenarios and collected results during 14 hours for each. Node 1 (level 1) collects sensor data and sends the data to the sink node. In scenario 1 the sink node evaluates a threshold and sends an actuation command down to a level 1 node, which applies the command. In scenario 2 the sink forwards the data to the middleware through the cabled network for evaluation of a threshold. The middleware machine sends then an actuation to the sink node in the other wsn network, which forwards it to a level 1 actuator. We now analyze whether the results are according to predicted.

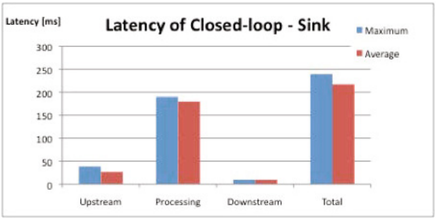


Fig. 11. Closed loop at Sink Node avg + maximum

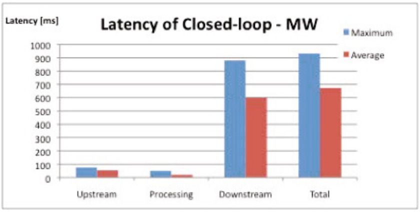


Fig. 12. Closed loop in Middleware avg + stdev

For the closed loop decision at the sink node, the results were lwsnD = 27 (avg) to 39 (max). This accounts for the time needed from the start of the schedule in Figure 6 to the receiving of the sensed value cs by the sink node. The Processing time in Figure 11 includes the time waiting for the downstream slot back from the sink to the level 1 node doing the actuation. The value (around 180 to 190 msecs) is less than the time required for waiting for a downstream actuation slot. However, this is explained by the fact that the timing synch downstream slots were used to send the actuation value sooner. Finally, the downstream latency is, as expected for a level 1 node, around 10 msecs. Results concerning middleware-based closed loop are in Figure 12 . In this case the Upstream latency (54 average, 76 maximum) includes the cabled part of the system (our computed upper bound would be 130 msecs. This is 100 msecs bound for the gateway, cabled network and middleware, plus 30 msecs at most for the wsn). The downstream part took 599 in average and 880 worst case. The predicted upper bound would be 100 msecs for the cabled part, plus about 475 msecs for waiting for a slot (half the epoch size), plus 10 msecs (slot time for sending the actuation value to the level 1 node). This gives an average 585 msecs, very close to the actual 599 msecs. Finally, our predicted maximum latency for the downstream part would be obtained by replacing the 475 msecs by the maximum distance between downstream slots in the schedule, which is 870, assuming the timing control slots can be used by the closed loop actuation values. The sum is then 875 + 100 + 10 = 985 msecs. Compared to the actual value of 880 msecs, this upper bound prediction overshot by 100 msecs.

The experiments compared computational predictions with actual latency values for sensing, configuration commands and closed loop actuation between the two networks, concluding that the prediction approach is sufficiently accurate.

5 Conclusions and Future Work

In this paper we have investigated a simple model for predicting application-level end-to-end operation bounds and average latencies for sensing, commanding, actuating and closed loop control in a networked control system architecture with cabled ethernet and wireless sensor network components. The objective is to derive predictions for application-level operations, as opposed to latencies of only packets traveling between two points in a single network, and to consider the heterogeneous system made of cabled and schedule-based tree-organized wsn sub-networks. We have shown through a simple testbed that the predictions are acceptable and account well for the scenarios. We have focused on a particular organization of the networked control system with wsns. Future work includes adapting the approach to other organizations, comprehensive planning of the approaches and automated constraints-based schedule planning and determination.

Acknowledgments. We would like to thank the major contributors to this paper. The work and people involved in this paper were financed by University of Coimbra, the National Science Foundation (FCT) and the European Commission under the contract FP7-ICT-224282 (GINSENG).

References

1. Rowe, A., et al.: RT-Link: A Global Time-Synchronized Link Protocol for Sensor Networks. Elsevier Ad hoc Networks, Special Issue on Energy Efficient Design in Wireless Ad Hoc and Sensor Networks (2007)
2. HART Communication Foundation, Wireless HART Data Sheet (April 2010), http://www.hartcomm.org/
3. Pister, K.S.J., Doherty, L.: TSMP: time synchronized mesh protocol. In: Proc. IASTED Symp. Parallel and Distributed Computing and Systems, Orlando, FL, USA (2008)
4. Shashi Prabh, K.: Real-Time Wireless Sensor Networks, Ph.D. Thesis, Department of Computer Science, University of Virginia, Charlottesville, VA, USA (2007)
5. Suriyachai, P., Brown, J., Roedig, U.: Time-Critical Data Delivery in Wireless Sensor Networks. In: Rajaraman, R., Moscibroda, T., Dunkels, A., Scaglione, A. (eds.) DCOSS 2010. LNCS, vol. 6131, pp. 216–229. Springer, Heidelberg (2010)
6. Ergen, S.C., Varaiya, P.: PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks. IEEE Trans. Mobile Comput. (2006)
7. Liu, X., Goldsmith, A.: Wireless medium access control in networked control systems. In: Proc. IEEE American Control Conference (2004)
8. Liu, X., Goldsmith, A.: Wireless network design for distributed control. In: 43rd IEEE Conference on Decision and Control (2004)
9. Yepez, J., Marti, P., Fuertes, J.M.: Control loop performance analysis over networked control systems. In: IEEE 28th Annual Conference of the Industrial Electronics Society, IECON 2002. Dept. of Autom. Control and Comput. Eng.,Tech. Univ. of Catalonia, Barcelona, Spain (2002)
10. Zhang, W., Branicky, M.S., Phillips, S.M.: Stability of Networked Control Systems. IEEE Control Systems Magazine (2001)
11. Lian, F.-L., Moyne, J., Tilbury, D.: Network Design Consideration for Distributed Control Systems. IEEE Transactions on Control Systems Technology 10(2), 297 (2002)

Intelligent Multicast Tree Construction Protocol with Optimal Bandwidth Allocation for WSNs

Nedal Ababneh, Antonio M. Ortiz, Nicholas Timmons, and Jim Morrison

WiSAR lab, School of Engineering
Letterkenny Institute of Technology
Port Road, Letterkenny, Ireland
{nedal,antonio,nick,jim}@wisar.org

Abstract. This paper addresses the problem of multisession multicast tree construction with bandwidth and rate allocation in wireless sensor networks. Previous work has shown that when the goal is to find multicast routing tree, the problem becomes NP-complete. In this work, we present a heuristic Multisession Multicast Routing and Bandwidth Allocation protocol, termed MMBA, that makes use of fuzzy logic to evaluate nodes' and network conditions during the multicast tree construction process. Rate assignment is optimized in order to be able to accept as many data streams (i.e., sessions) at the highest possible data rate in the sensor network as possible, allowing source nodes to transmit at maximum available rate, while maximizing the overall network throughput and utility. We conduct extensive evaluations to study the performance of the proposed protocol compared to existing approaches such as shortest path, Steiner and minimum transmission tree. Simulation results show that our protocol effectively improves the network throughput and utilization, while conserving per node energy consumption.

Keywords: Wireless Sensor Networks, Multicast Tree Construction, Multisession, Bandwidth Allocation, Data Streaming, Fuzzy Logic.

1 Introduction

A wireless sensor network (WSN) is a network consisting of several spatially distributed sensing devices equipped with wireless communication capabilities [1]. The main objectives of these networks are the continuous and collaborative monitoring of a given object, volume or surface. Application domains include monitoring, intrusion detection, traffic control or manufacture tracking among others. Nodes composing WSNs are resource constrained in terms of processing power, available memory, energy and bandwidth. In a WSN, each node can communicate with the devices that are within its radio range (neighbors), and non-neighbor nodes can communicate with each other by using multi-hop paths, in which intermediate nodes act as relays. In this case, the communication involves the use of a routing protocol to decide which nodes will perform data

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 344–357, 2012.

forwarding. When communication occurs between a source and one single destination, it is called unicast. When several destinations are considered, we talk about multicast, and broadcast, if all nodes are considered destinations.

Multicast is a communication paradigm that performs one-to-many or many-to-many, based on defined groups and constituted by members whose interest is to receive or share identical information for a specific application [2]. Multicast allows the sender to transmit a message (destined for multiple receivers) only once, instead of sending it to each end-point separately. In terms of bandwidth efficiency, using a multicast session usually outperforms multiple unicast sessions [3]. Multicast can be used in WSNs in several ways such as a source node is able to send data to multiple sinks, or sinks can distribute control messages to a set of sensor nodes. The design of multicast protocols for WSNs is a challenging task due to the resource constrained nature of WSN nodes. The most restricted resource in these networks is the energy, since nodes are usually battery-powered and battery replacement is not considered a practical solution in most cases. Thus, communication protocols, must be as much energy-efficient as possible, but also considering other metrics such as bandwidth among others.

In this paper, we study the guaranteed multicast data streaming problem with variable data rates in WSNs, where source nodes can send data to several destinations at different rates (i.e., for each source node), exploiting the available bandwidth. For that, we use fuzzy logic (FL) to evaluate nodes' conditions during multisession multicast tree construction process, while providing optimal bandwidth and data rate assignment to nodes in the multicast tree(s). The Multisession Multicast Routing and Bandwidth Allocation protocol, termed MMBA, proposed herein is an adaptive cross-layer approach that performs network discovery, tree construction and bandwidth allocation for multisession multicast data streams in WSNs. MMBA uses fuzzy logic to evaluate network nodes by considering hop count, residual energy and remaining link capacity, and selects those nodes with better evaluation values to be part of the multicast tree, thus being able to accept more data streams at the highest possible data rate. MMBA provides optimal rate assignment in order to be able to accept as many data streams as possible at the maximum possible data rate in the network, while maximizing the overall network utility. In addition, optimal rate assignment guarantees multi-hop data streaming since the severe energy and bandwidth constraints in WSNs do not allow source nodes to transmit at the maximum data rate in all situations.

The remainder of this work is organized as follows: Section 2 briefly provides the related work in this research area. Section 3 introduces the network model. In Sect. 4, the detailed design of MMBA is presented, which is then evaluated in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

Several approaches for multicasting in ad hoc and sensor networks have been proposed in the literature with diverse objectives and network conditions. The use of multicast for the resource constrained wireless sensor networks favors an appreciable energy saving, thus generally improving the overall network lifetime. In turn, rate assignment makes easier to take advantage of the full available bandwidth, allocating node transmissions in an effective manner. Together, multicasting and rate assignment enable an efficient use of the WSN resources. Due to its broadcast nature, the wireless medium is well suited for performing multicast communications, when one (or several) source aims to transmit data to several destinations. A review of multicasting protocols for ad hoc networks is given in [4], where the authors classify multicast approaches into minimum energy and maximum lifetime algorithms, detailing solutions for directional and omni-directional antennas.

In [5], authors studied the minimum power broadcast/multicast routing problems and proposed several greedy heuristics under a scenario in which each node can continuously adjust its transmission power. The work in [6] provided several approximation algorithms based on the minimum Steiner tree algorithm to the heuristics proposed in [5], after demonstrating that the proposed heuristics have linear approximation ratios. Ruiz *et al.* in [7] demonstrated that for wireless multi-hop networks, the Steiner tree is no longer offering the lowest bandwidth consumption, and reformulates the problem of optimal multicast tree in terms of minimizing the number of transmissions. In [8], authors proposed an energy-aware multicast scheme for ad hoc and sensor networks, that is based on minimum spanning tree and selects at each hop, the neighbor providing the smallest ratio between the cost needed to transmit the message to that neighbor and the progress toward the destination when using it. The experiments only show results for energy consumption, not analyzing other parameters such as throughput or end-to-end delay, etc. Cheng *et al.* [9] proposed a distributed heuristic for minimum transmission multicast routing that is based on selecting forwarding routes which can connect more multicast receivers. However, multisession is not considered, and only one multicast flow is active at a time. In addition, in data streaming applications minimum transmission multicast tree does not guarantee high network throughput.

All the aforementioned proposals have been designed for fixed-rate scenarios with all nodes having the same transmission rate. When the bandwidth drop, the throughput per node will be decreased equally. In our proposal, data rate will be assigned to the nodes (i. e., sources) according to their residual energy. Data rate assignment has been considered in a number of papers. In [10], authors proposed Rate Adaptive Multicast (RAM) protocol that adapts the communication rate to the link quality in order to reduce the overall network traffic. To achieve this objective, among several paths available between a source and a receiver, RAM selects the path with the lowest total transmission time. The work in [11] considered the problem of constructing a rate-based multicast tree by flooding *explore* messages from the source to the destinations. When reached, the destinations send back *Ack* messages specifying their required rate, and the multicast tree is constructed on the way back to the source of these *Ack* messages. However, this protocol is only able to work with one source, and the evaluation is just based on the number of messages necessary to create the communication tree. Multicast tree construction is also discussed in [12], where the authors aim to maximize the overall receiver utility function, but yet, the experimental evaluation does not show comparison with other proposals.

3 The Network Model

We consider a static multi-hop wireless sensor network modeled by an undirected graph $G = (V, E)$, called a connectivity graph, where V is the set of vertices representing the nodes in the network, and is composed of a group of sources denoted as V_S, a group of forwarders (i.e., relay nodes) denoted as V_F and a group of receivers denoted as V_R. E is the set of edges (i.e., communication links) that represents the communication network topology, edge$(v_i, v_j) \in E$ iff v_i, v_j are within each other's communication range. Hence, nodes form a multi-hop network among themselves to relay traffic to the receiver(s). Also, each edge (v_i, v_j) has a physical capacity L_{ij}, which represents the maximum amount of traffic that could pass through this particular link, and each node $v \in V$ represents a node in the network, and all nodes in the network are assumed to work on the same fixed transmission power with circular transmission range R_T. At any given time, a node may either transmit or listen to a single wireless channel, with channel capacity C.

Suppose $(v_s, GroupID)$ is a given multicast request as in [9], where v_s is the source and $GroupID$ specifies a set of multicast receivers V_R (i.e., all multicast receivers have the same $GroupID$), Given a graph $G = (V, E)$, a source node v_s and a set of receivers V_R, the multicast routing problem, can be defined as follows: finding a multicast tree T in G which connects the source v_s to every multicast receiver $v_i \in V_R$. Such a tree includes a set of forwarding nodes $V_F \subset V$ so that for each $v_i \in V_S$, $v_i \cup V_R \cup V_F$ are connected. In a multicast tree T, all the leaf nodes are destinations that only receive multicast packets; only the non-leaf nodes take the forwarding task. It is worth to mention that in this work a receiver node can be used to relay other nodes' traffic, while source nodes cannot, this helps the rate assignment model to assign the highest possible data rate for each source. Since optimal multicast tree construction is proved to be NP-complete [13] and, thus, requires heuristic solutions, in the next section, we present our heuristic to tackle the problem.

4 Protocol Description

In this section, we describe the design and implementation issues of the proposed protocol in depth.

4.1 Multicast Tree Construction

To achieve high network performance in WSNs, the route construction within the network is a crucial task. In this section, we present MMBA protocol that accounts for several source nodes (multisession) providing data streaming in WSNs. MMBA uses fuzzy logic in order to combine hop count, residual energy and remaining capacity metrics as illustrated below. The output provided by the fuzzy logic is used as a metric to evaluate relaying nodes in the multicast tree. This protocol assigns the best capacity on the set of possible routes. An illustration of the tree construction process is given in Alg. 1. The set of candidate parent nodes of a network node i is denoted by CN_i. The node i chooses the parent node, denoted by $prnt(i)$, that has a maximum value in the fuzzy logic evaluation, denoted by $FL(j)$ in Alg. 1. It first scans advertisement messages sent by neighbor nodes for possible set of candidate parent nodes, CN_i, and for each node j among the set CN_i, it calculates the fuzzy logic evaluation value $FL(j)$. Then, with the obtained values for all nodes in CN_i, it chooses among candidate parent nodes the one that maximizes the fuzzy logic evaluation value. To join, node i sends a JOIN message to j. Upon receiving the JOIN message, j adds i to its children list denoted by $Child(j)$, and sends an ACCEPT message to i.

Algorithm 1. Multicast tree construction

```
Input: node $i$ and $CN_i$
procedure ParentSelection($prnt(i)$)
    $\beta \leftarrow 0$
    for all $j \in CN_i$ do
        $\alpha_j \leftarrow FL(j)$
        if $\alpha_j > \beta$ then
            $\beta \leftarrow \alpha_j$
            $J \leftarrow j$
        end if
    end for
    $prnt(i) \leftarrow J$
    send($J$, JOIN-MSG)
return $prnt(i)$
end procedure
```

In this work, fuzzy logic is adopted to evaluate relay nodes, and the evaluation value is used as a metric in order to select those nodes with better overall conditions to be part of the multicast trees. Fuzzy logic is a decision system approach that works similarly to human control logic. It provides an output value that represents the combined evaluation of the input parameters, based on the experience of the system designer. A Fuzzy Logic System (FLS) is a nonlinear mapping of an input data vector into a scalar output [14]. A typical FLS, widely used in fuzzy logic controllers is composed of *fuzzifier*, *fuzzy rules*, *inference engine* and *defuzzifier*.

Fuzzy logic uses human language to define inputs, outputs and their relationships. The fuzzy rule set relates the state of the input variables to an state of the output variable. Calculations are performed by an inference engine. The operation of a FLS can be summarized as follows: the fuzzifier takes crisp input data and converts them into fuzzy values. The inference engine combines these fuzzy values by using the fuzzy rule set and provides a fuzzy output. Finally, the defuzzifier converts the fuzzy output into a numerical value that can be used by external systems. The execution of a FLS requires less computational power than conventional mathematical operations such as arithmetic operations [15].

The parameters used as input in the FLS included in MMBA are as follows:

1. Hop count: this parameter represents the number of retransmissions necessary to reach the destination node. It is closely related to energy consumption and end-to-end delay. Network resources are also affected by the number of hops since longer paths require more retransmissions.
2. Residual energy: the consideration of the remaining energy is very significant in resource constrained WSNs to avoid hot spots and to postpone draining nodes available energy as long as possible.
3. Available link capacity: this will allow MMBA to assign higher data rates for source nodes in order to maximize the use of the available bandwidth.

These parameters have been selected since they are the most representative for the problem tackled herein. Other parameters such as signal strength, number of neighbors (node density), node priority or node capabilities, have not been considered since they are not important for our application, but could be easily included in the system if required. Each one of the input parameters are characterized into several fuzzy sets. The fuzzy sets for the considered parameters are defined as follows: 1) $HopCount \subset \{close, medium, far\}$, 2) $ResidualEnergy \subset \{low, medium, high\}$ and 3) $LinkCapacity \subset \{poor, acceptable, available\}$. These input linguistic terms constitute the antecedent of the fuzzy rules, and the output (decision) variable is used to decide whether to include the node in the multicast tree. The fuzzy sets associated to the output variable are as follows: $Decision \subset \{bad, average, good\}$. An illustrative example of the membership functions for the input and output variables in our FLS is given in Fig. 1. For example, considering hop count variable, *label1* corresponds to *close*, *label2* to *medium* and *label3* to *far*, and the values $\{X_0, ..., X_4\}$ have been adjusted according to each input variable.

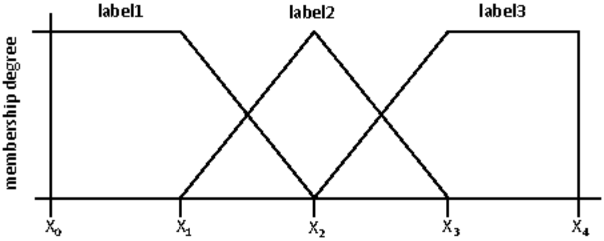


Fig. 1. Membership function example

The geometric pattern of triangles is commonly used to determine the appropriate membership functions and thanks to its simplicity, it requires low processing power [14]. The fuzzy rule base defines the relationship between input and output variables. Table 1 depicts the fuzzy rule set, where rules can be interpreted as follows: IF *HopCount* is *close* AND *ResidualEnergy* is *medium* AND *LinkCapacity* is *acceptable* THEN *Decision* is *average.*

Table 1. Fuzzy rule set

Hop Count	**Residual Energy**			**Link Capacity**
	low	**medium**	**high**	
close	bad	bad	average	**poor**
close	bad	average	average	**acceptable**
close	average	good	good	**available**
medium	bad	bad	bad	**poor**
medium	bad	average	average	**acceptable**
medium	average	average	good	**available**
far	bad	bad	bad	**poor**
far	bad	bad	average	**acceptable**
far	average	average	average	**available**

4.2 Rate and Bandwidth Allocation

We solve the rate and bandwidth allocation problem based on the tree constructed above in Sect. 4.1. The performance of the proposed model benefits from a well-structured multicast routing tree. For simplicity, we consider a single-radio, multi-channel WSN, where potentially interfering wireless links should operate on different channels, enabling multiple parallel transmissions. The problem therefore consists of increasing the number of accepted source nodes (i.e., streams/sessions) in the multicast routing tree at highest possible data rate while balancing out energy consumption to extend network operational lifetime.

The Problem Definition. The resulting multicast tree (final tree) $T' = (V', E')$ is a subgraph of G, where E' represents the communication links in the final tree, and V' is the set of nodes (i.e., sources, forwarders and receivers) included in the final tree. In order to evaluate the relative importance of nodes and the benefits gained when accepted in the network, we propose a *utility* function for each source in the network (represented below by the objective function). The utility of streaming data from node v_i is denoted by U_i. This utility depends on the minimum acceptable data rate W_{min} by node v_i, and the maximum possible data rate of a node W_{max}. It also depends on the residual energy e_i, while r_i is the current rate (i.e., ratio to W_{max}) of data generated at source node v_s. The utility of streaming data decreases with decreasing received data rate and available energy level, and the utility becomes insignificant beyond a certain value (i.e., R_{min} in this case). In cases where the data rate for a given source node falls below the acceptable level (i.e., R_{min}), we propose to stop live data streaming and put the node offline.

Integer Linear Program Formulation. Let z_i be a 0-1 integer variable for each node $v_i \in V$, such that $z_i = 1$ if the node v_i is accepted as a traffic source in the resulting tree $v_i \in V'$. Let r_i be a positive real variable for each $v_i \in V$, representing the effective data rate of v_i such that $r_i = 0$ if v_i is not included in the resulting routing tree (i.e., $z_i = 0$). Let X^s_{ij} be a 0-1 integer parameter for each edge $(v_i, v_j) \in E$, $X^s_{ij} = 1$ if the edge is included in the resulting tree (i.e., edge $(v_i, v_j) \in E'$) for a given session (i.e., source) $s \in V_S$. Furthermore, let y^s_{ij} be a positive integer variable for each edge $(v_i, v_j) \in E'$, showing the amount of data transmitted from node v_i to node v_j for session $s \in V_S$. The ILP for the rate (r) and bandwidth (y) allocation problem can thus be stated as follows:

Objective function (Utility)

$max \sum_{i \in V_S} (z_i \times U_{min} + (r_i - (R_{min} \times z_i)) \times e_i \times U^r_{step})$

The first term of the utility function is the minimum utility for each source node in the network, the second term denotes the utility evolution with rate and residual energy. Multiplying each term by z_i guarantees the consideration of the included vertices nodes in the resulting tree only, and multiplying the second term by the coefficient U^r_{step} ensures utility evolution rate. Also, each accepted node $v_i \in V'_S$ is assigned rate $r_i \geqslant R_{min}$. The coefficient U_{min} is the minimum utility of each accepted source node v_i, and must be set to any positive value greater than zero .

Constraints

$$X^s_{ij} \leqslant y^s_{ij}, \forall s \in V_S, \forall i \in V, \forall j \in V : (i, j) \in E \tag{1}$$

$$X^s_{ij} \leqslant \sum_{\forall s \in V_S} y^s_{ij} \leqslant L_{ij}, \forall s \in V_S, \forall i \in V, \forall j \in V : (i, j) \in E \tag{2}$$

$$\sum_{\forall s \in V_S, \forall j \in V:(j,i) \in E} y^s_{ji} \times X^s_{ji} \leqslant C_i, \forall i \in V \tag{3}$$

$$y^s_{ij} \times X^s_{ij} = r_i \times W_{max} \times X^s_{ij}, \forall s \in V_S, \forall i \in V_S, \forall j \in V \tag{4}$$

$$y^s_{ij} \times X^s_{ij} = r_i \times W_{max} \times X^s_{ij}, \forall s \in V_S, \forall i \in V, \forall j \in V \tag{5}$$

$$z_i \geqslant r_i, \forall i \in V_S \tag{6}$$

$$r_i \geqslant R_{min} \times z_i, \forall i \in V_S \tag{7}$$

$$r_i = 0, \forall i \in V_F \cup V_R \tag{8}$$

$$z_i = 0, \forall i \in V_F \cup V_R \tag{9}$$

$$e_i \geqslant E_{min} \times X_{ij}, \forall i \in V, \forall j \in V : (i,j) \in E \tag{10}$$

Constraint (1) ensures not included edges have uplink effective rate of zero. Constraint(2) ensures that the total uplink effective rate of each included edge for all sources (i.e., sessions) in the resulting tree is bounded by the maximum physical link capacity. Constraint (3) provides an upper bound (i.e., the cell capacity) on the relay load constraint, it ensures that the incoming flow is always less than cell capacity C_i. Constraints (4) and (5) are for flow conservation. Constraint (4) implies that for each source node all included uplink edges have the same rate, which is the source data rate, and constraint (5) ensures that the data rate for all edges included in a session (i.e., each source node is represented by one session) $v_i \in V'_S$ are equal to the data rate assigned to that source. Since all data flows are originated from source nodes and do not return to the nodes, it will not lead to cycles in our solution. All data flows will eventually reach the receivers . Constraint (6) ensures that node data rate is assigned to accepted sources in the final multicast tree only, i.e., nodes not included in the resulting tree have rate equal to zero. Constraint (7) ensures that each accepted source node $v_i \in V'_S$ in the resulting tree has to be assigned rate $r_i \geqslant R_{min}$, this is to satisfy the QoS requirements. Constraints (8) and (9) ensure that forwarder and receiver nodes have rate equal to zero and can not be accepted as traffic source in the solution, respectively. Finally, Constraint (10) ensures that available energy for each accepted node v_i in the resulting tree has to be $\geqslant E_{min}$, as E_{min} is the residual energy value where a node is still able to send/receive messages properly.

5 Performance Evaluation

For the evaluation, we have conducted an extensive set of experiments using a VC++ coded simulator. To benchmark our protocol, we compare it to three other multicast routing protocols, namely *shortest-path tree* (SPT), *minimum transmissions tree* (min Tx Tree) and *Steiner tree*. To ensure the fairness of the comparison, we used the optimal bandwidth and rate allocation model presented in Sect. 4.2 for all solutions. To be accurate, we solved the integer linear program presented in Sect.4.2 by using AMPL and CPLEX. In the simulations, we generate random topologies with 50 nodes deployed in a 500 x 500 m^2 terrain, where number of sources varies from 1 to 7 and number of receivers is fixed at 10 (i.e., 20%), both sources and receivers are randomly selected in each run.

The wireless channel capacity is $C = 2$ Mbps, and link capacity $L = 1$ Mbps. The transmission range R_T is 150 m. Each source node is sending a Constant Bit Rate (CBR) stream to the sink with radios capable of transmitting up to 512 Kbps. The minimum acceptable data rate generated by each node R_{min} is 128 Kbps and R_{max} is fixed at 512 kbps. The minimum utility of each accepted node $U_{min} = 100$ and $U^r_{step} = 1/5$. For all simulation results in this paper, each experiment is an average of 10 different random topologies, and the simulation lasts for 5000 s.

Utility and Throughput: Figures 2 and 3 plot the utility and throughput, respectively, as a function of number of source nodes. The utility of a data transmission rate below R_{min} (i.e., 128 kbps) is considered to be insignificant, hence, the node is put offline. Utility for each offline node is assumed to be zero. Other settings are the same as above. We can clearly see in Fig. 2 that MMBA protocol experiences a utility increase with the increase of number of source nodes, which yields a better utility. This is because as more source nodes are deployed, MMBA will try to accommodate as many of them as possible at the best possible data rate, which results in a better throughput and thus a better utility. Figure 3 shows the throughput with respect to the number of source nodes in the network. However, among all evaluated approaches, the proposed solution MMBA works the best which leads to the highest network throughput, followed by the performance of Steiner Tree, SPT and min Tx Tree, respectively. This is because MMBA selects route with maximal possible residual capacity at each hop, thus the packets are dispersed widely and concurrent transmission can be fully utilized. It is worth mentioning that although sometimes the shortest path routing does not necessarily indicate the higher network throughput, which is true in most cases, it performs better than min Tx Tree in terms of throughput in this experiment.

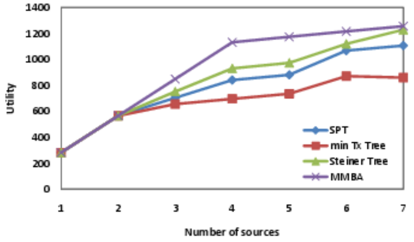


Fig. 2. Network utility at increasing number of sources

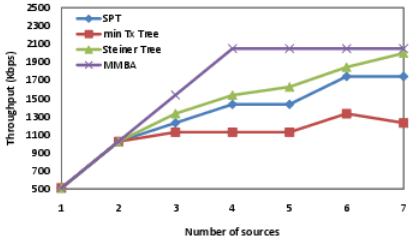
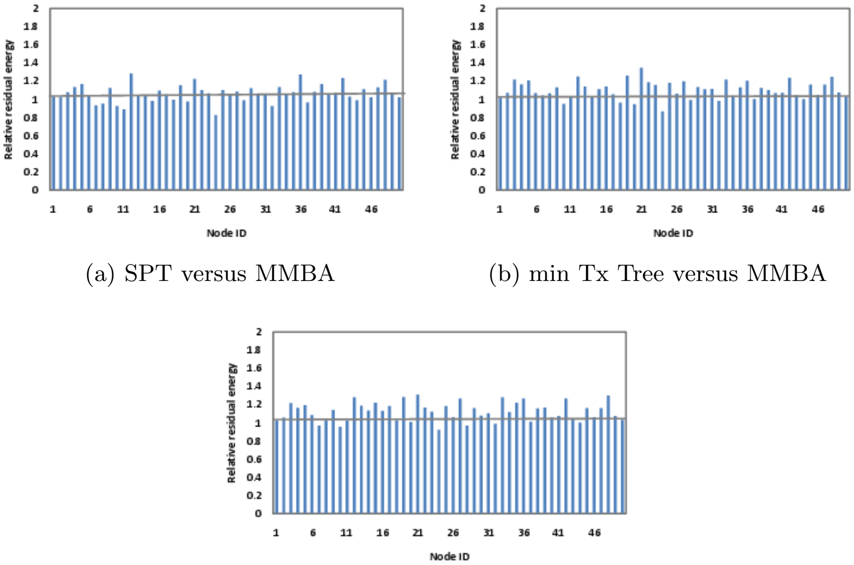


Fig. 3. Network throughput at increasing number of sources

Load Distribution: Network lifetime is a crucial metric of a WSN, but it can be a crude measure of actual energy consumption because node life is binary, so n about to die nodes are considered good as n fresh nodes. In order to observe how well MMBA promotes load balancing among the nodes, we instead plot the relative residual energy per node. We ran a simulation for 2000 s, at the outset, each node had 2 J battery energy. For simplicity, we only account for the radio receiving and transmitting energy. Figures 4(a), (b) and (c) show relative residual energy across nodes of SPT, min Tx Tree and Steiner Tree, respectively, at the end of simulation as compared to the performance with MMBA (shown as horizontal solid line). Each bar represents the difference of residual energy of SPT, min Tx Tree and Steiner Tree divided by the residual energy in MMBA. The plots are separated to avoid clutter. It is obvious that Steiner Tree and min Tx tree achieve the best performance by maintaining a higher value of residual energy. It is interesting to note that MMBA performs almost as good as Steiner Tree and min Tx Tree in terms of residual energy even though it delivers the highest throughput in the network. In addition, given that the SPT enables a node to reach the sink using the minimum number of hops, but does nothing to balance network load. SPT approach does not aim at minimizing the cost of the trees, it shows a lower performance compared to any of other approaches, consistent with the result reported in Figs. 5 and 6.



(a) SPT versus MMBA

(b) min Tx Tree versus MMBA

(c) Steiner Tree versus MMBA

Fig. 4. Relative residual energy distribution of nodes after 2000 s simulation of MMBA compared to SPT, min Tx Tree and Steiner Tree. The nodes are initially equipped with 2 J battery energy.

Multicast Routing Trees Size and Number of Transmissions Required: Figure 5 shows the impact of the number source nodes on the total size of the routing trees (i.e., total number of nodes in the resulting routing backbone) that each solution obtains. We note that all curves depict a larger routing tree for larger number of high priority nodes, which suggests that when increasing number of high priority nodes in the network the proposed protocol tries to accept as many nodes in the network as possible and thus increases the routing tree size. Tree size influences the overall number of transmissions and consumed energy and thus network operable lifetime. The larger the tree the more energy consumed in the network. As expected the Steiner Tree is the one offering the lowest tree size, while the min Tx Tree as well as the other approaches offer a higher mean tree size. This is clearly due to the fact grouping paths for several receivers makes them not to use their shortest paths (e.g., SPT and MMBA) and minimum number of transmissions (e.g., min Tx tree). As we can see, this metric is much more variable to the number of source nodes than the number of transmissions as depicted in Fig. 6. Consistent result is also obtained for the number of transmission required as shown in Fig. 6. As more nodes are selected in the multicast routing tree, the overall number of transmissions becomes higher. From Fig. 5, we realize that more number of nodes are obtained by the proposed MMBA solution. This is because shorter links can support higher data rates (capacity). It is often possible to obtain higher throughput by multi-hopping since higher data rates are used. As the distance increases, more robust burst profiles (modulation and coding techniques) are needed to reduce bit error rate (BER) which results in lower data rate. Another point is that MMBA always favor a higher capacity path to the one with less number of hops in order to maximize network throughput.

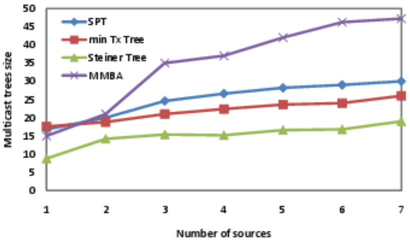


Fig. 5. Multicast routing tree size at increasing number of sources

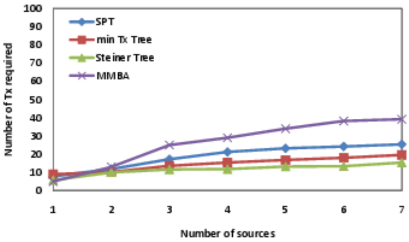
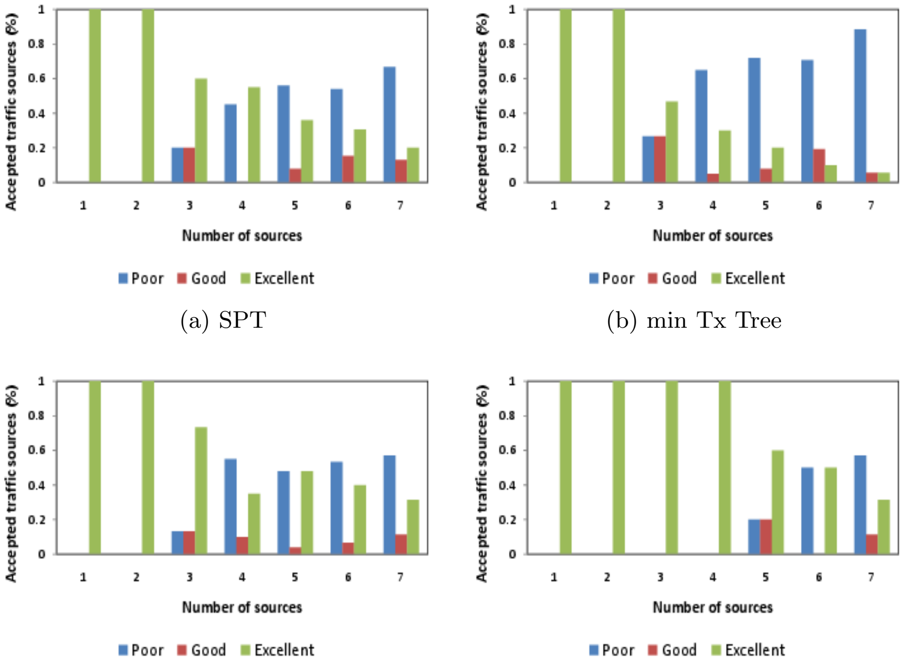


Fig. 6. Total number of transmissions at increasing number of sources

Quality of Received Data Streams: The influence of number of source nodes deployed on the data quality received from the accepted source nodes is also evaluated. Figures 7(a) to (d) show the impact of different number of source nodes on the streaming data quality coming from the accepted source nodes, while the data streams is the actual information transferred across the wireless links. We classified the data quality based on the allocated data rate into three levels: poor (128 - 256 Kbps), good (256 - 384 Kbps) and excellent (384 - 512 Kbps). From the figures, we can see that the percentage of source nodes accepted at better quality, and thus higher utility, decreases with the the number of source nodes in all solutions. However, it is apparent that the MMBA protocol incurs the best results with fair distribution of the available bandwidth across the accepted



(a) SPT

(b) min Tx Tree

(c) Steiner Tree

(d) MMBA

Fig. 7. Received data quality of accepted source nodes as a function of number of source nodes

source nodes followed by Steiner Tree, SPT and then min Tx Tree, respectively. The MMBA protocol outperforms the other solutions, this is because more routes are available in the network with possibly better capacity, and the MMBA is able to select a better capacity one. As a result, the best data rates are allocated to source nodes.

6 Conclusion

This paper proposes MMBA, a cross-layer heuristic solution for multisession multicast tree construction with bandwidth allocation for wireless sensor networks. It uses fuzzy logic to evaluate nodes' and network conditions taking into account hop count, residual energy and available bandwidth at each node. MMBA adopts an adaptive resource allocation policy to continuously meet QoS requirements. In this paper, we confirm that minimum transmission tree and Steiner tree are not best-suited for wireless sensor networks where available bandwidth is very limited. The performance evaluation has shown that MMBA is able to overcome other evaluated approaches in terms of throughput, capacity and network utilization, while balancing out energy consumption among nodes in the network ensures longer operational network lifetime.

Acknowledgment. The work described in this paper is carried out under the auspices of Enterprise Ireland Applied Research Enhancement (ARE).

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks (38), 393–422 (2001)
2. Huitema, C.: Routing in the Internet, 2nd edn. Prentice Hall (1999)
3. Abdollahpouri, A., Wolfinger, B.E., Lai, J.: Unicast versus Multicast for Live TV Delivery in Networks with Tree Topology. In: Osipov, E., Kassler, A., Bohnert, T.M., Masip-Bruin, X. (eds.) WWIC 2010. LNCS, vol. 6074, pp. 1–14. Springer, Heidelberg (2010)
4. Guo, S., Yang, O.W.W.: Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. Computer Communications 30, 2129–2148 (2007)
5. Wieselthier, J., Nguyen, G., Ephremides, A.: On the constuction of energy-efficient broadcast and multicast trees in wireless networks. In: Proc. of the IEEE INFOCOM, pp. 585–594 (2000)
6. Wang, P.J., Calinescu, G., Yi, C.W.: Minimum-power multicast routing in static ad hoc wireless networks. IEEE/ACM Trans. Netw. 12(3), 507–514 (2004)
7. Ruiz, P.M., Gomez-Skarmeta, A.F.: Approximating optimal multicast trees in wireless multihop networks. In: Proc. of the IEEE ISCC (2005)
8. Frey, H., Ingelrest, F., Simplot-Ryl, D.: Localized minimum spanning tree based multicast routing with energy-efficient guaranteed delivery in ad hoc and sensor networks. In: Proc. of the IEEE WoWMoM (2008)
9. Cheng, L., Das, S.K., Cao, J., Chen, C., Ma, J.: Distributed minimum transmission multicast routing protocol for wireless sensor networks. In: Proc. of the ICPP (2010)
10. Nguyen, U.T., Asif, A., Xiong, X.: Multirate-aware multicast routing in manets. In: Proc. of the IEEE MASS (2006)
11. Singh, G., Pujar, S., Das, S.: Rate-based data propagation scheme in sensor networks. In: Proc. of the IEEE WCNC (2004)
12. Zhu, Y., Pu, K.Q.: Adaptative multicast tree construction for elastic data streams. In: Proc. of the IEEE GLOBECOM (2008)
13. Plesnik, J.: The complexity of designing a network with minimum diameter. Networks 11, 77–85 (1981)
14. Mendel, J.M.: Fuzzy logic systems for engineering: a tutorial. Proc. of the IEEE 83, 345–377 (1995)
15. Su, W., Bougiouklis, T.C.: Data fusion algorithms in cluster-based wireless sensor networks using fuzzy logic theory. In: Proc. of the 11th WSEAS Int. Conf. on Communications (2007)

Reliable Broadcast Protocol Independent of System Parameters for Ad Hoc Networks with Liveness Property

Jerzy Brzeziński, Michał Kalewski, and Cezary Sobaniec

Institute of Computing Science
Poznań University of Technology
Piotrowo 2, 60–965 Poznań, Poland
{Jerzy.Brzezinski,Michal.Kalewski,Cezary.Sobaniec}@cs.put.poznan.pl

Abstract. The *MANET liveness property* ensures that any partition in an ad hoc network is not permanently isolated. For networks that fulfil the property a few crash-tolerant broadcast protocols have been proposed. However, it has also been proved that the minimum time of direct connectivity between nodes, and thus the correctness of the protocols, depends on the total number of hosts in a network and on the total number of messages that can be disseminated by each node concurrently. In this paper, we propose an improved version of the reliable broadcast protocols that works correctly, even though the minimum time of direct connection between nodes allows them to exchange (send and respond to) at least only two messages, making the correctness of this protocol independent of system parameters.

1 Introduction

Mobile ad hoc networks or MANETs [10, 1, 2] are composed of autonomous and mobile communication devices which communicate through wireless links without any stable network infrastructure or central points. The distance from a transmitting device at which the radio signal strength remains above the minimal usable level is called the *transmission* (or *wireless*) *range* of that host. Each pair of such devices, whose distance is less than their transmission range, can communicate directly with each other. A message sent by any host may be received by all hosts in its vicinity. Hosts can come and go or appear in new places, so the resulting network topology may change all the time and can get partitioned and reconnected in a highly unpredictable manner. Thus, mobile hosts in an ad hoc network can exchange information in areas that do not have preexisting infrastructure in a decentralised way (control of the network is distributed among the hosts).

One of the fundamental communication operations in ad hoc networks is broadcast—a process of sending a message from one host to all hosts in a network. It is important for any broadcast protocol to provide some delivery

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 358–370, 2012.

guarantee beyond "best-effort", since broadcast is a basic communication requirement to construct other and more complex distributed algorithms like consensus or coherency protocols. But in case of dynamic environments this can be hard or even impossible to achieve [6]. Therefore, heuristic broadcast protocols with only probabilistic guarantees have been mainly proposed for the use in MANETs (e.g. [4, 9]). On the other hand, the *MANET liveness property*, which ensures that no partition is allowed to be permanently isolated, allows us to develop reliable broadcast (or dissemination) protocols with deterministic guarantees [11–13].

For networks that fulfil the MANET liveness property a few crash-tolerant broadcast protocols have been proposed. However, it has also been proved that the minimum time of direct connectivity between nodes, and thus the correctness of these protocols, depends on the total number of hosts in a network and on the total number of messages that can be disseminated by each node concurrently [3].

In this paper, we propose an improved version of the reliable broadcast protocols that works correctly, even though the minimum time of direct connection between nodes allows them to exchange (send and respond to) at least only two messages, making the correctness of the protocol independent of system parameters.

The paper is organised as follows. First, following [12, 13], the formal model of ad hoc systems with the liveness property is described in Section 2. A short overview of the crash-tolerant broadcast protocols proposed in [12, 13] is presented in Section 3. Section 4 contains the modified version of the reliable broadcast protocols with time constraints independent of system parameters along with its proof of correctness. Finally, the paper is shortly concluded in Section 5.

2 System Model

In this paper, a *distributed ad hoc system* is considered. The units that are able to perform computation in the system are abstracted through the notion of a *host* (or *node*), and a *link* abstraction is used to represent the communication facilities of the system. It is presumed that each link always connects two nodes in a bidirectional manner. Thereby, the topology of the distributed ad hoc system is modelled by an undirected *connectivity graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of all nodes, $p_1, p_2, \ldots, p_n$, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of links between neighbouring nodes, i.e. nodes that are within transmission range of each other. If node p_i is able to communicate directly with node p_j, then there exists link (p_i, p_j) in set $\mathcal{E}$. (Note that (p_i, p_j) and (p_j, p_i) denote the same link, since links are always bidirectional.) The set $\mathcal{E}$ changes with time, and thus the graph $\mathcal{G}$ can get disconnected and reconnected. Disconnection fragments the graph into isolated sub-graphs called *components* (or *partitions* of the network), such that there is a path in $\mathcal{E}$ for any two nodes in the same component, but there is no path in $\mathcal{E}$ for any two nodes in different components.

2.1 Nodes and Communication

The considered system is composed of $N = |\mathcal{V}|$ uniquely identified nodes and each node is aware of the number of all nodes in the $\mathcal{V}$ set (that is of N). A node is either *correct* or *faulty*: a faulty node can crash (stops its processing) at any moment without any warning (*silent crash failure model*, [5]), while a correct node does not crash until processing ends. A crash need not be detectable. It is also presumed that the number of faulty nodes is bound to some known value f, such that: $0 \leqslant f < \frac{1}{2}N$. Thus, the $\mathcal{V}$ set contains at least $N - f$ correct nodes and $N - f > \frac{1}{2}N$. A node that has not crashed is said to be *operative*.

The nodes communicate with each other only by sending messages (*message passing*). Any node, at any time can initiate the dissemination of message m, and all nodes that are neighbours of the sender, at least for the duration of a message transmission, can receive the message. More formally, the links can be described using the concept of a *dynamic set function* [8]. Let $\mathcal{E}'$ be a product set of $\mathcal{V}$: $\mathcal{E}' = \mathcal{V} \times \mathcal{V}$, and $\Gamma(\mathcal{E}')$ be the set of all subsets (power set) of $\mathcal{E}'$: $\Gamma(\mathcal{E}') = \{\mathcal{A} \mid \mathcal{A} \subseteq \mathcal{E}'\}$. Then, the dynamic set function $\mathcal{E}_i$ of node p_i is defined as follows:

Definition 1 (Dynamic Set Function). *The **dynamic set** $\mathcal{E}_i$ of node p_i in some time interval $T = [t_1, t_2]$ is a **function**:*

$$\mathcal{E}_i : T \to \Gamma(\mathcal{E}')$$
$$\textit{such that } \forall t \in T : \ \mathcal{E}_i(t) \textit{ is a set of all links of } p_i \textit{ at time } t.$$

Let δ be the maximum message transmission time between neighbouring nodes. Then, we introduce the abstraction of a *reliable channel* (**RC**), as presented by Module 1. The interface of this module consists of two events: a *request event*, used to send a message, and an *indication event*, used to deliver the message. Reliable channels do not alter and lose (**RC1** *reliable delivery* property), duplicate (**RC2** *no duplication* property), or create (**RC3** *no creation* property) messages.

Module Name:

Reliable Channel (RC).

Events:

Request: $\langle$ *rc.Send*, m $\rangle$: Used to send message m.

Indication: $\langle$ *rc.Deliver*, p_s, m $\rangle$: Used to deliver message m sent by process p_s.

Properties:

RC1 (*Reliable Delivery*): Let p_s and p_d be any two nodes that are within a wireless range of each other, and let p_s send message m at some time t to p_d. If the two nodes remain operative at least until $t + \delta$, then message m is delivered by p_d within δ.

RC2 (*No duplication*): No message is delivered by any node more than once.

RC3 (*No creation*): If message m is delivered by some node p_d, then m was previously sent by p_s.

Module 1. Interface and properties of reliable channels

Finally, we can define *direct connectivity* as follows ([13, 12]):

Definition 2 (Direct Connectivity). *Let $T = [t,\, t + B]$, where $B \gg \delta$ is an application-specified parameter. Then, two operative nodes p_i and p_j are said to be **directly connected** at t iff:*

$$\forall \tau \in T\ (\,(p_i,\, p_j) \in \mathcal{E}_i(\tau)\,).$$

It is assumed that channels between directly connected hosts are *reliable channels.*

2.2 Network Liveness Property

Let $\mathcal{O}$ be a set of all operative nodes of $\mathcal{V}$ at some time t ($\mathcal{O} \subseteq \mathcal{V}$). Let $\mathcal{P}$ be a non-empty subset of $\mathcal{O}$, and $\overline{\mathcal{P}}$ be complementary set of $\mathcal{P}$ in $\mathcal{O}$ ($\overline{\mathcal{P}}$ contains all operative nodes at time t that are not in $\mathcal{P}$). Then, the *network liveness property* is specified as follows ([13, 12]):

Definition 3 (Network Liveness Property). *A distributed ad hoc system that was initiated at t_0 satisfies the **network liveness property** iff:*

$$\forall t \geqslant t_0\, \forall \mathcal{P}\, \exists I \geqslant B\ (\,I \neq \infty\ \wedge\ \exists\{p_i,\, p_j\}\ (\,p_i \in \mathcal{P}\ \wedge\ p_j \in \overline{\mathcal{P}}\ \wedge \\ (\textit{nodes } p_i \textit{ and } p_j \textit{ are directly connected at some } \tau \in [t,\, t + I]))).$$

In other words:

$$\forall t \geqslant t_0\, \forall \mathcal{P}\, \exists I \geqslant B\ (\,I \neq \infty\ \wedge\ \exists\{p_i,\, p_j\}\ (\,p_i \in \mathcal{P}\ \wedge\ p_j \in \overline{\mathcal{P}}\ \wedge \\ (\,\exists\{t_1,\, t_2\}\ (\,(t \leqslant t_1 < t_2 \leqslant t + I)\ \wedge\ (t_2 - t_1 \geqslant B)\ \wedge \\ (\forall t_c \in [t_1,\, t_2]\ ((p_i,\, p_j) \in \mathcal{E}_i(t_c)))))).$$

Informally, the network liveness property prevents permanent partitioning to occur by requiring that reliable direct connectivity must emerge between some nodes of every $\mathcal{P}$ and its complementary set $\overline{\mathcal{P}}$ within some finite, but unknown, amount of time I after each t.

3 Reliable Broadcast Protocols

Broadcast protocols enable us to send a message from one host to all hosts in a network, and are a basis of communication in ad hoc networks. It is important for any broadcast protocol to provide some delivery guarantee, especially if host failures are taken into account. The properties of broadcast operations considered in this paper are described by Module 2. The interface of this module consists of two events: a *request event*, used to broadcast a message, and an *indication event*, used to deliver the broadcast message. The **BCAST1** *progress* property ensures that at least $N - f - 1$ operative hosts will receive each disseminating message, and the **BCAST2** *termination* property ensures that every node eventually

Module Name:

Broadcast Protocol (BCAST).

Events:

Request: $\langle$ *bcast.Broadcast*, m $\rangle$: Used to broadcast message m.

Indication: $\langle$ *bcast.Deliver*, p_s, m $\rangle$: Used to deliver message m broadcast by process p_s.

Properties:

BCAST1 (*Progress*): For a broadcast of message m initiated at time t_b, at least $N - f - 1$ operative nodes (or $N - f$ including originator) receive the message within some bounded time if the sender does not crash, or if the sender crashes and a correct node receives m.

BCAST2 (*Termination*): Each node that has received m and remains operative, including originator, will discard m and stop transmitting any packets concerning the broadcast of m at some time after t_b.

Module 2. Interface and properties of the broadcast protocols

discards every disseminating message and, as a result, the broadcast of every message can be eventually terminated.

In [13, 12] the following protocols have been proposed to implement Module 2: (i) *Proactive Dissemination Protocol* (PDP), (ii) *Reactive Dissemination Protocol* (RDP), (iii) *Proactive Knowledge and Reactive Message* (PKRM) and (iv) *Optimised PKRM* (PKRM$_O$).

Proactive Dissemination Protocol. The simplest protocol of the four, *Proactive Dissemination Protocol*, requires that each node, which has message m, transmits it once every β seconds. Originator p_i of message m adds it to its $\mathcal{U}nrealised_i$ set (a set of all received but not yet realised messages) and initialises vector $K_i(m)$, as a boolean vector of N bits, to all zeros, and sets its own bit ($K_i(m)[i]$) to 1. The vector indicates the *knowledge* of the node on the propagation of m, i.e. $K_i(m)[j] = 1$ means that node p_i knows that message m has been received by node p_j. The message is always transmitted along with the $K_i(m)$ vector. When some node p_j receives m for the first time, it initialises its $K_j(m)$ vector to be equal to the received $K_i(m)$ and sets its own bit $K_j(m)[j]$ to 1. The message is also added to the $\mathcal{U}nrealised_j$ set. Then, whenever any host receives the message, it merges the received vector with its own using a bitwise OR operation. If node p_i has $N - f$ or more 1-bits in its $K_i(m)$ vector, it *realises* m, i.e. it cancels the periodic transmission of m, and moves the message to the $\mathcal{R}ealised_i$ set. If a node receives a message which has been realised according to its knowledge, i.e. $m \in \mathcal{R}ealised_i$, then within β seconds it transmits a special realisation packet $realise_i(m)$, and a node which receives the packet, realises m.

The β parameter is originally defined as a configurable parameter such that: $B \geqslant 2(\beta + \delta)$, to ensure that a node both: receives a message and sends that message during direct connectivity.

Let us also note that the original specification of the protocol states only that a node that transmitted m once, will thereafter send it once every β seconds until its realisation. However, no particular order of message sending is imposed when a host has more that one message to send every β seconds, except that the messages that have been received for the first time during last β seconds must be sent within β seconds.

Reactive Dissemination Protocol. The *Reactive Dissemination Protocol* assumes that nodes have information about their direct neighbours, i.e. each host knows its dynamic set $\mathcal{E}_i(t)$ for all t. In this case, node p_i propagates m only if: $\exists\{p_i, p_j\} \in \mathcal{E}_i(t)\ (\,K_i(m)[j] = 0\,)$, which is evaluated once every β seconds (β is as in the PDP).

Proactive Knowledge and Reactive Message. The *Proactive Knowledge and Reactive Message* combines the features of the PDP and RDP without requiring information about direct neighbours. In the PKRM protocol, each p_i that transmitted m once as an originator, then sends every β seconds only $K\_ptr_i(m)$ packets, which contain only an identifier of message m and sender's $K_i(m)$ vector. If receiver p_j of the packet does not have m, it sends within β seconds $request_j(m)$ packet, requesting m to be transmitted. Thus, if node p_i that has m, has received $request_j(m)$ packet in the past β seconds, it sends m. As in the first two protocols, β is fixed but this time: $B \geqslant 3(\beta + \delta)$ (to ensure that nodes exchange three times all messages during direct connectivity).

Optimised PKRM. Finally, in the PKRM$_O$ protocol, the number of message transmissions is optimised towards bandwidth reduction in the following way. First, hosts check messages received in the past every $\hat{\beta}$ seconds, where $\hat{\beta}$ is a random duration distributed uniformly in $(0, \beta)$. Second, a suppression of the proactive message dissemination has been added with the use of two additional counters: *eqvDataCount$_i$(m.id)* and *eqvKmCount$_i$(m.id)*. The former is incremented by one each time host p_i receives m. The *eqvKmCount$_i$(m.id)* counter is increased by one if node p_i receives m or $K\_pkt_j(m)$ and if the local $K_i(m)$ vector is equal to the received one—otherwise it is reset to 0. Then, if a transmission of m is expected, and *eqvDataCount$_i$(m.id)*$>\alpha$, the transmission is suppressed and *eqvDataCount$_i$(m.id)* is reset to 0. If a transmission of $K\_pkt_i(m)$ is expected, and *eqvKmCount$_i$(m.id)*$>\alpha$, the transmission is suppressed and *eqvKmCount$_i$(m.id)* is reset to 0. Parameter α is configurable *suppression threshold*.

It has been proved in [3], that for all these protocols the minimum value of the β parameter cannot be shorter than the total time of transmission of all messages that can be disseminated in the system by all nodes. To give it more precisely, let us also define a *limit* of the number of messages that a node can disseminate *concurrently*:

Definition 4 (Concurrent Dissemination Limit). *Node p_i can start disseminate new message m iff the number of messages originated by p_i in the $\mathcal{U}nrealised_i$ set is less than s, where s is* ***the concurrent dissemination limit****.*

In the simplest case when $s = 1$, the protocols work in a blocking manner. That is, a host will start disseminating a new message provided that the previous one, originated by it, is realised. Let $S = N \cdot s$ be the total number of messages that can be disseminated in the system by all nodes. It has been proved in [3] that if a network is composed of N nodes and if each node can disseminate concurrently at most s messages, then the PDP, RDP, PKRM and PKRM$_O$ protocols require that $\beta \geqslant N \cdot s \cdot \delta \geqslant S \cdot \delta$. This, in turn, may require in practise to adjust β configurable parameter when the number of nodes changes, and in extreme cases can lead to a difficulty in obtaining direct connections that ensure the liveness property, making the correctness of the protocols dependent on its parameters. Therefore, in next Section, we propose an improved version of the reliable broadcast protocols that works correctly even though the minimum time of direct connection between nodes allows them to exchange (send and respond to) at least only two messages, making the correctness of this protocols independent of system parameters.

4 A Reliable Broadcast Protocol Independent of System Parameters

In this Section, we introduce the *Time-adjusted Proactive Dissemination Protocol* (TaPDP) that implements the broadcast properties described by Module 2. The protocol also requires periodical message broadcasts, but with the use of this protocol each node broadcasts only one unrealised message within every β seconds, instead all of them. Moreover, to impose an order of messages, the TaPDP protocol uses Lamport's logical clocks [7].

Time-adjusted Proactive Dissemination Protocol. In this protocol, every message m is always transmitted in packets of the following structure:

$$[\ m,\ p_o,\ \mathtt{C}(m),\ K_i(m),\ R_i\ ],$$

where: p_o is an identifier of the originator of m; $\mathtt{C}(m)$ is a logical clock value associated with m by p_o; $K_i(m)$ is a knowledge vector (as in PDP) of sending node p_i, and R_i is a *realisation vector* of p_i. Vector R_i is of the size of N logical clock values, and $R_i[j] = v$ indicates that the logical clock value of the last message realised by p_i and originated by p_j is v.

The TaPDP protocol is presented in Algorithm 1, and it operates in the following manner. Originator p_i of message m increments its logical clock $\mathtt{C}_i$, initialises vector $K_i(m)$ (as in the PDP) and adds m, along with its identifier and the current logical clock value, to its $\mathcal{U}nrealised_i$ set (*bcast.Broadcast* **event**, lines: 4–10). Once every β seconds each node, which has at least one unrealised message, sends only one of them (**every** β *seconds* **event**, lines: 12–17). As a message to send p_i selects the one, which has the smallest value of its logical clock $\mathtt{C}(m)$ of all messages contained in the $\mathcal{U}nrealised_i$ set (line 13). If there are more than one messages with the same logical clock values equal to the smallest value, the message that has been originated by a node with the smallest identifier is selected (line 14). In this way, a single message is always chosen (line 15) in a *total order* on the $\mathcal{U}nrealised_i$ set. Namely, for any two messages: $m \in \mathcal{U}nrealised_i$ originated by p_j and $m' \in \mathcal{U}nrealised_i$ originated by p_k, m will be selected for broadcast before m' provided that $\mathtt{C}(m) < \mathtt{C}(m')$, or $\mathtt{C}(m) = \mathtt{C}(m')$ and $j < k$. Let us denote this by $m \xrightarrow{j<k} m'$. This relation is transitive, antisymmetric and total, and hence places also a total order on the set of all unrealised messages in the system [7].

Next, whenever node p_i receives a new message originated by p_o, i.e. a message that has not yet been realised by p_i according to its realisation vector ($\mathtt{C}(m) > R_i[o]$) and is not in its $\mathcal{U}nrealised_i$ set (*rc.Deliver* **event**, lines: 22–28), it updates its logical clock $\mathtt{C}_i$ (line 23) and $\mathcal{U}nrealised_i$ set (line 24), and creates its own $K_i(m)$ vector (lines: 25–26) for the received message. If, in turn, p_i receives a message that already is in its $\mathcal{U}nrealised_i$ set, then it only updates its $K_i(m)$ vector for that message (lines: 28–29). After that, the node checks if it has $N - f$ or more 1-bits in the $K_i(m)$ vector (line 30). If this is the case, p_i *realises* m, that is it removes m from its $\mathcal{U}nrealised_i$ set (line 31) and updates its realisation vector $R_i[o]$ to be equal to logical clock value $\mathtt{C}(m)$ of the realised message (line 32). Consequently, since all messages originated by p_o are selected for broadcast in the total order to their logical clock values, the value $R_i[o] = v$ means that all messages, originated by p_o for which $\mathtt{C}(m) \leqslant v$ holds, have already been realised.

It is possible that some messages have been realised by other operative nodes, while they are still unrealised by p_i. Therefore, p_i also updates its $\mathcal{U}nrealised_i$ set (line: 33 and *Update* **event**, lines: 37–42) by removing all messages originated by every $j := 1 \ldots N$ for which $R_s[j] > R_i[j]$ and $\mathtt{C}(m) \leqslant R_s[j]$ hold (and m is originated by p_j), and updates its realisation vector accordingly (lines: 38–41).

Finally, if p_i does not have any unrealised messages and receives a messages that is already realised, it broadcasts only its realisation vector R_i within next β seconds—this is indicated by the *Rsend* boolean variable (*rc.Deliver* **event**, lines: 34–35 and **every** β *seconds* **event**, lines: 17–19). If p_i receives the R_s vector, it updates its $\mathcal{U}nrealised_i$ set as above by triggering *Update* **event** (*rc.Deliver* **event**, lines: 43–45). Thereby, a node is able to receive information about realised messages, even though no unrealised message is broadcast periodically by its neighbour(s).

The minimal time constraints required by the TaPDP are as follows:

1. $\beta \geqslant \delta$: just to ensure that a message is not sent until the previous transmission is completed.
2. $B \geqslant 2(\beta + \delta) \geqslant 4\delta$: to ensure that each directly connected node is always able to receive a message and send a message within next β seconds.
   This is illustrated in Figure 1, where $\beta = 2\delta$ and $B = 2(\beta + \delta) = 6\delta$.
   In Figure 1(a) nodes p_i and p_j (which does not have unrealised messages) directly connect just after p_i sent its unrealised message at τ_1 (first arrow in the Figure). The message, at this time, cannot be received by p_j. After another β seconds (counted by p_i), the message is sent again at τ_3, and, at this time, p_j is able to receive it (second arrow in the Figure). Next, after

Algorithm 1. The Time-adjusted Proactive Dissemination Protocol

```
Upon event $\langle\, Init \,\rangle$ at node $p_i$ do
    $Unrealised_i \leftarrow \emptyset,\ \forall_{j:=1\ldots N}: R_i[j] := 0,\ \mathtt{C}_i := 0,\ Rsend := false$
End upon event

Upon event $\langle\, bcast.Broadcast,\ m \,\rangle$ at node $p_i$ do
    $\mathtt{C}_i := \mathtt{C}_i + 1$
    $\forall_{j:=1\ldots N}: K_i(m)[j] := 0$
    $K_i(m)[i] := 1$
    $Unrealised_i \leftarrow Unrealised_i \cup \{\, [\, m,\ p_i,\ \mathtt{C}_i \,] \,\}$
    trigger $\langle\, bcast.Deliver,\ p_i,\ m \,\rangle$
End upon event

Every $\beta$ seconds at node $p_i$ do
    if $Unrealised_i \neq \emptyset$ then
        $\mathtt{C}(m) := \mathsf{min}\,(\{\, \mathtt{C}(m') \mid [\, m',\ p_k,\ \mathtt{C}(m') \,] \in Unrealised_i \})$
        $o := \mathsf{min}\,(\{\, k \mid [\, m',\ p_k,\ \mathtt{C}(m') \,] \in Unrealised_i \wedge \mathtt{C}(m') = \mathtt{C}(m) \,\})$
        $m := m' :\ [\, m',\ p_k,\ \mathtt{C}(m') \,] \in Unrealised_i \wedge \mathtt{C}(m') = \mathtt{C}(m) \wedge k = o$
        trigger $\langle\, rc.Send,\ [\, m,\ p_o,\ \mathtt{C}(m),\ K_i(m),\ R_i \,] \,\rangle$
    else if $Rsend$ then
        trigger $\langle\, rc.Send,\ R_i \,\rangle$
    $Rsend := false$
End

Upon event $\langle\, rc.Deliver,\ p_s,\ [\, m,\ p_o,\ \mathtt{C}(m),\ K_s(m),\ R_s \,] \,\rangle$ at node $p_i$ do
    if $\mathtt{C}(m) > R_i[o] \wedge [\, m,\ p_o,\ \mathtt{C}(m) \,] \notin Unrealised_i$ then
        $\mathtt{C}_i := \mathsf{max}(\, \mathtt{C}_i,\ \mathtt{C}(m) \,) + 1$
        $Unrealised_i \leftarrow Unrealised_i \cup \{\, [\, m,\ p_o,\ \mathtt{C}(m) \,] \,\}$
        $\forall_{j:=1\ldots N}: K_i(m)[j] := K_s(m)[j]$
        $K_i(m)[i] := 1$
        trigger $\langle\, bcast.Deliver,\ p_o,\ m \,\rangle$
    else if $[\, m,\ p_o,\ \mathtt{C}(m) \,] \in Unrealised_i$ then
        $\forall_{j:=1\ldots N}: K_i(m)[j] := K_i(m)[j] \vee K_s(m)[j]$
    if $\sum_{j:=1}^{N} K_i(m)[j] \geqslant N - f \wedge [\, m,\ p_o,\ \mathtt{C}(m) \,] \in Unrealised_i$ then
        $Unrealised_i \leftarrow Unrealised_i \setminus \{\, [\, m,\ p_o,\ \mathtt{C}(m) \,] \,\}$
        $R_i[o] := \mathtt{C}(m)$
    trigger $\langle\, Update,\ R_s \,\rangle$
    if $Unrealised_i = \emptyset$ then
        $Rsend := true$
End upon event

Upon event $\langle\, Update,\ R_s \,\rangle$ at node $p_i$ do
    for $j := 1 \ldots N$ do
        if $R_s[j] > R_i[j]$ then
            $Unrealised_i \leftarrow Unrealised_i \setminus \{\, [\, m',\ p_k,\ \mathtt{C}(m') \,] \mid$
                      $[\, m',\ p_k,\ \mathtt{C}(m') \,] \in Unrealised_i \wedge k = j \wedge \mathtt{C}(m') \leqslant R_s[j] \,\}$
            $R_i[j] := R_s[j]$
End upon event

Upon event $\langle\, rc.Deliver,\ p_s,\ R_s \,\rangle$ at node $p_i$ do
    trigger $\langle\, Update,\ R_s \,\rangle$
End upon event
```

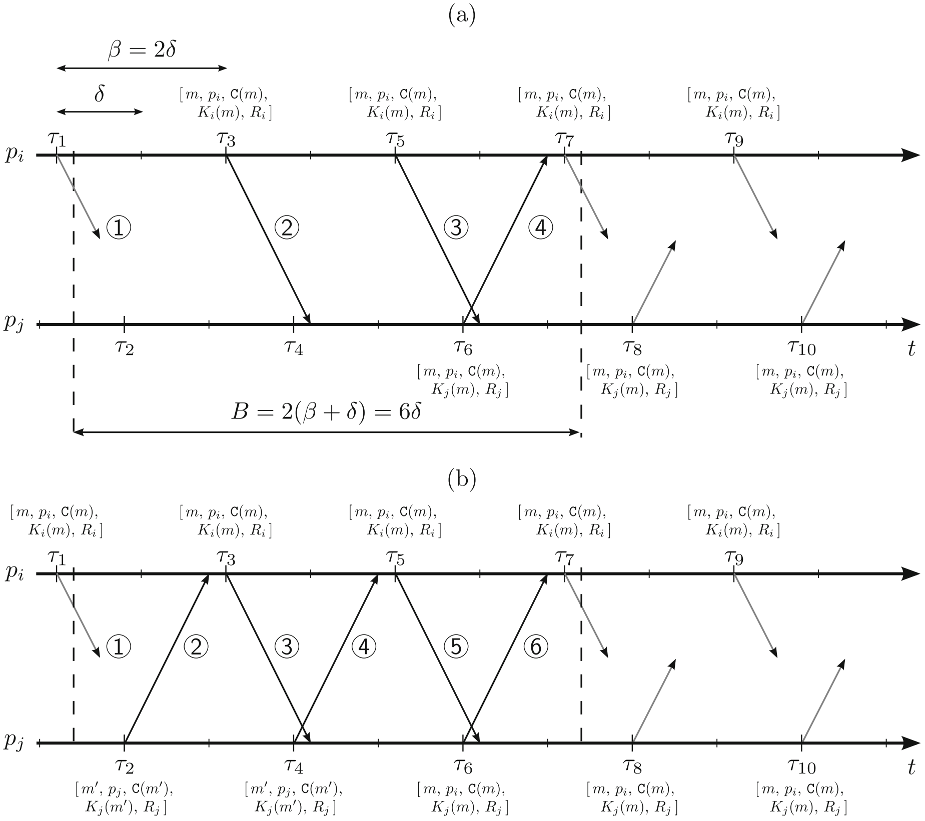

(a)
$\beta = 2\delta$
δ
$[m, p_i, \mathtt{C}(m), K_i(m), R_i]$
p_i
p_j
τ_1
τ_2
τ_3
τ_4
τ_5
τ_6
τ_7
τ_8
τ_9
τ_{10}
t
①
②
③
④
⑤
⑥
$[m, p_j, \mathtt{C}(m), K_j(m), R_j]$
$[m, p_i, \mathtt{C}(m), K_j(m), R_j]$
$B = 2(\beta + \delta) = 6\delta$
(b)
$[m', p_j, \mathtt{C}(m'), K_j(m'), R_j]$

Fig. 1. An example of direct connectivity between two nodes, p_i and p_j, with the use of the TaPDP protocol ($\beta = 2\delta$ and $B = 6\delta$): (a) only p_i has an unrealised message, (b) both nodes have unrealised messages

β seconds counted by p_j, the message (or vector R_j if the message has been realised by p_j) is sent by p_j at τ_6, and it is received by p_i (fourth arrow in the Figure).

Figure 1(b) shows similar direct connectivity, but in this case p_j also broadcasts periodically unrealised message m' such that: $m \xrightarrow{i<j} m'$. This message is sent by p_j at τ_2 and received by p_i just before τ_3 (seconds arrow in the Figure). Because $m \xrightarrow{i<j} m'$, then at τ_3 node p_i sends m, and in this way, p_j receives a new message (third arrow in the Figure). Node p_j will thereafter also broadcast periodically m, and thus, when it sends this message at τ_6 (sixth arrow in the Figure), p_i receives it and updates its $K_i(m)$ vector.

So, the parameters are independent of the total number of messages that each node can start disseminate concurrently, i.e. of the concurrent dissemination limit, and hence they are also independent of the total number of messages that can be disseminated in a system by all nodes simultaneously.

Now, the succeeding theorem regarding the protocol can be defined and proved:

Theorem 1. *If $\beta \geqslant \delta$ and $B \geqslant 2(\beta + \delta)$, then with the use of the TaPDP protocol, the **BCAST1** progress and **BCAST2** termination properties are met.*

Proof. Assuming $\beta \geqslant \delta$ and $B \geqslant 2(\beta+\delta)$, let us consider a system of $N > 1$ hosts, denoted $p_1, \ldots, p_N$, that was initiated at t_0, and let $f < \frac{1}{2}N$.

Let $\boldsymbol{\mathcal{U}}$ be a set of all unrealised messages in the system. Assuming $\boldsymbol{\mathcal{U}} \neq \emptyset$, let $\boldsymbol{m} \in \boldsymbol{\mathcal{U}}$ be an unrealised message at some time $t' \geqslant t_0$, originated by some operative p_o, such that: $\forall m \in \boldsymbol{\mathcal{U}} : \boldsymbol{m} \xrightarrow{o<j} m \wedge \boldsymbol{m} \neq m$.

Let $\mathcal{P}$ be a set of all operative nodes that have $\boldsymbol{m}$. Because $\boldsymbol{m}$ is unrealised, $\overline{\mathcal{P}}$ cannot be empty and there is at least one operative node, which has not received $\boldsymbol{m}$.

Each node $p_i \in \mathcal{P}$ broadcast $\boldsymbol{m}$ every β seconds. Consequently, whenever node p_i of $\mathcal{P}$ and node p_j of $\overline{\mathcal{P}}$ directly connect, which must occur in accordance with the network liveness property, node p_j receives $\boldsymbol{m}$ and node p_i updates its $K_i(\boldsymbol{m})$ vector so that: $K_i(\boldsymbol{m})[j] = 1$. As a result, p_j will also thereafter broadcast $\boldsymbol{m}$ periodically.

If at least one node of $\mathcal{P}$ remains operative, and since no node can be permanently isolated for any t, eventually $\boldsymbol{m}$ will be received by at least $N - f$ nodes (the minimum number of operative nodes) and some operative node(s) will realise $\boldsymbol{m}$.

Say now that only some operative hosts have realised $\boldsymbol{m}$, and let $\mathcal{P}'$ be a set of all these nodes.

Whenever node p_i of $\mathcal{P}'$ and node p_j of $\overline{\mathcal{P}}'$ directly connect, which must occur in accordance with the network liveness property, node p_j receives realisation vector R_i, in which: $R_i[o] \geqslant \mathtt{C}(\boldsymbol{m})$. Therefore, p_j will realise all messages originated by p_o for which $\mathtt{C}(m) \leqslant R_i[o]$, including $\boldsymbol{m}$ if p_j has it.

Because no node can be permanently isolated for any t, and since the number of operative nodes is finite, all operative nodes that have $\boldsymbol{m}$ will eventually realise it.

Finally, let node p_i initialises a broadcast of message m_b at some time $t_b \geqslant t_0$.

We observe that the number of unrealised messages for which $m \xrightarrow{j<i} m_b$ holds is finite. Because every message $\boldsymbol{m}$ (broadcast by operative node(s)) will be received by at least $N - f$ nodes and realised by all of them, therefore eventually every message m (broadcast by operative nodes) will also be eventually received by at least $N - f$ nodes and realised. Thus, if the originator of m_b remains operative, or any operative node receives m_b, m_b will be received by at least $N - f$ nodes (**BCAST1** *progress* property) and realised by all of them within some bounded time (**BCAST2** *termination* property). □

Finally, similar modifications can be also made as in the case of the PKRM and RDP protocols, since the protocols use only different message transmission controls, albeit exchanged messages during direct connections can be selected in the same manner as in the TaPDP with the use of Lamport's logical clocks. Moreover, in the TaPDP protocol each node broadcasts periodically only one unrealised message, so its performance can be further improved by sending some constant number of additional unrealised messages after the one that is always selected, while preserving time constraints independent of system parameters.

5 Conclusions

In this paper, we have defined exact model of ad hoc systems and its liveness property with the use of the dynamic set function. Based on the observation that already known broadcast protocols for ad hoc networks with the property require that the minimum time of direct connectivity between nodes depend on system parameters, we have introduced improved version of these protocols. The proposed protocol is independent of the total number of messages that can be disseminated by each node concurrently. Consequently, the protocol works correctly even though the minimal time of direct connections allows nodes to exchange (send and response to) only two messages, and it requires also less periodical message transmissions. Finally, its correctness has been proved analytically.

Acknowledgment. The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01–00–008/08.

References

1. Boukerche, A. (ed.): lgorithms and Protocols for Wireless, 1st edn. Mobile Ad Hoc Networks. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons (November 2008)
2. Brzeziński, J., Kalewski, M., Libuda, M.: A short survey of basic algorithmic problems in distributed ad hoc systems. Pro Dialog (Polish Information Processing Society Journal) 21, 29–46 (2006)
3. Brzeziński, J., Kalewski, M., Wawrzyniak, D.: On Time Constraints of Reliable Broadcast Protocols for Ad Hoc Networks with the Liveness Property. In: Wyrzykowski, R. (ed.) PPAM 2011, Part I. LNCS, vol. 7203, pp. 40–49. Springer, Heidelberg (2012)
4. Chandra, R., Ramasubramanian, V., Birman, K.P.: Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. In: Proceedings of the 21st International Conference on Distributed Computing Systems, ICDCS 2001, pp. 275–283. IEEE Computer Society Press (April 2001)
5. Gärtner, F.C.: Fundamentals of fault-tolerant distributed computing in asynchronous environments. ACM Computing Surveys 31(1), 1–26 (1999)
6. Gilbert, S., Lynch, N.A.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News 33(2), 51–59 (2002)
7. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Communications of the ACM 21(7), 558–565 (1978)

8. Liu, S., McDermid, J.A.: Dynamic sets and their application in VDM. In: Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing, SAC 1993, pp. 187–192. ACM Press (February 1993)
9. Luo, J., Eugster, P.T., Hubaux, J.P.: Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2003, pp. 2229–2239. IEEE Computer Society Press (April 2003)
10. Misra, S., Woungang, I., Misra, S.C. (eds.): Guide to Wireless Ad Hoc Networks. Computer Communications and Networks, 1st edn. Springer (February 2009)
11. Pagani, E., Rossi, G.P.: Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks. Mobile Networks and Applications 4(3), 175–192 (1999)
12. Vollset, E.W.: Design and Evaluation of Crash Tolerant Protocols for Mobile Adhoc Networks. Ph.D. thesis, University of Newcastle Upon Tyne (September 2005)
13. Vollset, E.W., Ezhilchelvan, P.D.: Design and performance-study of crash-tolerant protocols for broadcasting and supporting consensus in MANETs. In: Proceedings of the 24th IEEE Symposium on Reliable Distributed Systems, SRDS 2005, pp. 166–178. IEEE Computer Society (October 2005)

Exploiting Asymmetric Links in a Convergecast Routing Protocol for Wireless Sensor Networks

Bilel Romdhani[1], Dominique Barthel[2], and Fabrice Valois[1]

[1] University of Lyon, INSA-Lyon, CITI F-69621, France
FirstName.LastName@insa-lyon.fr
[2] Orange Labs R&D, 38243 Meylan, France
FirstName.LastName@orange.com

Abstract. Most existing routing protocols designed for WSNs assume that links are symmetric, which is in contradiction with what is observed in the field. Indeed, many links in real-world WSNs are asymmetric. Asymmetric links can dramatically decrease the performance of routing algorithms not designed to cope with them. Quite naturally, most existing routing protocol implementations prune the asymmetric links to only use the symmetric ones. In our experience, asymmetric links are a valuable asset to improve network connectivity, capacity and overall performance. We therefore introduce AsymRP (*Asymmetric Convergecast Routing Protocol*), a new routing protocol for collecting data in WSNs. AsymRP, a convergecast routing protocol, assumes 2-hop neighborhood knowledge and uses implicit and explicit acknowledgment. It takes advantage of asymmetric links to increase delivery ratio while lowering hop count and packet replication.

Keywords: Wireless Sensor Networks, Asymmetric links, Convergecast routing protocol.

1 Introduction

We argue that the existence of asymmetric links should not be neglected in WSNs. Many studies have demonstrated the presence of asymmetric links [1] [2]. These asymmetric links are caused by transmission power disparity, interference or radio irregularity [3]. Ignoring their presence degrades the performance of routing algorithms not designed to cope with them [3], while pruning them [4] [5] represents missed opportunities. In this work, we are interested in exploiting the asymmetric links present in real-world WSNs, whatever their origin. [6] [7] have already shown how the use of asymmetric links can reduce path length and decrease end-to-end transmission delay. We improve the use of asymmetric links even further by also reducing the number of replicated packets received by the sink node and by improving the data delivery ratio.

Facing a network containing asymmetric links, a first challenge is to detect them. We present a simple mechanism based on the exchange of neighborhood tables. A second challenge is to make good use of them to forward data messages

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 371–384, 2012.

to the sink node while still sending acknowledgment (ACK) messages back to the source node in order to avoid unnecessary retransmissions and to reduce message replication. To that end, we present a mechanism using a combination of implicit and explicit acknowledgments based on 2-hop neighborhood knowledge. Our proposal is able to deliver messages from source nodes to the destination sink node irrespective of topology and network density. A measure of success is to verify that paths that deliver data messages to the sink do contain asymmetric links.

The remainder of this paper is organized as follows. In Section 2, we present the related work. In Sections 3 and 4, we describe our proposal and evaluate its performance, respectively. Section 5 concludes the paper.

2 Related Work

As previously mentioned, there are two ways of dealing with asymmetric links: to avoid them or to ride them.

Protocols such as [5] [8] are examples of the former behavior. Typically, asymmetric and symmetric links are discriminated by exchanging neighbors lists. Nodes will then only use the symmetric links in the routing phase. COMPOW [5] goes a little further: it understands that nodes are heterogeneous and assumes that link asymmetry is only caused by heterogeneous transmission power. It therefore calculates a common transmission range which will be used by all nodes in the network. This transmission range is calculated to reduce interference, to eliminate asymmetric links and to ensure connectivity between nodes. The drawbacks of COMPOW are that it is centralized and that it is unfit to a changing environment.

By contrast, protocols such as [9] [10] [11] are designed for making use of asymmetric links. TRIF [9], used jointly with RREQ/RREP-based routing protocols, assumes that asymmetric links are caused by the existence of heterogeneous transmission ranges in the network. TRIF assumes that the transmission range is selectable among a few values: each RREQ is sent repeatedly using a decreasing transmission range. The RREQ header contains the value of the transmission range used for sending it. The receiver processes the RREQ if the range mentioned in the header is less than or equal to its own transmission range, otherwise it infers that the link cannot be used backwards and it drops the RREQ. A drawback of this protocol is that it supposes that heterogeneous transmission powers are the only reason for asymmetric links. Therefore, it can not be used when asymmetric links are caused by interferences or radio irregularity. [10] deals with asymmetric links by organizing *Volunteer Relaying.* Each node scans the neighbor lists of its own neighbors to detect asymmetric links between any pair of them. It then volunteers itself to relay the link discovery and maintenance information between such neighbors. Such a mechanism can cause packet replication when several neighbors volunteer themselves. Suppression techniques mitigate the problem, but at the cost of extra complexity.

DEAL [11] exploits asymmetric links at the link layer, based on two different mechanisms. First, DEAL uses a feedback scheme called *Source-Specified Relay* (SSR) to exploit local information at link layer and find the relay nodes for information relaying over the poor direction of asymmetric links. SSR has the same problem as *Volunteer Relaying* has, which is packet replication. Second, DEAL uses a link maintenance scheme called *Dynamic Driven Maintenance* (DDM) which supposes that the asymmetric links are a temporal phenomenon. DDM adopts different strategies in order to use the most efficient links at any given time. DEAL supposes that the network is dense and all the results presented in [11] suppose that nodes have 10 to 50 neighbors.

By contrast, our proposal, AsymRP, addresses the problem of the asymmetric links in a connected WSN without any assumption on the density of the network. With AsymRP, a node receiving a message destined to the sink locally decides whether to participate or not in the forwarding, based on information contained in the received message and on its 2-hop neighborhood knowledge. AsymRP is described in the next section.

3 AsymRP: Asymmetric Convergecast Routing Protocol for WSNs

In this paper, we propose a convergecast routing protocol dedicated to WSN with asymmetric links. AsymRP (*Asymmetric Convergecast Routing Protocol*) make uses of asymmetric links to improve data collection while avoiding redundant messages and reducing the hop counts from sensors to the sink.

3.1 Network Model and Hypothesis

We consider a WSN with many sensor nodes and one static sink node, all deployed at t = 0. We assume that the network includes asymmetric links and this it is connected: there is a path from any node to any other node in the network. No geographic information is available for any network node, yet we suppose that sensor nodes know their position relative to the sink along the bidirectional propagation paths, which we call rank. Sensor nodes closer to the sink node have smaller ranks. The rank is used as a gradient to direct data to the sink. Finally, we assume that each node knows its 2-hop neighborhood. In our performance evaluation, we do take into account the cost of discovering this 2-hop neighborhood.

3.2 AsymRP: The Data Collection Phase

The goal of this phase is to route data messages from sensor nodes to the sink node. In this phase, when a sensor node has data to send to the sink node, it broadcasts this data message to its 1-hop neighborhood. In the header of this data message, the sender node, called *Source*, includes its ID, its rank and its neighborhood table. Each sender node starts a timer, `timeout_relayed`, during which it checks if its message is relayed. If the timer expires and the sender node is not aware that its message was relayed by another node, it broadcasts this message a second time. The way `timeout_relayed` is computed ensures that nodes closest to the sink forward the packet (see section 3.4).

Algorithm 1 Data Collection Phase

```
if Rank(Candidate) < Rank(Source) then
  Candidate triggers a Timeout
  while Timeout not expires do
    if Message relayed by another node then
      Stop the Timeout
      Return
    end if
  end while
  if The link between Candidate and Source node is symmetric then
    Candidate relays the message towards the destination
  else
    Candidate search a Common neighbor with Source node
    if The Common Neighbor exists then
      Candidate sends explicit ACK to the Common neighbor
      Common neighbor forwards the explicit ACK to the Source node
      Candidate relays the message towards the destination
    else
      Candidate search an Intermediate neighbor
      if Intermediate node exists then
        Candidate sends explicit ACK to Intermediate node
        Candidate relays the message towards the destination
        Intermediate node forwards the explicit ACK using source ACK routing.
      end if
    end if
  end if
end if
```

Fig. 1. Data Collection Algorithm

When a broadcast data message from a *Source* node is received, each neighbor applies the algorithm described in figure 1: it first checks if it is closer to the sink node by comparing its rank with that of the *Source* node, in which case we call it a *Candidate* node. A *Candidate* node computes a timer `timeout_to_relay` and enters a contention phase. The objective of this timer is to promote the node closest to the sink node (i.e. having the smallest rank). The way this timer is computed is also discussed in section 3.4.

If a *Candidate* node detects that the message is forwarded by another node, the contention phase is aborted. By contrast, if the timer expires and no other node has forwarded the message, this node checks whether the radio link with the *Source* node is symmetric (by checking its 2-hop neighbor table). If the link is symmetric, this *Candidate* node relays the data message. This will be understood as an implicit ACK by the *Source* node. Conversely, if the *Candidate* node determines that the link from *Source* is asymmetric, then:

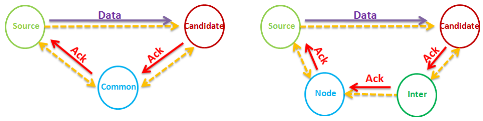
1. **First**, the candidate node tries to find in its neighbor table a *Common* neighbor (i.e a node that can communicate with the *Source* and the *Candidate* nodes as in figure 2(a)). If such a node exists, the *Candidate* node forwards

the data message and sends an explicit ACK to the *Common* neighbor. The latter forwards the ACK to the *Source* node (see figure 2(a)).

2. **Second**, if such *Common* node does not exist, the *Candidate* node tries to find in its neighbor table a node, called *Intermediate*, which satisfies the following two conditions:
   – One of the neighbors detected by the *Source* node can receive the message sent by this *Intermediate* node.
   – The *Intermediate* node has a symmetric link with the *Candidate* node.

   If this *Intermediate* node exists (see figure 2(b)), the *Candidate* forwards the data message and sends an explicit ACK to the *Intermediate* node which will forward it to one of the neighbors detected by the *Source* node, which in turn will forward it to the *Source* node as represented in figure 2(b).

This algorithm is iterated until the message arrives at the sink node. The sink node, when receiving a data message, responds by sending an explicit ACK message.



(a) *Common* Neighbor detection (b) *Intermediate* Neighbor detection

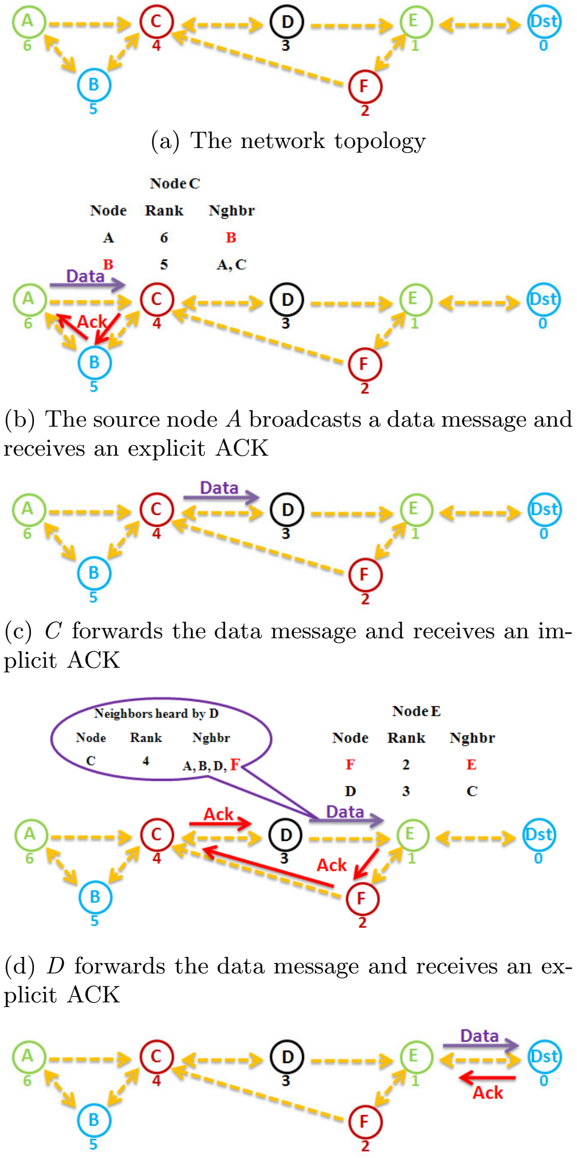
Fig. 2. Data collection and explicit ACK message using *Common* and *Intermediate* neighbors

3.3 Example

Consider the example shown in figure 3. We assume a simple network composed of 6 sensor nodes (A, B, C, D, E and F) and one sink node (Dst). We assume that each node has a rank that determines its relative position to the sink node (Dst). This rank is written below each node in figure 3. We assume that links A-C, D-E and F-C are asymmetric (see figure 3(a)). We assume that nodes have built their neighbor tables, which are not represented because of space limitation; we only show in 3(b) and 3(d) a part of the tables as used by AsymRP. We now walk through the steps of propagating a data message from A to Dst, illustrating different situations along the path.

(a) *A* broadcasts the data message and receives an explicit ACK (fig. 3(b))

The source node A broadcasts the data message to its neighborhood (figure 3(b)), including in the header its ID, its rank and its neighbor table. The message sent by node A is received by nodes B and C. Each of them starts a timer `timeout_to_relay`. The timer triggered at node C expires first since node C has a rank equal to 4, smaller than that of B, which is 5. Then, node C checks



(a) The network topology

Node C

Node	Rank	Nghbr
A	6	B
B	5	A, C

(b) The source node *A* broadcasts a data message and receives an explicit ACK

(c) *C* forwards the data message and receives an implicit ACK

Neighbors heard by D

Node	Rank	Nghbr
C	4	A, B, D, F

Node E

Node	Rank	Nghbr
F	2	E
D	3	C

(d) *D* forwards the data message and receives an explicit ACK

(e) *E* forwards the data message and receives an explicit ACK

Fig. 3. Example of a topology with asymmetric links: *A* sends a data message to *Dst*

whether the radio link between itself and A is symmetric or not. To that end, node C verifies in its neighbor table if it is in the neighbor list of A (figure 3(b)). This is not the case, since A only has B as a neighbor. Therefore, C tries to find in its neighbor table a neighbor common to itself and to source A. It determines that B is a common neighbor. So, it broadcasts the data message towards the sink and sends an explicit ACK to B that in turn forwards it to A (figure 3(b)). By receiving this ACK, B, which is in contention phase with C, stops its timer forwards the ACK message to A and drops the data message received from A.

(b) *C* forwards the data message and detects an implicit ACK (fig. 3(c))
The data message sent by C will be received by D, which is the only candidate to forward this data message (figure 3(c)). The radio link between C and D is symmetric so, after `timeout_to_relay` expires, D broadcasts this message to its neighborhood, which will be understood as an implicit ACK by D.

(c) *D* forwards the data message and receives an explicit ACK (fig. 3(d))
The data message sent by D is received by E. E determines that the radio link between itself and D is asymmetric and that there is no common neighbor between itself and D. Therefore, node E checks if it has an intermediate neighbor which can communicate with a node detected by D: indeed, F, a neighbor of E, can communicate with C which is detected by D (figure 3(d)). Hence, E forwards the data message after `timeout_to_relay` expires, and sends an explicit ACK to F which forwards it to C which in turn sends it to D (figure 3(d)).

(d) *E* forwards the data message and receives an explicit ACK (fig. 3(e))
Finally, the message broadcasted by node E is received by the sink Dst which replies with an explicit ACK (figure 3(e)).

3.4 Timeout Calculation

AsymRP uses two timers: the first one, `timeout_to_relay`, is computed by candidate nodes before forwarding data messages, while the second one, `timeout_relayed`, is calculated by the sender node.

- `Timeout_to_relay`: The `timeout_to_relay` is calculated by the candidate nodes which could relay the data message and which enters into the contention phase. The purpose of this timeout is to introduce priorities between candidate nodes. The node with the highest priority will be the next hop to relay the message toward the sink node. The goal is to favor nodes closer to the destination and to promote the use of asymmetric links. Thus, the timeout calculated will be proportional to the rank of the candidate node (the smaller the rank, the shorter the timeout). In the case where the asymmetric links are caused by the heterogeneity in power transmission range, this timer will also be inversely proportional to the transmission range level of candidate nodes (the higher the transmission range, the shorter the delay before relaying). The goal of this second condition is to promote the use of longest links, in order to reduce the number of hops.

- `Timeout_relayed`: This timer is initiated by sender nodes. It is used to ensure that the data message is relayed by another node towards the sink. This timer should be larger than the upper bound of `Timeout_to_relay` calculated by neighbors of the sender node added to three times the estimated propagation delay of the ACK message. Indeed, the maximum time that a node can wait to hear its message relayed by a direct neighbor is equal to the upper bound of the waiting time the candidate node computes to relay the message. If the message is relayed by a node that the sender node can not hear, the sender node must wait for an explicit ACK which can be sent through up to three hops.

4 Performance Evaluation

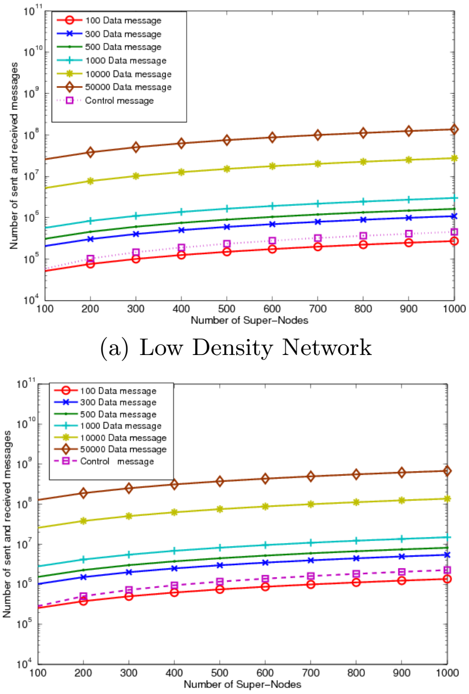
In this section, we begin with a numerical study in which we evaluate the energy consumption of AsymRP and compare it with the energy consumption of TRIF [9]. We then present the results of simulations under realistic assumptions comparing AsymRP and TRIF. To be fair in the comparison between the two protocols, we have added to TRIF an ACK message sent by the sink node when it receives a data message. Without this addition, the nodes in the contention phase would retransmit the same message towards the sink node. Without loss of generality, we consider that asymmetric links are caused by different transmission ranges and we define two types of nodes:

- **Normal-nodes:** sensor nodes having a regular transmission range.
- **Super-nodes:** sensor nodes having other, superior, transmission ranges.

4.1 Numerical Evaluation

This section focuses on the evaluation of the energy consumed by AsymRP and TRIF [9]. Our proposal, AsymRP, requires neighborhood knowledge and there is a tradeoff between the energy required to get this information and the energy saved during the data collection phase. Obviously, in applications where data collection is frequent compared to network topology changes, the cost of neighborhood discovery can be made insignificant compared to the cost of periodic data collection. Due to limited space, we do not reproduce here the calculation of the number of sent and received messages. For further details, we invite the reader to consult the research report [12]. We set the number of nodes in the network to 1000 and the range of super-nodes to 6 times the range of normal-nodes. We examine both a low density network and a high density one, where the normal nodes have an average of 6 and 30 neighbors, respectively. In this section, we start by evaluating the cost of the neighborhood discovery and the data collection phase for AsymRP. Then, we compare the energy consumption of AsymRP and TRIF, both for a high and a low density network. In this section, energy cost is approximated by the number of messages exchanged.

AsymRP, Tradeoff between Neighbor Discovery Cost and Data Collection Cost: Figures 4(a) and 4(b) represent the number of total sent and received messages for a low and a high density network, respectively. We note that, irrespective of the network density, the cost of neighbor discovery becomes negligible (an order of magnitude less) compared to the data collection cost as soon as the number of data messages originated at the network reaches 1/3 of the total number of nodes in the network.

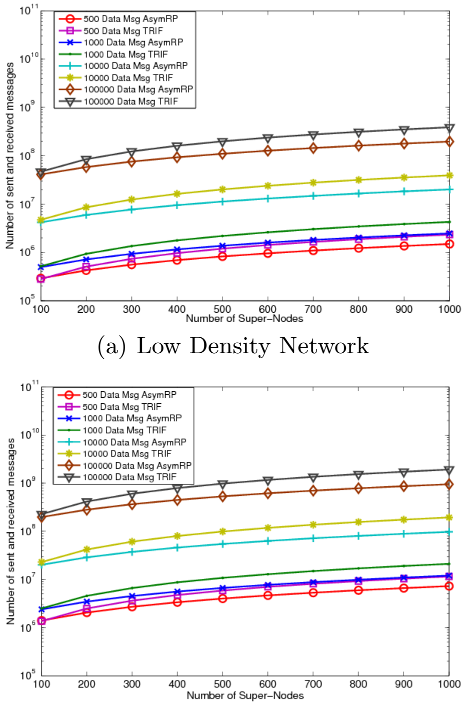


(a) Low Density Network

(b) High Density Network

Fig. 4. AsymRP: Total number of sent and received messages for neighbor discovery and data collection phase

AsymRP and TRIF Energy Consumption Comparison: Here, we compare the amount of messages sent and received with AsymRP (neighbor discovery messages and data messages) and TRIF, in a high and a low density network (Figure 5(a) and figure 5(b), respectively). In both cases, AsymRP uses less messages than TRIF, which goes toward consuming less energy. The gap between the two curves representing the messages exchanged with TRIF and AsymRP increases when the number of data messages generated in the network increases. AsymRP consumes less energy than TRIF because, for each super-node sending one extra data message, AsymRP generates only one transmission and one reception at all its neighbors. Whereas with TRIF, when a super-node sends



(a) Low Density Network

(b) High Density Network

Fig. 5. AsymRP vs. TRIF: Total number of sent and received messages

one extra data message, many transmissions occur (in a number equalling the number of possible transmission ranges of that node) and they generate multiple receptions at the neighbors (see [12] for more details).

4.2 Network Simulation

In this section, we describe the parameters used in our simulations. We then present the main results of our simulations, comparing AsymRP to TRIF [9] on several performance metrics.

Simulation Parameters: We consider a regular grid topology. We select a random number of source nodes, which send periodic data messages to the sink node at staggered instants. The sink node is placed at the center of the network. We consider that asymmetric links are caused by the presence of different transmission ranges in the network. We assume that three ranges are possible: normal-nodes have a regular transmission range while super-nodes have a transmission range equal to either three or six times (balanced mix) the transmission range level of normal-nodes. These super-nodes are uniformely located in the network. The percentage of total super-nodes varies from 10% to 50%.

Table 1. Simulation Parameters

Parameter	Value
Sensor Nodes	120
Node range	1x, 3x and 6x regular range
Number of source nodes	1 .. 50
Number of packet sent	1 packet / minute / source node
Propagation	Two ray ground
MAC Protocol	CSMA/CA-like MAC protocol
Confidence Interval	95%
Simulator	WSNet [13]

Table 1 summarizes the main characteristics of the network.

Percentage of Asymmetric Links used with AsymRP: Figure 6 represents the percentage of asymmetric links used with AsymRP. When the source nodes are far from the sink node, the percentage of asymmetric links used with AsymRP increases. We verify also that an increase in the number of super-nodes yields a higher percentage of asymmetric links exploited in paths with AsymRP.

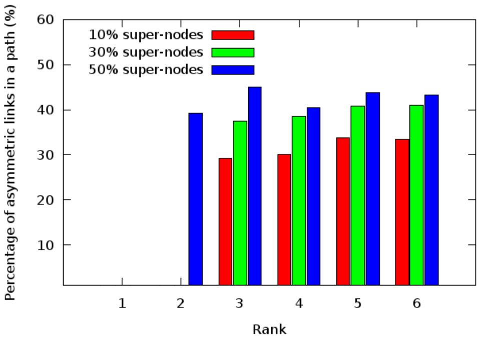


Fig. 6. AsymRP: Percentage of asymmetric links used

Comparison of the Number of Hops Performed: In this section, we evaluate the average hop count that a packet needs to reach the final destination. Figure 7 represents the average hop count using TRIF and AsymRP for each rank of the source nodes. We verify that AsymRP offers a lower hop count when compared to TRIF. This is obviously the benefit of exploiting the asymmetric links. We also verify that, with AsymRP, an increase in the number of super-nodes decreases the number of hops. This is because they offer more opportunities for long range transmission.

Replicated Received Packets and Delivery Ratios: Figure 8(a) represents the amount of replicated data messages received at the sink node for AsymRP, for TRIF when using ACKs sent by the sink node and for TRIF without ACKs, all for 50 source nodes and in three scenarios: 10%, 30% and 50% super-nodes are randomly deployed

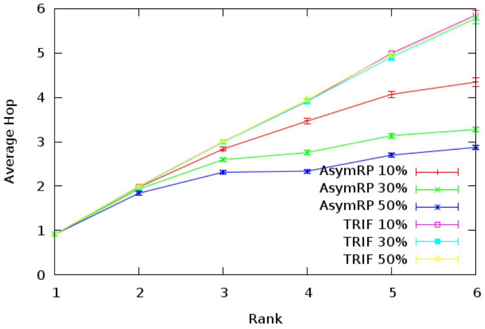
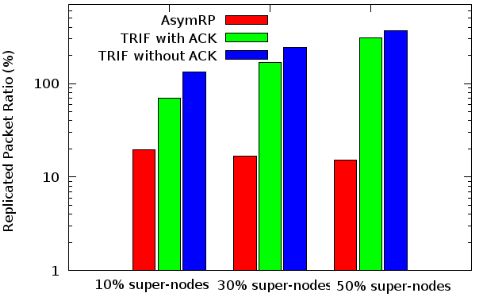
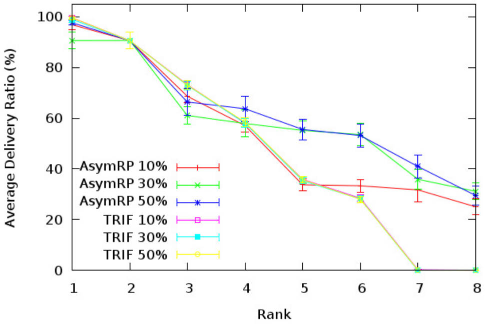


Fig. 7. AsymRP vs TRIF: Number of hops performed



(a) Replicated packet ratio at sink



(b) Delivery Ratio

Fig. 8. AsymRP vs TRIF: Replicated packet and Delivery Ratios

in a grid topology. Figure 8(a) highlights that in all cases, the amount of replicated packets received at the sink with AsymRP is less than the amount of replicated packets with the two variants of TRIF. Figure 8(a) shows that, with the two variants of TRIF, a higher number of super-nodes increases the replication ratio. This is because each super-node iteratively sends the message with decreasing transmission ranges. By contrast, AsymRP has a replication ratio that is very low (around 10%) and independant of the number of super-nodes. This is because, in AsymRP, each source node only sends one message, with its own maximum transmission range.

Figure 8(b) represents the delivery ratio for source nodes indexed by their rank. Obviously, the longer the path, the lower the delivery ratio. Since TRIF does not use asymmetric links, the paths are longer. This explains the decrease in delivery ratio when the source node is far (having a high rank) from the sink node. Hence the delivery ratio for TRIF is equal to 0% for further source nodes (with rank equal to 7 and 8 in figure 8(b)). Moreover, figure 8(b)) shows that when the number of asymmetric links in the network increases (by increasing the number of super-nodes) the delivery ratio of AsymRP increases: we verify that, with AsymRP, the use of asymmetric links does indeed reduce the number of hops.

5 Conclusion

In this paper, we proposed a data collection algorithm for WSNs that include asymmetric links. Simulations highlight that AsymRP benefits from the presence of asymmetric links and meets the requirements of providing a higher delivery ratio, a lower hop count and a lower packet replication, compared to TRIF. We studied and evaluated the energy consumption of the neighborhood discovery and data collection phases. We noted that the cost of neighbor discovery becomes negligible compared to that of data collection as soon as the number of data messages originated at the network reaches 1/3 of the total number of nodes. Then, we compared the energy consumption of AsymRP and TRIF by computing the total number of sent and received messages. We showed that, irrespective of the network density, AsymRP consumes less energy than TRIF. We are now working on evaluating the overhearing cost induced by the broadcast transmissions, on simulating other topologies and also on using other metrics, such as the amount of remaining energy in each node, to compute the timeouts needed by AsymRP. The use of such metrics will avoid over-exploiting some nodes during the collection phase. This will even out the energy consumption and therefore increase the lifetime of the network even further. For the consideration of the dynamics of asymmetric links in the network, we are studying the use of adaptive periodic neighborhood discovery phases based on a self regulating algorithm such as *Trickle* [14].

References

1. Heurtefeux, K., Valois, F.: Is rssi a good choice for localization in wireless sensor network? In: The 26th IEEE International Conference on Advanced Information Networking and Applications (2012)
2. Kim, K.H., Shin, K.G.: On accurate measurement of link quality in multi-hop wireless mesh networks. In: The 12th Annual International Conference on Mobile Computing and Networking (2006)

3. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.A.: Impact of radio irregularity on wireless sensor networks. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (2004)
4. Kawadia, V., Kumar, P.: Power control and clustering in ad hoc networks. In: The 22nd Annual Joint Conference of the IEEE Computer and Communications (2003)
5. 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 (2002)
6. Nesargi, S., Prakash, R.: A tunneling approach to routing with unidirectional links in mobile ad-hoc networks. In: The 9th International Conference on Computer Communications and Networks (2000)
7. Shah, V., Krishnamurthy, S.: Handling asymmetry in power heterogeneous ad hoc networks: A cross layer approach. In: Proceedings 25th IEEE International Conference on Distributed Computing Systems (2005)
8. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.A.: Models and solutions for radio irregularity in wireless sensor networks. ACM Trans. Sen. Netw. 2, 221–262 (2006)
9. Le, T., Sinha, P., Xuan, D.: Turning heterogeneity into an advantage in wireless ad-hoc network routing. Ad Hoc Netw. 8(1), 108–118 (2010)
10. Sang, L., Arora, A., Zhang, H.: On exploiting asymmetric wireless links via one-way estimation. In: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (2007)
11. Chen, B.B., Hao, S., Zhang, M., Chan, M.C., Ananda, A.L.: Deal: discover and exploit asymmetric links in dense wireless sensor networks. In: The 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (2009)
12. Romdhani, B., Barthel, D., Valois, F.: Exploiting Asymmetric Links in a Convergecast Routing Protocol for WSNs. Rapport de recherche RR-7586, INRIA (2011)
13. Hamida, E.B., Chelius, G., Gorce, J.M.: Scalable versus accurate physical layer modeling in wireless network simulations. In: Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation, Washington, DC, USA (2008)
14. Levis, P., Patel, N., Culler, D., Shenker, S.: Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation, NSDI 2004 (2004)

Energy Efficient k-Anycast Routing in Multi-sink Wireless Networks with Guaranteed Delivery*

Nathalie Mitton[1], David Simplot-Ryl[1], Marie-Emilie Voge[2], and Lei Zhang[1]

[1] INRIA Lille, Nord Europe
firstname.lastname@inria.fr
[2] Univ. Lille Nord de France
Marie-Emilie.Voge@lifl.fr

Abstract. In k-anycasting, a sensor wants to report event information to any k sinks in the network. This is important to gain in reliability and efficiency in wireless sensor and actor networks. In this paper, we describe KanGuRou, the first position-based energy efficient k-anycast routing which guarantees the packet delivery to k sinks as long as the connected component that contains s also contains sufficient number of sinks. A node s running KanGuRou first computes a tree including k sinks among the M available ones, with weight as low as possible. If this tree has $m \geq 1$ edges originated at node s, s duplicates the message m times and runs m times KanGuRou over a subset of defined sinks. Simulation results show that KanGuRou allows up to 62% of energy saving compared to plain anycasting.

1 Introduction

Wireless sensor networks have been receiving a lot of attention in recent years due to their potential applications in various areas such as monitoring and data gathering. Sensor measurements from the environment may be sent to a base station (sink) in order to be analyzed. Other sensors may serve as routers on a path established to deliver the report. In large sensor networks, there may exist a bottleneck (around sink) if a single sink collects reports from all sensors. Scenarios with multiple sinks are then being considered, where each sensor reports to at least one sink, usually the nearest one. In wireless multi-sink sensor networks, anycasting is performed when any of sinks may receive the report from sensors, and meet application demands. However, the cost of anycasting may depend on the distance between the receiving sinks and the reporting sensor. It is therefore desirable that selected algorithm reaches one of sinks close to the event. For reliability, load-balancing and security purposes, it is then useful to ensure that at least k sinks receive the messages (where the overall number of sinks is greater than k) whatever the k sinks. To date, there is no so much

* This work was partially supported by CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante and the ANR BinThatThinks project.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 385–398, 2012.

work in the literature. Most of works are adaptation of wired solutions [9] and are thus centralized. Others use flooding [10] and not suitable for high dynamic networks (such as wireless sensor networks). A distributed k-anycast routing protocol based on mobile agents is proposed in [11] but requires a regular update of routing tables which also have to maintain paths towards every sink.

In this paper, we introduce KanGuRou (k-ANycast GUaranteed delivery ROUting protocol), a **position-based**, **energy-efficient** localized k-anycast routing protocol that **guarantees delivery** (therefore loop-less), is **memoryless**, and **scalable**. Unlike [11], it does not maintain any routing table and does not need to add any information neither on nodes nor in the message, which makes it scalable regardless of the number of sinks/nodes. It inspires from energy-efficient anycast EEGDA algorithm [5] and the splitting techniques of MSTEAM [4], proposing a new tree construction to ensure reaching k sinks. At each step, the current node s computes a spanning tree over k sinks with minimal cost. A message replication occurs when the tree spanning s and the set of sinks has multiple edges (later called branches) originated at the current node. Since there may be more sinks than the k to be reached, all of them are not spanned by the tree. The number of sinks k' spanned by each branch determines the number of sinks to be reached by each message. All sinks (not only the ones spanned by the tree) are distributed over every edge. The next hop is chosen in a cost-over-progress (COP) fashion, *i.e.* to the neighbor v which minimizes the ratio between the cost to reach v and the progress provided by v. The cost from s to v is the cost of the energy-weighted shortest path (ESP). The progress is computed as the difference between the weight of the trees computed by s and v resp. If s has no neighbor with positive progress, node s applies a EEGDA-face like routing, which is a face-based recovery mode. We prove that KanGuRou guarantees delivery to exactly k sinks. We present two variants which differ in the way the tree is computed. KanGuRou is evaluated through extensive simulations and results show that both variants of KanGuRou are energy efficient. Results show that KanGuRou allows up to 62% of energy saving and that every variant performs better regarding the percentage of sinks to reach.

The remaining of the paper is organized as follows. Section 2 gives an overview of the literature about k-anycasting and present works on which KanGuRou is based. Section 3 introduces our notations. Section 4 presents KanGuRou. Section 5 presents simulation results. Finally Section 6 concludes the paper.

2 Related Works

k-Anycast was first introduced in [9] for wired networks. Propositions in wireless networks firstly appeared in [8] proposing centralized solutions and thus does not really meet wireless networks requirements. [10,2] presents a reactive approach (flooding) and two advanced proactive approaches in which sinks have previously been gathered into components of at most k members and these components are then reached during the routing. To the best of our knowledge, the only distributed k-Anycast routing protocol is based on mobile agents and proposed in [11]. The protocol forms multiple components and each component has at least k members. Each component can be treated as a virtual server, so k-anycast service is distributed to each component. In this protocol, each routing node only needs to exchange routing information with its neighbors, so the protocol saves much communication cost and adapts to high dynamic networks. Nevertheless, although a first step toward, this algorithm needs to maintain routing tables at each node with as many entries as sinks and is not scalable.

In this paper, we introduce KanGuRou which is a position-based k-anycasting protocol. KanGuRou is an extension of the anycasting protocol proposed in [5] to the k anycasting. In [5], authors describe EEGPA the first localized anycasting algorithms that guarantee delivery for connected multi-sink sensor networks based on a GFG approach. Let $S(x)$ be the closest actor/sink to sensor x, and $|xS(x)|$ be distance between them. In greedy phase, a node s forwards the packet to its neighbor v that minimizes the ratio of cost of sending packet to v through an ESP over the reduction in distance $(|sS(s)| - |vS(v)|)$ to the closest sink. If none of neighbors reduces that distance then recovery mode is invoked. It is done by face traversal where edges are replaced by paths optimizing given cost.

KanGuRou also inspires from the multicast routing MSTEAM proposed in [4]. MSTEAM is a localized geographic multicast scheme based on the construction of local minimum spanning trees (MSTs), that requires information only on 1-hop neighbors. A message replication occurs when the MST spanning the current node and the set of destinations has multiple edges originated at the current node. Destinations spanned by these edges are grouped together, and for each of these subsets the best neighbor is selected as the next hop. MSTEAM has been proved to be loop-free and to achieve delivery of the multicast message as long as a path to the destinations exists.

3 Model and Notations

Network. We model the network as a graph $G = (V, E)$ where V is the set of sensor nodes and $uv \in E$ iff there exists a wireless link between u and $v \in V$. We suppose that nodes are equipped with a location service hardware such a GPS and are able to tune their range between 0 and R. We note $|uv|$ the Euclidean distance between nodes u and v. We note $N(u)$ the set of physical neighbors of node u, *i.e.* the set of nodes in communication range of node u ($N(u) = \{v \,|\, uv \in E\}$) and $V(G)$ the set V of vertices in G. $S = \{s_i\}_{i=0,1,..M}$ is the set of sinks, with M the number of sinks. Every node is aware of every sink and of its position. We note as $CT_S(s)$ the closest node in S to node s ($CT_S(s) = \{v \,|\, |sv| = \min_{w \in S} |sw|\}$). For a graph $G = (V, E)$ and a set $A \subseteq V$, we denote by $G|_A$ the subgraph of G which contains only nodes of A: $G|_A = (A, E \cap A^2)$.

Tree. Let $T = (V', E')$ be a tree and $a \in V'$ a vertex of T. $\mathrm{st}(T, a)$ is the subtree of T with root a. T is an MST if its weight noted $||T||$ is minimal. The weight of the tree denotes the sum of the weight over all tree edges ($||T|| = \sum_{uv \in E'} |uv|$). In an Euclidean MST, the weight of an edge is equal to its Euclidean length.

A tree $T = (V', E') \subset G$ is a k-MST if $|V'| = k$ and that $||T||$ is the tree with minimum weight over all trees of k vertices from G.

Energy. We assume that every node is able to adapt its transmission range. We use the energy model defined in [7], *i.e.* the energy spent to send a message from nodes u to v is such that $cost(|uv|) = |uv|^\alpha + c$ if $|uv| \neq 0$. where c is signal processing overhead; α is a real constant (> 1) for signal attenuation. From this energy cost, we introduce the cost of the energy-weighted shortest path ($cost_{ESP}(s, d, t)$) from nodes s to d when aiming at target t. We compute the energy-weighted shortest path (ESP) only over nodes that are in the forwarding direction of the final target to avoid either creating routing loops or embedding the path in the message. Therefore, the shortest path computed from node s to node d is relative to the final target t. Let $x_0x_1...x_ix_{i+1}..x_n$, be the node IDs on the ESP from $s = x_0$ to $d = x_n$. We define the ESP cost as

$$cost_{ESP}(s, d, t) = \sum_{i=0}^{n-1} cost(|x_ix_{i+1}|) \tag{1}$$

4 Contribution

4.1 General Idea

In this section, we present the main idea of KanGuRou which goal is to reach any k sinks among all available sinks S. Nevertheless, given a source node s, the k closest sinks to s in Euclidean distance are not necessarily the k closest sinks in number of hops. Therefore, the routing messages in KanGuRou may change target sinks along the routing path. For instance, on Fig. 1, 5 closest sinks of s are S_1, S_2, S_5, S_6 and S_7. But S_1 is not reachable directly and the path to S_1 will meet S_4 which may be reached instead. In addition, the source cannot determine the k sinks in advance and send k messages, one toward each sink because *(i)* several messages may follow the same path by sections which is useless and costly and *(ii)* since targets may change along the path, this cannot ensure that several messages will not reach the same sink.

KanGuRou (Algo. 1) proceeds as follows. Fig. 1 illustrates it.

(1). Node s holding the message first checks whether it is a sink. If so, it removes itself from the set of available sinks and decrements the number of sinks k to reach. If $k = 0$, the algorithm stops. (Line 2).

(2). Node s computes a tree $T(s)$ by running Algo. 3 (k-MST(s,S,k)) or Algo. 4 (k-Prim(s,S,k))) detailed later in Section 4.4, depending of the variant of KanGuRou (Line 7). $T(s)$ contains node s and exactly k sinks of S. If there are several edges/branches originated at s, a message duplication occurs. On Fig. 1, $T(s)$ appear in red and contains sinks S_1, S_3, S_5, S_6 and S_7. There are two branches originated at node s: one toward S_1 and one toward S_5.

(3). s distributes the remaining sinks (Line 8), *i.e.* sinks that are not in $T(s)$ (Sinks S_2, S_4 and S_8 on Fig. 1) over every branch. Thus, for every successor a of

s in $T(s)$ ($a \in \text{succ}_{T(s)}$), a subset $S_a \subset S$ of the sinks is assigned to a as detailed in Section 4.5. On Fig. 1, branch of S_1 is assigned with Sinks S_1, S_3 and S_4 while Sinks S_2, S_5, S_6, S_7 and S_8 are associated to branch of S_5.

(4). At this step, node s knows: *(i)* its successors $a \in \text{succ}_{T(s)}$ in $T(s)$ (Sinks S_1 and S_5 on Fig. 1), *(ii)* the number of sinks k_a to reach per successor a, *i.e.* the number of sinks in the subtree of a $st(T, a)$ (2 in branch of S_1 and 3 in branch of S_5 on Fig. 1), *(iii)* the set of available sinks to reach per branch, *i.e.* S_a defined at the previous step. Node s then sends as many packets as the number of its successors in $T(s)$. (Loop line 9) Thus, for each branch of $T(s)$, *i.e.* $\forall a \in \text{succ}_{T(s)}$, s selects a next hop based on a Greedy-Face-Greedy approach as follows. For every a, s computes the weight of the k_a-MST for each of its neighbors $u \in N(s)$ over S_a targets $||\text{k-MST}(u, S_a, k_a)||$. On Fig. 1, s will compute 3-MST over Sinks S_2, S_5, S_6, S_7 and S_8 to find the next hop for branch S_5 and 2-MST over Sinks S_1, S_3 and S_4 for branch S_1. If there exists no neighbor u for which the weight of tree over S_a $||k\text{-MST}(u, S_a, k_a)||$ is smaller than $||sT(T, a)|| + |sa|$ (weight of the branch of $T(s)$ dedicated to a), node s switches to recovery mode (line 16) till reaching a node with positive progress towards a. If so, next hop v for branch toward a is determined through the greedy mode in a COP fashion (Line 18). Message is sent to node v with parameters k_a and S_a which will run KanGuRou again (Line 19) and so on till k_a sinks have been reached in this branch. As shown in [5], this ensures the packet delivery as soon as the network is connected.

Algorithm 1. KanGuRou(s, k, S) – Run at node s to reach k targets in S.

```
 1: if $s \in S$ then
 2:    $k \leftarrow k - 1$; $S \leftarrow S \setminus \{s\}$
 3:    if $k = 0$ then
 4:       exit {All sinks of this branch have been reached}
 5:    end if
 6: end if
 7: $T(s) \leftarrow \text{k-MST}(s, S, k)$ or $\text{k-Prim}(s, S, k)$ {$k$-MST of $S \cup \{s\}$ rooted in $s$}
 8: $T'(s) \leftarrow \text{AllocateMST}(s, S, T(s))$ {Allocate remaining targets to $T(s)$}
 9: for all $a \in \text{succ}_{T(s)}(s)$ do
10:    $S_a \leftarrow V(\text{st}(T', a))$ {Nodes in sub-tree of $T'$ rooted in $a$}
11:    $k_a \leftarrow |T \cap S_a|$ {Number of targets to be reached in $S_a$.}
12:    $v \leftarrow CT_{S_a}(s)$
13:    $W \leftarrow ||sT(T, a)|| + |sa|$
14:    $A \leftarrow \{v \in N(s) \mid ||\text{k-MST}(v, S_a, k_a)|| < W\}$
15:    if $A = \emptyset$ then
16:       RECOVERY$(s, k_a, S_a, W)$
17:    else
18:       $v \leftarrow u \in A$ which minimizes $\frac{\text{cost}_{ESP}(s,u,a)}{W - ||\text{k-MST}(u, S_a, k_a)||}$
19:       KanGuRou$(v, k_a, S_a)$
20:    end if
21: end for
```

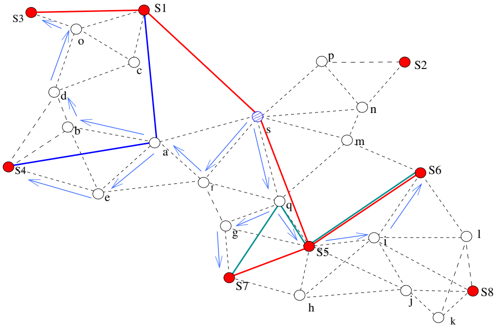


Fig. 1. Sinks appear in red. Red links represent the 5-MST rooted in s, blue links the 2-MST rooted in a over S_1, S_3 and S_4, green links the 3-MST rooted in q over S_2, S_5, S_6, S_7 and S_8. Arrows show the message path.

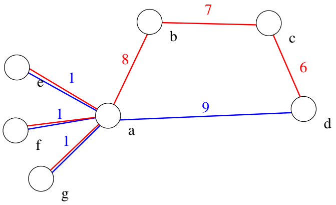


Fig. 2. Illustration of MST and k-MST for $k = 4$. If root is node d, the optimal 4-MST (in blue) includes edges da, ae, af, ag while edge ad will not be included in the MST (in red). So, k-MST is not always included in the MST.

To sum up, let assume that node s on Fig. 1 runs KanGuRou toward $k = 5$ sinks. First, s computes a 5-MST, $T(s)$ (red tree). $T(s)$ has two branches, so s duplicates the message. First message is sent toward branch of S_1 and has to reach 2 sinks among S_1, S_3 and S_4. s computes the COP and selects node a. To reach node a, message is sent to node f since path sfa is less energy consuming than following the direct edge sa. Node a runs KanGuRou and its tree has two branches. So node a duplicates again the message. First copy has to reach one sink among S_1 and S_3 while second copy has to reach S_4. S_4 is reached via path aeS_4 in a greedy way while other copy is sent along path $bdoS_3$. Second message sent by node s has to reach 3 sinks among S_2, S_5, S_6, S_7 and S_8. Greedy algorithm chooses node q. Tree computed on node q has 2 branches originated at q, so q duplicates the message. First copy is sent to node g which forwards it to Sink S_7. Second copy is sent to S_5. S_5 is a sink but the message still has to reach another sink so S_5 forwards it to its neighbor i which directly forwards the message to S_6. At last, 5 sinks have been reached: S_3, S_4, S_5, S_6 and S_7.

4.2 The Greedy Mode

Greedy mode is similar to the one used in [5]. When node s runs greedy algorithm toward Sink a, it computes the subtree $sT(T(s), a)$ of $T(s)$ rooted in a. The weight W of the subtree issued from s toward a is thus the weight of $||sT(T(s), a)||$ plus the weight of the edge sa to reach it: $W = ||sT(T(s), a)|| + |sa|$. Then, to select the next hop, node s performs a COP approach in which *(i)* the cost considered is the cost of the energy weighted shorted path (Eq. 1) from node u to its neighbor v,*(ii)* the progress is the reduction of the weight of trees $W - ||\text{k-MST}(u, S_a, k_a)||$. Only neighbors providing a positive progress are considered. If no such node exists, the greedy approach fails and s switches

Algorithm 2. RECOVERY(u,k,S,W) - Run at node u.

```
 1: $(V', E') \leftarrow$CDS$(V, E) \cup S \cup \{u\}$ {Extract a CDS graph from $G$}
 2: $(V', E'') \leftarrow$GG$(V', E')$ {Build the Gabriel Graph of $G'$}
 3: $u' \leftarrow u$, $T \leftarrow$ k-MST$(u', S, k)$
 4: while $||k - MST(v, k, S|| > W$ do
 5:    $v \leftarrow$ FACE$(u', T)$ {Compute the next node on the proper face}
 6:    while $u' \neq v$ do
 7:       $u' \leftarrow$ESP$(u', v, CT_T(u'))$ {Compute the ESP from $u'$ to $v$}
 8:    end while
 9: end while
10: KanGuRou$(v, k, S)$
```

to recovery mode. If there exist neighbors u such that $W > ||\text{k-MST}(u, S_a, k_a)||$, node u which minimizes $\frac{\text{cost}_{ESP}(s,v,a)}{W - ||\text{k-MST}(u,S_a,k_a)||}$ is selected. Note that when computing k-MST(u, S_a, k_a), all potential sinks are considered, not only the ones in $sT(T(s), a)$. For instance, on Fig. 1, 2-MST computed by node a (blue tree) over S_1, S_3 and S_4 includes S_1 and S_4 (while the one rooted in s includes S_1 and S_3).

4.3 The Recovery Mode

Recovery mode is detailed in Algo. 2. A node u enters the recovery mode while trying to reach k targets among the sinks in S if it has no neighbor which k-MST has a smaller weight than its own weight W toward the considered branch. u runs RECOVERY till reaching a sink or a node v for which $||\text{k-MST}(v, k, S)||$ is smaller than W (Line 4 in Algo. 2)[1].

To determine what neighbor to reach, it applies an EtE-like Face routing [3]. EtE-like Face routing differs from the traditional Face [1] routing in the way that it does not run over the planar of the whole graph but on the planar of a connected dominated set (CDS) graph only (Lines 1-2). This allows considering longer edges. Face algorithm is applied to determine next hop v to reach over the faces on the CDS (Line 5). v is then reached by following an ESP (Line 7).

4.4 Computing the k-MST

Note that computing an exact k-MST is NP-complete. Also note that a k-MST is not necessarily included in the MST as example plotted on Fig. 2 shows. Thus, KanGuRou proposes to use two different tree constructions, both of them being an approximation of the k-MST algorithm. As we will see later, the choice of the variant used in the tree construction will depend on the number of sinks M available in the network and the number k of sinks that need to receive the information. It is important to highlight that this tree is computed on the complete graph of sinks $\varsigma = (S, E_\varsigma)$ with $E_\varsigma = \{uv \,|\, u, v \in S^2\}$. This is independent from the underlying topology.

[1] Unlike in anycasting, recovery in k-anycasting may reach a sink since the distance considered is not between a node and the closest sink but to the closest k sinks.

Algorithm 3. k-MST(u, S, k) – Return a k-MST of $S \cup \{u\}$ rooted in u.

```
 1: $T \leftarrow (\{u\}, \emptyset)$ {initialize the tree with root $u$}
 2: $A \leftarrow S$ {set of nodes to be considered.}
 3: while $k > 0$ do
 4:    for all $v \in A$ do
 5:       $w \leftarrow x \in T$ which minimizes $|xv|$
 6:       $P(v, 1) \leftarrow w$ {Path from $v$ to $T$ in 1 hop with minimum cost.}
 7:       $l(v, 1) \leftarrow |vw|$ {Weight of the path from $v$ to $T$ in 1 hop with minimum cost.}
 8:    end for
 9:    for $i = 2$ to $k$ do
10:       for all $v \in A$ do
11:          $y \leftarrow x \in T$ which minimizes $|vx|$
12:          $\forall w \in A$ $z \leftarrow x \in T$ which minimizes $|wx|$
13:          Select $w \in A$ such that $|wz| < |vy|$ which minimizes $(l(w, i-1) + |vw|)/i$
14:          $p(v, i) \leftarrow p(w, i-1).w$ {Path from $v$ to $T$ in $i$ hops with minimum cost.}
15:          $l(v, i) \leftarrow l(w, i-1) + |vw|${Weight of $p(v, i)$.}
16:       end for
17:    end for
18:    select $v \in A$ and $j \in [1 \dots k]$ which minimizes $l(v, l)/j$
19:    while $p(v, j) \neq \emptyset$ do
20:       $(w, x) \leftarrow$ first edge in $p(v, j)$ {$w$ is supposed to be in $T$ while $x$ is not in $T$}
21:       $T \leftarrow T \cup (\{x\}, \{(w, x)\})$; $A \leftarrow A \setminus \{x\}$; $k \leftarrow k - 1$
22:       $p(v, j) \leftarrow p(v, j) \setminus \{(w, x)\}$
23:    end while
24: end while
25: Return T.
```

First Variant: The first variant (later called KanGuRou) applies Algo. 3 and builds a tree with exactly $k+1$ vertices (k sinks and the source) in an iterative way. It starts with a tree which only contains the root (Line 1), node s on Fig. 1. It then has to choose exactly k sinks in S to add in T. To do so, at each step, it computes the shortest path from any vertex to the tree in exactly i hops, for all i from 1 to $k-i$ for all vertices. On Fig. 1, for $i = 1$, s computes the distance from itself to every sink. For $i = 2$, s considers 2-hop paths from itself to every sink and keeps the shorter one as sS_1S_3 to reach S_3. To reduce the complexity of computing a path from a node u to T, it only considers nodes closer than u to T. On Fig. 1, node s will not compute any 2-hop path from s to S_2 since S_2 is the closest sink. Weight of every path is then normalized by the progress it provides, *i.e.* the number of sinks on the path (Line 18) and the path with the lowest weight is then added to the tree. And so on till the final tree includes k sinks. In this way, note that S_2 is not included in path since step 1, path sS_5S_7 (weight 2) is chosen ($\frac{|sS_5|+|S_5S_7|}{2}$ is smaller than all other path ratios as $\frac{|sS_1|+|S_1S_3|}{2}$ or $\frac{sS_2}{1}$). Then at step 2, path sS_1S_3 is added ($\frac{|sS_1|+|S_1S_3|}{2} < \frac{|sS_1|+|S_1S_3|+|S_3S_6|}{3}$, etc) and at last, path S_5S_6 is added.

Algorithm 4. k-Prim(u, S, k) – Return a k-MST of $S \cup \{u\}$ rooted in u.

```
T ← ({u}, ∅) {initialize the tree with root u}
A ← S {set of nodes to be considered.}
while k > 0 do
    w ← x ∈ A which minimizes |xCT_T(x)|
    T ← T ∪ ({w}, {(w, CT_T(w))})
    A ← A \ {w}; k ← k − 1
end while
Return T.
```

Second Variant: Original Prim algorithm [6] consists in adding iteratively to the current tree (initialized with the root node) the edge with minimum weight which has exactly one extremity vertex in the tree, and so on till every vertex has been added to the tree. KanGuRou-kPrim (Algo. 4) performs similarly but stops when the tree includes and exactly k sinks.

To illustrate the difference between both variants, let us consider Fig. 2 and assume a tree construction rooted in node d with $k = 4$. Algo. 4 adds iteratively the edge (and corresponding nodes) with the lowest weight, *i.e.* nodes c, b, a and e (in the order). Resulting tree has a weight of 22. Algo. 3 does not consider edges one by one but multi-hop paths. It thus adds nodes a and e at once ($\frac{|da|+|ae|}{2}$ is the best ratio), then nodes f and g. Resulting tree has a weight of 12.

4.5 Distributing Sinks over Branches

Once the k-tree rooted in current node has been computed, the set of sinks has to be distributed over each branch. The number of sinks to be reached by branch is given by the number of sinks actually part of the branch. If s is the node in charge of the message, it computes its k-MST $T(s)$. If k_a is the number of sinks to be reached in the branch of $T(s)$ rooted in a, we have $\sum_{a \in \text{succ}_{T(s)}} k_a = k$. The set of potential sinks to reach S_a is sent with the message over each branch a. S_a includes the k_a sinks included in the tree but also part of 'free' ones. S_a sinks have to be selected carefully in order to ensure that exactly k sinks will receive the message. They are such that: *(i)* $\bigcup_{a \in \text{succ}_{T(s)}} S_a = S$ since every sink is candidate and *(ii)* $S_a \cup S_b = \emptyset \, \forall \, a, b \in \text{succ}_{T(s)}$ in order to avoid that a message sent on 2 different branches reaches the same sink in which case, the overall number of sinks receiving the message will be less than k.

In KanGuRou, each sink is assigned to the closest branch regardless of the size of the branches. However, we are aware that this solution is not necessarily the most adequate one since most of remaining sinks may be assigned to the same branch which might be the smallest one. Alternative solutions might be:

- Sinks may be distributed evenly between both branches, based on distance.
- Sinks may be distributed proportionally to the number of sinks to reach per branch.

However, setting in advance the number of sinks to assign to each branch will lead to some other issues. Indeed, issue will appear when sinks are at equal distance of several branches and when a sink p is closer to Branch A, but that Branch A has already been assigned enough sinks, all closer than p. We leave to further work a deeper study on this point.

4.6 Packet Delivery to Exactly *k* Sinks Guaranteed

We show that KanGuRou delivers a message to exactly k sinks as long as the underlying network is connected. Because of page restriction, we only give here the sketch of the proof[2].

Theorem 1. *KanGuRou guarantees the packet delivery to exactly k sinks as long as the network is connected and that the number of sinks in the connected component including s is greater or equal to k.*

Proof. We apply a mathematical induction.

Initial Step. *Theorem 1 is true for $k = 1$.*
When $k = 1$, the 1-MST computed by s running KanGuRou comes to finding $CT_S(s)$, *i.e.* the closest sink to s. KanGuRou comes to EEGDA [5], been proven to guarantee packet delivery as long as the underlying network is connected.

Induction Step. *Assuming that Theorem 1 is true for $k = i - 1$, $1 < i$, we have to prove that Theorem 1 is true for $k = i$.*

When node s runs KanGuRou, it may either duplicate and forward the message or just forward it once. When s splits the message, s runs several times KanGuRou for $k < i$. If s forwards, itt forwards until finding a sink that will then run KanGuRou for $k = i - 1$ or to a node that split the message and then runs several time KanGuRou for $k < i$, for which Theorem 1 is assumed to be true.

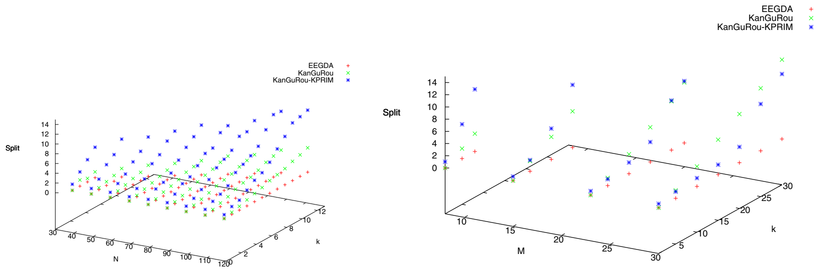
5 Simulation Results

In this section, we evaluate the performances of KanGuRou under the WSNet[3] simulator with an IEEE 802.15.4 MAC layer. As there is no comparable algorithm in the literature since KanGuRou is the first position-based algorithm from the literature, we compare the two variants KanGuRou and KanGuRou-kPrim to running k times the plain EEGDA anycast routing protocol [5] to measure the gain provided by KanGuRou. We deploy N nodes (from 35 to 115) at random in a square of $100\text{m} \times 100\text{m}$, every node can adapt its range between 0 and 30m.

Fig. 3 shows the number of times the message is split/duplicated for each algorithm. Obviously, the number of splits performed by EEGDA is equal to 1 whatever the parameters since EEGDA performs independent anycast routings.

[2] Complete proof is available at researchers.lille.inria.fr/~mitton/kangourou.html

[3] WSNet: http://wsnet.gforge.inria.fr/



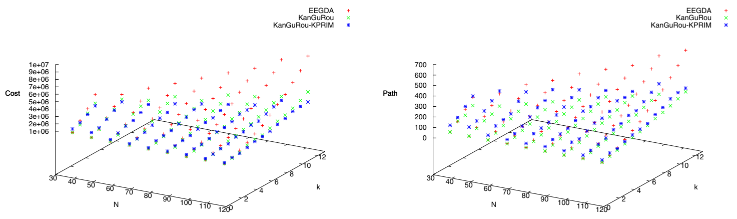
(a) $M = 10\% \times N$

(b) $N = 75$

Fig. 3. Number of splits for each algorithm. M = number of sinks.

For both versions of KanGuRou, it is worth noting that when k increases for a given number of available sinks M and of nodes N, the number of splits also increases. This is expected since algorithms need to reach more sinks and respective trees are bigger and thus the message is more likely to be duplicated to reach sinks. Also, for a fixed k, the number of splits increases when the number of nodes (and thus of available sinks) increases. This is due to the fact that more choices are given to the algorithm and thus more ramifications appear (Fig. 3(a)). We can also note (Fig. 3(b)) that the number of duplications is not really impacted by the overall number of available sinks M in the network (number of splits for a given k). At last, we can observe that the number of duplications increases when M increases (in proportion of N) more quickly for KanGuRou than for KanGuRou-kPrim. Yet, for a low value of M, KanGuRou-kPrim produces more duplications than KanGuRou while for high values of M, KanGuRou duplicates more often messages.

First, the number of sinks M is set to be 10% of the total deployed nodes N. Fig.4 shows the energy consumption (computed based on Eq. 1) and the path length in terms of N and k (k varies from 1 to M). Note that for $k = 1$, results are the same for all three algorithms since KanGuRou comes to EEGDA. Simulation results show clearly that KanGuRou, KanGuRou-kPrim result in significant gains on the energy consumption (up to 62.51% (44.33% in average) and up to 74.22% (53.84% in average) respectively) and path length (up to 62.17% (49.07% in average) and up to 56.61% (21.90% in average) respectively) compared to the traditional algorithm EEGDA. An amelioration was indeed expected since in KanGuRou, part of the path is mutualized. Nevertheless, the gain remains important. Globally, we can see that behavior of every algorithm is similar whatever the parameters. Regarding the energy consumption, results show that KanGuRou-kPrim consumes less energy compared to KanGuRou when k is important, and KanGuRou performs better for low k. This is due to the fact that when k increases (for a constant M), k-Prim algorithm gets closer and closer to the optimal k-MST construction. This is also linked to the number of message duplications illustrated by Fig. 3. A high number of splits implies shorter paths.

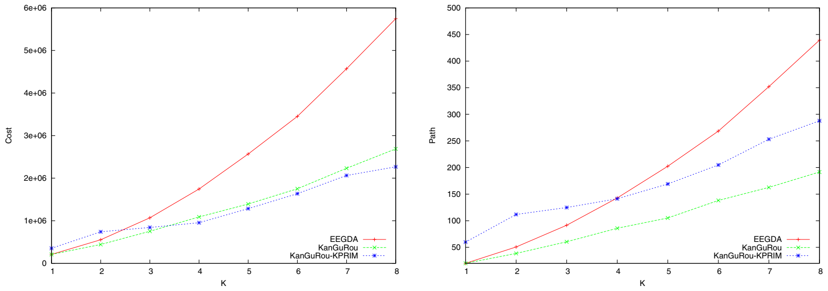


(a) Energy consumption

(b) Path length

Fig. 4. Algorithms performances with regards to N and k for $M = 10\% \times N$

Figure 5 gives a closer look at the energy consumption and the path length in terms of k when the total deployed nodes N is a constant ($N = 75$) and M is set to be 8 sinks. We can see KanGuRou-kPrim performs better regarding energy consumption when k is greater than 3, and KanGuRou always has a gain of the path length compared to the other two algorithms in this case.

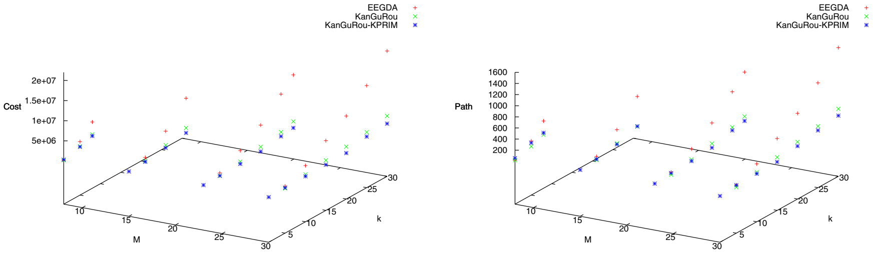


(a) Energy consumption

(b) Path length

Fig. 5. Algorithms performance in terms of k over $M = 8$ sinks among $N = 75$ nodes

In the second scenario (Fig. 6), we fix the number of the total deployed nodes N to 75 and evaluate the performances of the three algorithms (EEGDA, KanGuRou, KanGuRou-kPrim) regarding the overall number of sinks M in the network. Obviously, when k increases for a given number of available sinks M, the path and the energy consumption increase since there are more sinks to reach. Similarly, when the number of sinks to reach k is fixed and that the number of available sinks M increases, the path and the energy consumption decrease since algorithms have more choice among sinks and can join closer ones. An important feature is that results show that KanGuRou-kPrim performs better than KanGuRou for high values of M and k. Once again, this is linked to the number of path splitting and that the greater k, the closer to the optimal k-MST, k-Prim algorithm is.



(a) Energy consumption

(b) Path length

Fig. 6. Algorithms performances with regards to M and k for $N = 75$ nodes.

To sum up, the simulation results of different scenarios clearly show that *(i)* KanGuRou variants result in a significant gain of energy consumption and path length compared to the traditional algorithm EEGDA, *(ii)* depending of the percentage of sinks to be reached, one variant of KanGuRou performs better than the other one. When k is small (when $k \leq 30\% \times M$), KanGuRou always consumes less energy than KanGuRou-kPrim, *(iii)* when k is important (when $k > 30\% * M$), KanGuRou-kPrim brings a significant gain compared to KanGuRou especially when M is important. This is highlighted by Fig. 7 which has a closer look at this feature. Figure clearly shows that up to a given number of available sinks, KanGuRou-kPrim performs better than KanGuRou ($M = 23$ on figure).

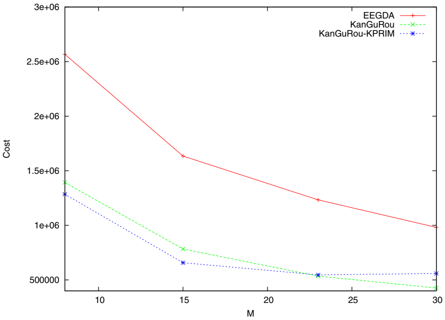


Fig. 7. Algorithms performances for $k = 5$ and $N = 75$ nodes

6 Conclusion and Future Works

In this paper, we have introduced KanGuRou, the very first position-based k-anycast routing protocol which is energy efficient and guarantees the packet delivery. Two variants are proposed for the construction of the tree. KanGuRou performs well when the number of sinks to reach is lower than 30% of the available sinks in the network while KanGuRou-kPrim performs better for higher values of k. In future work, we intend to claim theoretically how far KanGuRou is from the optimal centralized algorithm and provide some complexity analysis. We also intend evaluate the properties of KanGuRou more deeply (robustness toward mobility, wireless instability, etc).

References

1. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks 7(8), 609–616 (2001)
2. Wu, B., Wu, J.: k-anycast routing schemes for mobile ad hoc networks. In: IPDPS (2006)
3. Elhafsi, E.H., Mitton, N., Simplot-Ryl, D.: Energy Efficient Geographic Path Discovery With Guaranteed Delivery in Ad hoc and Sensor Networks. In: IEEE PIMRC (2008)
4. Frey, H., Ingelrest, F., Simplot-Ryl, D.: Localized mst based multicast routing with energy-efficient guaranteed delivery in sensor networks. In: WOWMOM (2008)
5. Mitton, N., Simplot-Ryl, D., Stojmenovic, I.: Guaranteed delivery for geographical anycasting in wireless multi-sink sensor and sensor-actor networks. In: IEEE INFOCOM (2009) (short paper)
6. Prim, R.C.: Shortest connection networks and some generalizations. Bell System Technical Journal 36, 1389–1401 (1957)
7. Rodoplu, V., Meng, T.: Minimizing energy mobile wireless networks. IEEE JSAC 17, 1333–1347 (1999)
8. Wang, W., Li, X.Y., Frieder, O.: k-anycast game in selfish networks. In: ICCCN (2004)
9. Wang, X.: Analysis and design of a k-anycast communication model in ipv6. Comput. Commun. 31, 2071–2077 (2008)
10. Wu, B., Wu, J.: k-anycast routing schemes for mobile ad hoc networks. In: IPDPS (2006)
11. Xu, X., Gu, Y.-L., Du, J., Qian, H.-Y.: A distributed k-anycast routing proto col based on mobile agents. In: WiCOM (2009)

An Admission Control Scheme Based on Links' Activity Scheduling for Wireless Mesh Networks

Juliette Dromard, Lyes Khoukhi, and Rida Khatoun

Troyes University of Technology
12 rue Marie Curie 1010 Troyes Cedex France

Abstract. Wireless Mesh Networks (WMNs) are low cost, easily deployed and high performance solution to last mile broadband Internet access, however they have to deal with a lack of bandwidth which prevents the deployment of applications with strict constraints. To overcome this limitation, we introduce a novel WMN model integrating both a transmission scheduling algorithm and a bandwidth-based admission control scheme. Most existing admission control schemes under-exploit the channel's capacity (due to approximations in node's bandwidth and flows consumption estimation) and under exploit the possibilities of parallel transmissions. In this paper, we propose a network model based on relation between links to get an accurate estimation of nodes bandwidth and flows consumption. Based on this model, we present an admission control scheme which relies on a transmissions scheduling algorithm favouring parallel transmissions, and on an advertisements scheme enabling nodes to be aware of the activities going on in their vicinities. Thus, nodes gain control over their channel and can thus estimate more precisely their bandwidth and exploit the spatial reuse from parallel transmissions. The overall network capacity and fairness is so improved.

1 Introduction

WMNs are autonomous networks, made up of three types of entities; mesh clients (MCs) which inject data relayed by fixed mesh routers (MRs) in a multi hop ad hoc fashion until reaching a mesh gateway (MG). MGs act as a bridge between the wireless network and the Internet. These networks are low cost, easily deployed, resilient and enable ubiquitous wireless technology. Indeed, they can extend Internet access in areas where cables' installation is impossible or economically not sustainable such as hostile areas, battlefields, old buildings and rural areas, etc [1].

However, due to unfairness between flows created by nodes' competition to access the shared network and to wireless unreliable channels, the capacity of WMN nodes are limited. This capacity tends to decrease with the number of hops between a node and a gateway, and this can compromise the scalability of WMNs. These issues prevent the deployment of very strict constraints applications.

To overcome the lack of capacity and interference issues of WMNs, it is of central importance to integrate QoS schemes. The main QoS schemes are QoS routing, admission control (AC), resources reservation, traffic policing, traffic

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 399–412, 2012.

scheduling, and QoS MAC protocol [3]. These schemes are often included in each other; indeed, an AC scheme may integrate a traffic scheduling, a resource reservation scheme, etc. However, these schemes mainly differ from each other by their goals. For instance, an AC scheme aims at accepting or rejecting a new flow as long as the WMN is able to guarantee its QoS and the QoS of previously accepted flows. The AC's efficiency depends on the accuracy estimation of nodes' available bandwidth and of flows consumed bandwidth. However, the bandwidth estimation in a WMN is hard to calculate precisely due to stochastic nodes' access to the channel and to the dynamic of the network. Most of existing AC models, as we will be described in the next section, suffer from these issues.

In this paper, we propose a novel AC model based on a time slotted mode in a topology-aware WMN. Its originality lies on:

– a novel and accurate method to estimate link's bandwidth considering its interference range calculated according its link's length, parallel transmissions and distinguishing different link's impacts with its close links.
– a transmissions scheduling algorithm which guarantees interference free and favours spatial reuse from parallel transmissions.
– a bandwidth-based admission control relying on the transmissions scheduling algorithm including an advertisements scheme of scheduled transmissions. This advertisements scheme enables nodes to get knowledge to take efficient decisions.

The paper is organized as follows. Section 2 briefly discusses related work in the field. Our network model and link's bandwidth estimation method is presented in Section 3. Section 4 describes the proposed transmissions scheduling algorithm, whereas, Section 5 details the proposed bandwidth-based admission control protocol. Section 6 analysis our simulation's results, and finally the last section concludes and introduces our future work.

2 State of Art

The aim of an AC is to determine whether the available resources is sufficient to meet the requirements of new flows while preserving resource level for existing flows. The network capacity is an important element to consider in the design of AC policies; it is considered as the principal limiting network resource factor, [3], due mainly to bottleneck collision domains [4]. Indeed, in a WMN, the throughput of each node decreases as $O(\frac{1}{n})$ where n is the total number of nodes [4].To optimize the network capacity use, some researchers focus their attention on pure AC strategies. AC has a central impact on avoiding congestion, ensuring fairness between nodes, spreading the network's load, etc. In [3], the authors provide an excellent survey on AC schemes for 802.11 multi-hop mobile ad hoc networks (MANETs). This survey illustrated that AC protocols mainly differ from each other by the chosen parameters and the parameter's estimation methods on which rely the AC decision. These parameters are usually node's available bandwidth and flow's consumed bandwidth.

CACP (Contention Aware Admission Control for Ad Hoc Networks) [10] is a landmark in the design of AC protocols, it is the first work which considers the capacity of neighboring stations situated in the node's carrier sensing range and the intra-route contention for a flow admission. CACP's parameter's estimation is thus more accurate, and so is the admission control. However, they don't deal with hidden nodes and spatial reuse, and they estimate node's collision domain and carrier-sensing range at two hops, which is not always the case [2].

The authors in [5] present an AC scheme for multi-rate wireless ad hoc networks. Their AC improves estimation's parameters by considering the probability of spatial reuse. Despite these improvements, this protocol still underestimates the available capacity [3]. Furthermore as CACP, the work does not consider hidden nodes and approximates collision domain range.

Shen et Al. [8] propose a novel probabilistic approach to estimate the available bandwidth which does not trigger any overhead and considers the impact of hidden terminals in WMNs. Upon this available bandwidth estimation, they design ACA (Admission control Algorithm), which differentiates QoS levels for various traffics.

The authors of RCAC (Routing on Cliques Admission) [7] consider packet loss and end to end delay parameters in the AC parameters. They achieve scalability since they partitioned the network into cliques where only clique heads are involved in the AC decision.

Most of the cited works do not introduce the interference range and do not differentiate it from the carrier sensing range (see section 2), and yet the interference range has an important impact on node's bandwidth as it determines the collision domain. Furthermore, they estimate node's bandwidth as though all node's links had the same bandwidth which is not the case as explained in the next section. Some works express the lack of accuracy of the parameters' estimation due to nodes' stochastic access to the network (most of these schemes are based on IEEE 802.11) which does not allow nodes to enforce parallel reuse and to get control on their channel.

3 Network Model

Our network model is based on transmissions scheduling and includes a new method to calculate link's bandwidth which takes in consideration the link's interference range calculated according its length, parallel transmissions and which distinguishes links' impact on each other. We are so going to model the network architecture with the nodes and their connections, the transmission scheduling on link and links bandwidth.

3.1 The Architecture Model

In our work, we only consider the WMN's backhaul made up with mesh routers. Mesh routers, also called nodes or stations in our paper, have all the same radio parameters. A node possesses two radio ranges, the transmission range and the carrier-sensing range, and each link possesses an interference range. Some explanations are given hereafter :

- transmission range (R_{tx}) : stations situated in a node's transmission range can receive, if there is no interference, all the messages sent by the node. There is a directed link from the node (the transmitter) to each station (the receiver) situated in its transmission range.
- carrier-sensing range (R_{cs}) : a node can hear but not always understand the packets sent by stations situated in its carrier sensing range. The carrier-sensing range is often twice larger than the transmission range.
- interference range ($R_{ti}(e)$) : a link possesses an interference range around the link's receiver. A station in this range can interfere with the link's transmission. This interference range depends on the length l of the directed link. According to the analytical model proposed in [9], this range is equal to $1,72 \times l$ with the Two Ray signal propagation model and $3.16 \times l$ and with the Free Space one.

As shown figure 1, the interference range of a link is not always include inside the carrier sensing range of the link's transmitter, which creates a hidden nodes zone. The collision domain (CD) of a directed link is the set of directed links whose transmitter is situated in its interference range and which can so interfere with its transmissions.

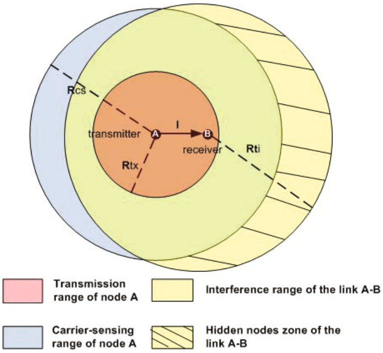

Rcs
transmitter
A
l
B
receiver
Rti
Rtx
Transmission range of node A
Interference range of the link A-B
Carrier-sensing range of node A
Hidden nodes zone of the link A-B

Fig. 1. Radio ranges and the hidden nodes zone

We formally model the backbone of a WMN as a directed graph, $G(V, E)$, where V represents the set of mesh nodes, and E represents the set of links between the nodes, $E = \{(u, v) \in V^2 | \, d(u, v) \leq R_{tx}\}$ with $d(u, v)$ is the Euclidean distance between nodes u and v. Among the nodes of V, we consider $V* \in V$ be the set of gateways that connect the network to the internet.

We denote $G'(V, D)$ the collision domain hypergraph, where $D = \{D(e), \forall e \in E\}$ is the set of the network's CD. Each link's CD is made up of the set of links which transmitters is situated in its interference range; the activation of one of these links prevents it from successful transmissions. For every directed edge $e \in E$ with $e = (u, v)$, its CD is the set $D(e) = \{(w, z) | (w, z) \in E \ d(v, w) \leq R_{ti}(e)\}$, $R_{ti}(e)$ is its interference range. This hypergraph can be represented by the *collision domain matrix* G' with dimension $|E|^2$; each element of G' is denoted by g'_{ij} and its value indicates whether the i^{th} link belongs to the CD of the j^{th} link :

$$g'_{ij} = \begin{Bmatrix} 1 \; si \; e_i \in C(e_j) \\ 0 \quad otherwise \end{Bmatrix} \tag{1}$$

3.2 Transmissions Scheduling Model

Network's nodes are synchronized and split the time in different intervals; the transmissions scheduling (TS) intervals and time unit (TU) intervals (see figure 2). Each TS interval is composed of TUs. A transmission scheduling aims at selecting the TUs of the TS interval, during which a flow is sent on a link. TS interval has $nb(TU_{TS})$ time units. In what follows to alleviate the figures, we fix the channel capacity at 1Mbit/s, the period of a TS interval at 1 second and of a TU at 0,1 ms. Each node possesses a view of the transmissions made by its link's CD, it is its links interference range view. In the figure 2, node C and D have a view of their link C-D's interference range, this view shows a unique transmission : flow f_1's transmission. Transmissions scheduling is represented by the *Transmission Matrix* T of dimension $|E| \times nb(TU_{TS})$, where the value of each element t_{ij} of T indicates whether the i^{th} link has scheduled a transmission at the j^{th} TU of the TS interval.

$$t_{ij} = \begin{cases} 1 & if \; e_i \; transmits \; at \; the \; j^{th} \; TU \\ 0 & otherwise \end{cases} \tag{2}$$

3.3 Link's Bandwidth Model

The bandwidth is a measure of available or consumed data communication. During a CA, nodes on the possible route(s) from a source to a destination are going to check whether they are able to support the bandwidth that should consume the new flow on their links. A route is eligible if it is made up of links that all can support the flow. The performance of the CA depends on the estimation accuracy of links' available bandwidth and of consumed bandwidth by flows and on the optimal utilization of the bandwidth. However, estimating these bandwidth is not a trivial task as the link bandwidth is shared with neighboring links, and flows can use simultaneously the same link bandwidth. Any link $e \in E$ possesses two sets of links which influence its bandwidth:

- the set of e's *impacting links*. e's *impacting links* are the set of links which transmitter belongs to e's interference range. When an *impacting link* of e is in activity, it interferes with e's transmissions and consumes TUs on e's TS interval; link e is so impacting by its activity. For example in figure 2, link G-H is an *impacting link* of E-F and consumes E-F's TUs.

- the set of e's *impacted links*. e's *impacted links* are the set of links which possess the transmitter of link e in their interference range. When e is active, it interferes with the transmissions of its *impacted links* and consumes their TUs. For example in figure 2, link C-B is an *impacted link* of E-F. Indeed, C-B consumes TUs of link E-F if it sends data.

Sometimes, an *impacting link* of e can be also an *impacted link* of e but not always. It would be always the case, if all the links had the same interference range and so the same length.

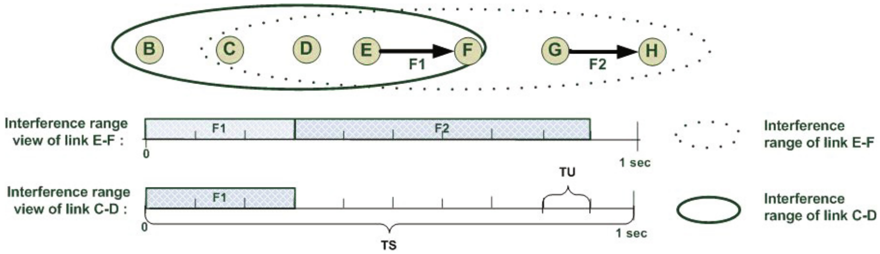

B
C
D
E
F
F1
G
H
F2
Interference range view of link E-F :
F1
F2
0
1 sec
Interference range of link E-F
TU
Interference range view of link C-D :
F1
0
1 sec
TS
Interference range of link C-D

Fig. 2. Views of link E-F and C-D interference range. f_1's rate is 300 kbit/s and $f_2's$ rate is 600 kbit/s.

Link's Available Bandwidth. It is the amount of data that can be sent through a link without interference and blocking issues. The interference and blocking phenomenon lead us to differentiate the *link* av*ailable local bandwidth* which only considers the interference issues, and the *link available global bandwidth* which considers both blocking and interference issues. Interference occurs when a link and one of its *impacting links* are active simultaneously, so, a link must not be activated when one of its *impacting link* is already emitting. The *link available local bandwidth* is the amount of unconsumed bandwidth observed in its interference range. The *available local bandwidth* of a link e, $B_{loc}(e)$, is proportional to the number of free TUs noted $nb(TU_{free}(e))$ in its TS interval, thus :

$$B_{loc}(e) = \frac{nb(TU_{free}(e)) \times TU}{TS} \times C \tag{3}$$

with TU the time in second of a time unit and TS the time in second the TS interval and C the channel capacity. However a link can respect its *local available bandwidth* and yet prevents some link(s) it impacts from sending previously admitted flow(s); these flow(s) are so blocked. Indeed, when a link is activated, it uses *impacted links' TUs* and decreases so their bandwidth. If the *available local bandwidth* of one of its *impacted links* becomes inferior to the bandwidth requirements of its flows, the link can not send any longer all its flows, one ore more of its flows are so blocked. Figure 2 illustrates a scenario before blocking and figure 3 after blocking. According to its interference range's view (see figure 2), the link C-D has enough *available local bandwidth* (700 kbit/s) to send the flow f_3 of 400kbit/s rate. Nevertheless, this flow 's transmission blocks the transmission of flow f_1 on link E-F as shown figure 3.

However, blocking can be in some cases avoided thanks to spatial reuse. Indeed, a link possesses *impacted links* which can send in parallel if they don't belong to each other's *impacting links* set. If *impacted links* of a link send in parallel rather than successively, then, they consume less links bandwidth, it can so prevent the link's available bandwidth from becoming inferior to its flow(s) requirement(s). In the figure 2, as the links C-D and G-H are impacted links of E-F and don't belong to each other set of impacting links, they can send simultaneously their flow f_3 and f_2, avoiding flow f_1's blocking (see figure 4).

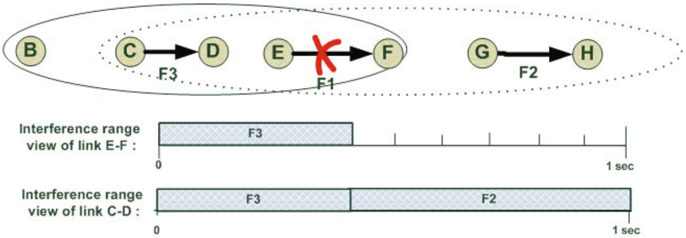

B
C
D
E
F
G
H
F3
F1
F2
Interference range view of link E-F :
F3
0
1 sec
Interference range view of link C-D :
F3
F2
0
1 sec

Fig. 3. The transmission of flow $f_3$blocks flow f_1 transmission

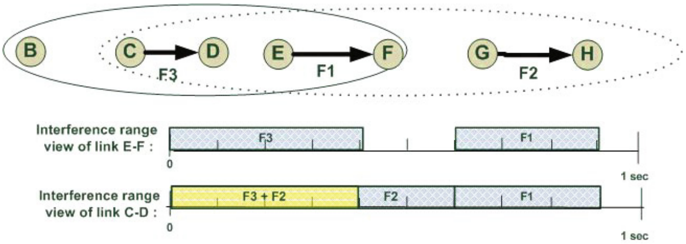

B
C
D
E
F
G
H
F3
F1
F2
Interference range view of link E-F :
F3
F1
0
1 sec
Interference range view of link C-D :
F3 + F2
F2
F1
0
1 sec

Fig. 4. Flows f_3 and f_2 are transmitted in parallel avoiding the blocking of f_1

If the node knows the transmissions scheduling of the set of its impacted and impacting links, it can so schedule its transmissions in order not to use the TUs during which these links has planned to be activated. It can so avoid blocking flows or interference. We propose the algorithm 1 which enables to identify the TUs available for a link e, i.e the TUs available for the link to transmit data without interference or blocking issues. This algorithm returns for a link e, a unit matrix of available TUs $R(e)$, which elements $r(e)_{1i}$ specifies if the i^{th} TU of the TS interval is available for the link e, if the TU is available $r(e)_{1i} = 1$, if it is not $r(e)_{1i} = 0$. The number of e's available TUs in its TS interval noted $nb(TU_{av}(e))$, is equal to the sum of the elements of its matrix $R(e)$:

```
input  : $g'_{xz}$ (the $z^{th}$ collon of matrix $G'$ representing $e_z$'s set of
         impacting links), $g'_{zx}$ (the $z^{th}$ line of matrix $G'$ indicating $e_z$'s set of
         impacted links) and matrix $T$
output : $R(e_z)$ (a linear unit matrix of dimension $nb(TU_{TS})$)
$P = (g'_{xz})^t \times T$ // an element $p_{1j}$ of $P$ indicates the number of
    $e'_z s$ impacting $link(s)$ which uses the $j^{th}$ TU
$P' = g'_{zx} \times T$ // an element $p'_{1j}$ of $P$ indicates the number of
    $e'_z s$ impacted $link(s)$ which uses the $j^{th}$ TU
$P = P' + P$ // an element $p_{1j}$ of $P$ indicates the number of
    $e's$ impacted and impacting links which use $j^{th}$ TU
for $j = 1, j < nb(TU_{TS}), j++$ do
    if $p_{1j} \geq 1$ then
        $r(e_z)_{1j} = 0$ // link $e_z$ ca not be activated during the $j^{th}$ TU
            due to blocking and or interference issues
    else
        $r(e_z)_{1j} = 1$
    end
end
return $R(e_z)$
```

Algorithme 1 : Available Time units of link e_z

$$nb(TU_{av}(e)) = \sum_{i=1}^{i=nb(TU_{TS})} r_{1i} \tag{4}$$

The global available bandwidth of a link is proportional to its available TUs. For a link e, its global available bandwidth noted $B_{glob}(e)$ is:

$$B_{glob}(e) = \frac{nb(TU_{av}(e)) \times TU}{TS} \times C \tag{5}$$

Flow's Consumption on Link's Bandwidth. Each flow requires a bandwidth which can be expressed in term of TUs required in the TS interval. Here is the formula which convert the flow's f requirements bandwidth BR_f in number of TUs required by the flow $nb(TU_f)$:

$$nb(TU_f) = \left\lceil \frac{BR_f}{C \times TU} \right\rceil \tag{6}$$

By sending a flow f, a link consumes $nb(TU_f)$ TUs at its TS interval, and also $nb(TU_f)$ TUs or minus at its *impacted links*. It can consume less at an *impacted link*, if it sends during TUs already consumed by another impacting links of this impacted link. In case of parallel transmissions, which link consumes the TUs ? For example in the figure 4, is it the link G-H or C-D which consumes the four first TUs of E-F's TS interval ? Thereafter, we admit that for parallel transmission, the link sending the older flow is the one which consumes the TUs. So, when a flow f is sent on a link during $nb(TU_f)$ time units, it consumes on each of its *impacted link* e a number of TU noted $nb(TU_f(e))$. $nb(TU_f(e))$ is equal to the number of TUs required by the flow minus $nb(TU_{f-par}(e))$, the TUs during which it is sent in parallel with an older flow on another *impacting link* of e :

$$nb(TU_f(e)) = nb(TU_f) - nb(TU_{f-par}(e)) \tag{7}$$

For example, in the figure 4, the flow f_3 has been admitted after flow f_1, they are sent in parallel on parallelizable links of E-F, so flow f_3 does not consumes any TU of link E-F whereas f_1 consumes 4 TUs. Parallel transmissions enable to consume less link's bandwidth. To enforce parallel transmissions, nodes need to control their channel and be aware of transmissions scheduling going on in their link's vicinity. This enforcement can be realized using a scheme of transmissions scheduling.

4 Transmission Scheduling Scheme

Each node is going to schedule its transmissions and more precisely schedules the TUs in its TS interval during which it is going to activate its links. The transmissions scheduling is made according to flow's requirements. In this section, we present a transmission scheduling scheme which aims at improving the network capacity thanks to spatial reuse. To reach this goal, it includes an advertising scheme for transmission scheduling and an algorithm of transmission scheduling. We assume that each node in the network is aware of the network topology. A node could get these information thanks to a manual intervention or a learning scheme.

4.1 An Advertising Scheme

To schedule a transmission, each node must be aware of what are the transmissions scheduling on its vicinity's links. Thus, the node must know the interference range view of its links and of their *impacted* and *impacting* links. To achieve this goal, a node possesses an advertising scheme; each time a node schedules a transmission on one of its link, it broadcasts an advertisement packet which is relayed by its neighbors to reserve TUs for its transmission. A node, which receives the packet, registers the information of the transmission scheduling, and forwards the packet if the packet's Time-To-Live (TTL) has not expired. This TTL must be large enough so that all nodes obtain the advertisements it needs to retrieve the interference range view of the *impacted* and *impacting* links of the node's links. An advertising packet includes, the position of the TUs in the TS interval during which the node has planned the transmission, the link used for the transmission and the flow id to transmit. 20% of the TS interval must never be scheduled and stay available for control packets.

4.2 Transmissions Scheduling Algorithm

A node receiving an AC's request for a flow must decides firstly whether it accepts or rejects the flow. If it accepts, it then schedules the flow's transmission. We propose a transmission scheduling algorithm which, if the link's available bandwidth is sufficient, selects the TUs during which the link must be activated in order to favor spatial reuse. It is a progressive filling algorithm as it schedules TUs for a transmission by observing its impacted links, one by one, from the less to the most load, and until it reserves all the TUs required for the flow transmission. If the algorithm succeeds, it returns a unit matrix P of dimension $nb(TU_{TS})$, where elements p_{1j} indicates whether the j^{th} TU is reserved for the transmission $p_{1j} = 1$ or not $p_{1j} = 0$. Every elements of matrix P is initially initialized at 0, as no TU is then reserved for the transmission. To schedule a transmission for a flow f on a link e, the transmission scheduling algorithm achieves the following steps :

1. Checks if the link has enough available bandwidth to admit the flow, if it is not the case the algorithm ends and the scheduling fails. As a link must always keep 20% of its TS interval's TUs free for control packets, the link e can admit the flow if :

$$nb(TU_{av}(e)) - (nb(TU_{TS}) \times 0,20) > nb(TU_f) \tag{8}$$

2. Determines the set of impacted links I of link e and computes the matrix $R(e)$, obtained with the algorithm 1 and which represents the available TUs of link e.
3. Chooses among the set I, the impacted link e_z which has the less available TUs and so bandwidth and computes $R(e_z)$ the matrix of available TUs of link e_z obtained with the algorithm 1.
4. Computes the potentially reserved TUs matrix R' of dimension $nb(TU_{TS})$, from the two previous matrix $R(e)$ and $R(e_z)$:

$$r'_{1j} = \begin{cases} 1 & if\, r(e)_{1j} = 1\, and\, r(e_z) = 0 \\ 0 & otherwise \end{cases} \tag{9}$$

   Element ${r'}_{1j}$ of R' indicates whether e can send data in parallel with another *impacting link* of e_z's set of impacting links at the j^{th} TU. If it can $r'_{1j} = 1$, otherwise $r'_{1j} = 0$.
5. Computes $nb(TU_{resv})$ the number of TUs of e's TS interval which are either already reserved or could be potentially reserved :

$$nb(TU_{resv}) = R' + P \tag{10}$$

   If $nb(TU_{resv})$ is equal to $nb(TU_f)$, it adds to matrix P, the matrix R', the algorithm ends and returns P. If it is superior, it removes randomly in R' some potentially reserved TUs in order that when it adds to matrix P, the matrix R', the TUs reserved are equal to $nb(TU_f)$, then the algorithm returns P and ends. If it is inferior, every TU in R' becomes reserved TUs, so it adds to matrix P, the matrix $R'$and then go to the next step.

6. TUs which has just been reserved in step 5 are not longer considered as available for link e. The matrix of available TUs of link e is so modified:

$$R(e) = R(e) - R' \tag{11}$$

7. It then retrieves from the set I the impacted link e_z. If the set I is not empty, it goes back to the step 3, otherwise it goes to step 7
8. It chooses randomly among the available TUs of link e , the TUs it sill needs to reserve, to reach the flow requirements. It modifies in consequence the matrix P and returns P before ending.

5 Admission Control Policy

Our admission control policy is based on the transmission scheduling scheme presented in the last section. The AC relies on the reactive routing protocol AODV [6](Ad hoc On Demand Distance vector) routing protocol and takes place in two steps, the route discovery via the Route Request packet (RREQ) and the route establishment, via the Route Response packet (RREP).

5.1 Route Discovery

During this step, the source broadcasts a route request, in which it specifies the bandwidth required for the flow. Each node receiving the route request carries out these four following steps.

In a first time, it checks whether it has already received the packets or not. In this case, it discards the packet. Secondly it registers temporally the information contained in the RREQ. Indeed, each node adds to the RREQ its transmission scheduling for the flow. In a third step, it checks whether it is able or not to support this flow and schedules the flow transmission of the link the RREQ has just crossed and for which the is the receiver.Finally during the fourth step, it adds its transmission's scheduling information to the packet, then, broadcasts the RREQ.

This scheme goes on till the TTL expires or the destination is reached. If the destination successfully performs the three first steps, then it adds its transmission's scheduling information to a RREP packet and broadcasts the RREP along the RREQ inverse path .

5.2 Route Establishment

During this step, each node except the source, receiving the RREP performs these four following steps.

In a first time, it checks whether it has received advertising packets which invalidate the transmission scheduling the node has made during the step of route discovery. In this case, it discards the RREP. Secondly, it registers the information contained in the RREP. Each node adds to the RREP its transmission scheduling for the flow it made during the route discovery. In a third step, it broadcasts an advertising packet, for the link scheduling made during the route discovery. Finally during the fourth step, it adds the transmission scheduling it made during the route discovery. Then, it unicasts the packet to the following node, of the RREQ reverse path.

This process goes on till the source is reached. After the source has made the two first step successfully, it starts sending the flow.

6 Performance Evaluation

In order to evaluate our proposed model using ns2, we have chosen to compare it with the 'original model' which is the basic MAC IEEE 802.11 (i.e. without our model functionalities) using AODV protocol. We calculate the TUs in the link scheduling TS interval during which a link must be activated to send the flow. We evaluate the performance of our solution in terms of throughput and packet loss rate. We use a chain topology of eleven nodes, where every node is separated of 9 meters, and the middle node represents the gateway (see figure 5. Simulation parameters and their default values are listed in table 1. In all the following scenarios, every node except the gateway sends one CBR flow with the same rate.

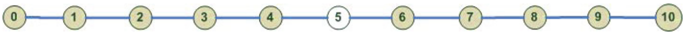
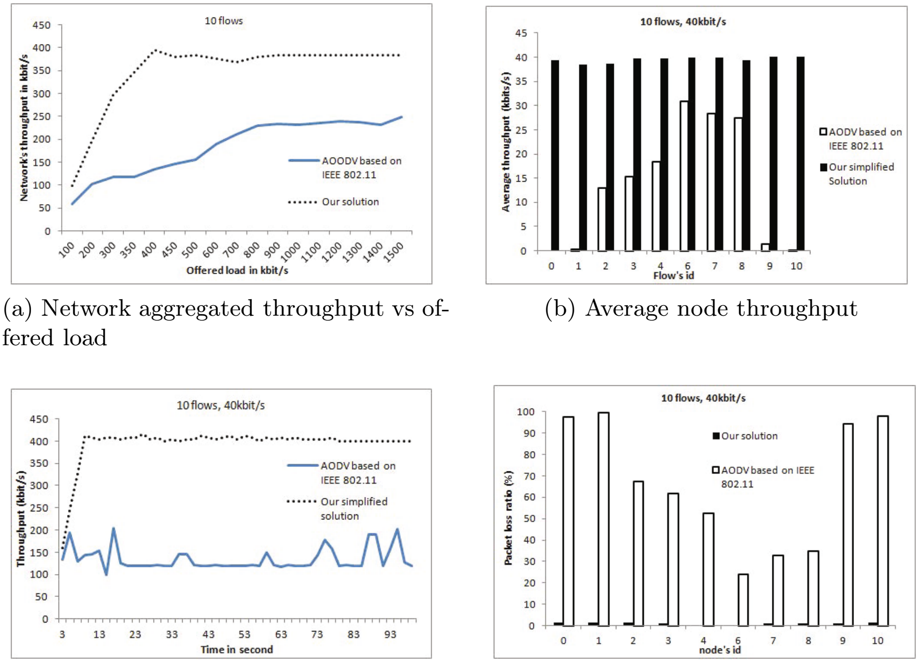


Fig. 5. Network's topology

Table 1. Simulation parameters

Layer	Parameters	Values
Physical layer	transmission range	10 meters
	carrier-sensing range	20 meters
	channel capacity	54Kbit/s
MAC layer	TS interval	1 second
	TU interval	50000µs
Transport layer	UDP size packet	1000 bytes

Figure 6a presents the variation of the aggregate network throughput with the increase in offered load. It shows an increase in the aggregated network throughput of our model compared to the original model, this is explained by the fact that our model does not suffer of interference and takes advantage of spatial reuse. Our solution reaches its maximum network throughput at 400kbit/s when the offered load is approximately of 400kbit/s. Beyond this offered load, the aggregated network throughput stays stable. For all the following figures, each node sends except the gateway a flow of rate 40kbit/s.



(a) Network aggregated throughput vs offered load

(b) Average node throughput

(c) Throughput in time

(d) Packets loss ratio

Fig. 6. Performance evaluation

Figure 6b compares the average node's throughput between our model and the original model. Our scheduling scheme establishes fairness between nodes, as every flow achieve the same throughput. In contrary, the original model leads to unfairness between nodes, the node's throughput decreases with the increase in the number of hops from the node to the gateway. Figure 6c compares the throughput evolution versus the time of simulation. As we can observe, our scheduling scheme presents a network throughput superior to that of the original model. However, it takes a few seconds to reach its stable throughput, this is explained by the fact that a flow needs to wait at each intermediate node the next link to be activated. Figure 6d compares the nodes packet loss ratio between our model and the original model. Our solution presents a very low ratio of packet loss as transmission scheduling enables interference free, lost packets are only control ones. Indeed in the original model, packet loss ratio increases with the increase in the number of hops from the node to the gateway.

7 Conclusion and Future Works

In this paper, we have introduced an original method to calculate links bandwidth and a new admission control based on transmissions scheduling scheme in WMNs. To calculate, with accuracy, the network bandwidth, we take in consideration the relation between links highlighted by our network model. Our admission control enables nodes to gain control over their channel thanks to transmission scheduling and an accurate knowledge of their surroundings, and thus can take advantage of spatial reuse to enhance the network throughput. We have proposed a transmission scheduling scheme integrating a progressive scheduling algorithm favouring parallel transmissions and nodes with an important charge. The simulation results have shown the pertinence of our model in terms of throughput, packets loss and gain in fairness between nodes. In a future work, we plan to study the impact of the proposal model on other QoS parameters such as delay and jitter. Furthermore, it would be interesting to integrate a learning scheme permitting a node to be aware of the network topology and to discover more efficiently its vicinities during the design stage of the network.

References

1. Wireless mesh networks: a survey. Computer Networks 47(4), 445–487 (2005)
2. Aoun, B., Boutaba, R.: Max-min fair capacity of wireless mesh networks. In: 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), pp. 21–30 (2006)
3. Hanzo, L., Tafazolli, R.: Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey. IEEE Communications Surveys & Tutorials 11(4), 78–108 (2009)
4. Jun, J., Sichitiu, M.L.: The nominal capacity of wireless mesh networks. IEEE Wireless Communications [see also IEEE Personal Communications] 10(5), 8–14 (2003)
5. Luo, L., Gruteser, M., Liu, H., Raychaudhuri, D., Huang, K., Chen, S.: A qos routing and admission control scheme for 802.11 ad hoc networks. In: Proceedings of the 2006 Workshop on Dependability Issues in Wireless ad hoc Networks and Sensor Networks, DIWANS 2006, pp. 19–28. ACM, New York (2006)
6. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (aodv) routing (2003)
7. Rezgui, J., Hafid, A., Gendreau, M.: Distributed admission control in wireless mesh networks: Models, algorithms, and evaluation. IEEE Transactions on Vehicular Technology 59(3), 1459–1473 (2010)
8. Shen, Q., Fang, X., Li, P., Fang, Y.: Admission control based on available bandwidth estimation for wireless mesh networks. IEEE Transactions on Vehicular Technology 58(5), 2519–2528 (2009)
9. Xu, K., Gerla, M., Bae, S.: How effective is the ieee 802.11 rts/cts handshake in ad hoc networks, vol. 1, pp. 72–76 (2002)
10. Yang, Y., Kravets, R.: Contention-aware admission control for ad hoc networks. IEEE Transactions on Mobile Computing 4(4), 363–377 (2005)

Capillary Machine-to-Machine Communications: The Road Ahead

Vojislav B. Mišić[1], Jelena Mišić[1], Xiaodong Lin[2], and Dragan Nerandzic[3]

[1] Ryerson University, Toronto, ON, Canada
[2] University of Ontario Institute of Technology, Oshawa, ON, Canada
[3] Ericsson Canada Inc., Mississauga, ON, Canada

Abstract. Machine-to-Machine (M2M) communications are expected to include billions of smart devices in the next three to five years. However, existing communication standards are incapable of providing satisfactory performance for M2M traffic. In this paper, we outline some advances that will enable existing wireless personal area networks, in conjunction with existing cellular communication standards, to be adapted to the needs of M2M traffic.

Keywords: machine-to-machine communications (M2M), capillary M2M, wireless personal area networks, wireless security.

1 Introduction

Data communication networks are among the most developed areas of technology in modern society. Machine-to-Machine (M2M) communications, also known as Machine-Type Communication (MTC), is one of the emergent communication areas poised for rapid growth in the coming years. M2M refers to data communication between smart electronic devices (hereafter referred to as nodes or terminals) that do not need human supervision or interaction, and they can (or will) be found in such diverse areas as smart power/smart grid, electronic health, transportation management, safety and security, and city automation. The number of M2M-enabled terminals is expected to grow from 50 million in 2008 to 200 million in 2014, and to tens of billions by 2020 [1].

At the same time, M2M networks must satisfy a number of very stringent requirements [2]. First, the sheer number of M2M-enabled smart devices is simply huge – well beyond any number that can be accommodated by current, or even future, cellular networks. Yet the traffic from each individual M2M terminal will be of low volume and the messages will be rather short – typically, periodic messages reporting relevant resource usage, with ad hoc messages reporting irregular events or emergencies. In many M2M applications (e.g., smart metering, fleet management, e-healthcare), a given M2M terminal will typically communicate with a single M2M server or user through the public data network (PDN), i.e., the Internet.

Downlink traffic from M2M servers to the terminals will be of much lower volume; it is expected to consist of short control packets, addressed to individual M2M devices, or groups formed on a per-type or per-area basis. In both cases, the overhead imposed by the communication protocols is likely to be higher than the actual message payload, which means that data aggregation at some point – preferably as close to the terminal itself as possible – would be desirable.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 413–423, 2012.

M2M terminals are expected to operate unattended for long periods of time which necessitates robust security mechanisms, in particular authentication, as well as the ability to detect compromised or misbehaving terminals and isolate them from the network as quickly as possible. (Other mechanisms, such as tamper-proof design, would also be needed, but they are beyond the scope of this paper.)

All of the above is further complicated by the fact that M2M terminals will often operate on restricted power, which necessitates energy efficiency and puts an additional burden on data communication capabilities.

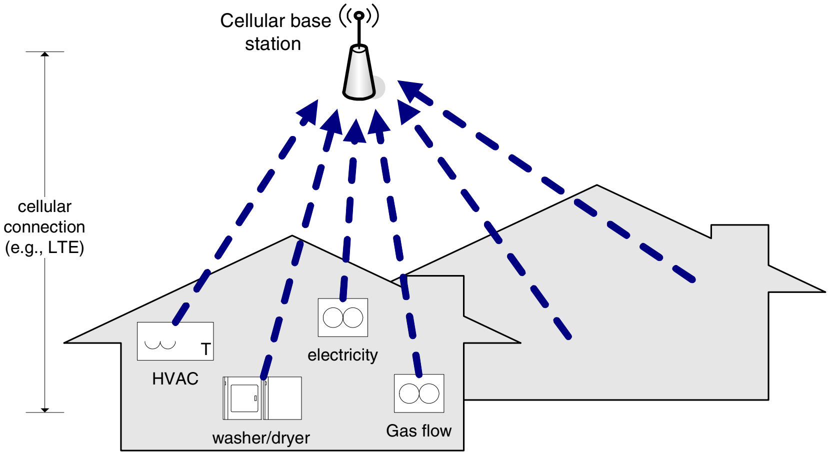


Fig. 1. Cellular M2M topology

In recent years, a number of possible approaches to M2M system and network solutions were proposed, and two main strategies have emerged. *Cellular* M2M refers to the architecture wherein each M2M terminal would have a direct link to a 4G cellular network such as 3GPP Long Term Evolution (LTE) [3] or WiMAX [4], from which it could connect to the relevant M2M servers through suitable gateway(s); this architecture is schematically shown in Fig. 1. Cellular approach offers many benefits, such as ubiquitous coverage, global connectivity with a number of providers, and well developed charging and security solutions. However, most of the requirements above can't be fulfilled by the current cellular technology, including advanced ones like WiMAX or LTE, without substantial changes in the relevant standards which incurs considerable cost in terms of both time and money.

In *capillary* M2M, terminals in a given area (e.g., a residential building) are organized in a mesh- or tree-topology capillary network, typically but not necessarily wireless, connected via a gateway to the cellular network. This architecture is schematically shown in Fig. 2. The main advantage of the capillary approach over the cellular one lies in its ability to aggregate and shape M2M traffic for further transport to the relevant M2M servers using a wired or wireless, or indeed a cellular network such as LTE or WiMAX. Yet even this solution requires new advances in terms of medium access, data aggregation, and security. Some of these advances are outlined in this paper; it is expected that they will ultimately lead to feasible capillary M2M networks that will serve the needs of current and future M2M traffic.

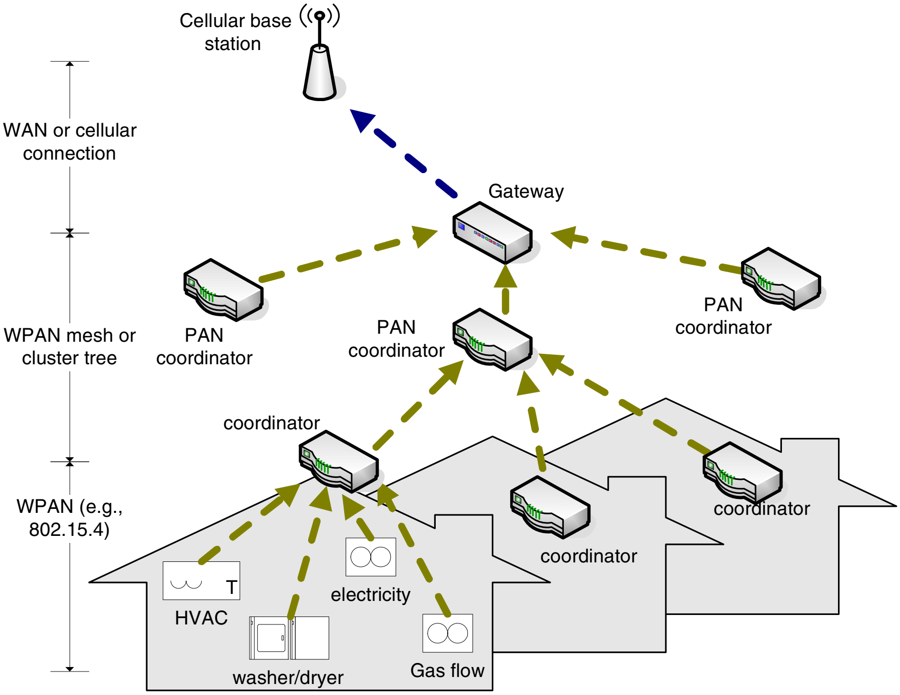


Fig. 2. Capillary M2M topology

The paper is organized as follows. We first present some of the main challenges of, and existing approaches to, cellular M2M, in Section II. In Section III, we describe changes in a number of important areas that will enable capillary M2M to fulfill the requirements for M2M traffic. Security aspects of M2M in the capillary environment are highlighted in Section IV. Finally, Section V concludes the paper.

2 Challenges of Capillary M2M

As noted above, the main obstacle to wider use of M2M, i.e., handling a large number of M2M terminals, might be removed by the use of capillary M2M, shown in Fig. 2 above. Most industrial proposals in this area are based on IEEE 802.15.4 low power WPAN technology with a suitable networking layer such as ZigBee [5,6], but this solution has serious problems, esp. in large-scale deployment:

If 802.15.4 standard is to be used, it must operate in unlicensed Industrial, Scientific, Medical (ISM) band at 2.4GHz, because the raw data rate of 250 kbps that is achievable in the ISM band is much higher than the corresponding data rates in other bands (868 and 915MHz). However, operation in the ISM band also leads to interference from IEEE 802.11b/g (WiFi) networks due to channel overlap, as shown in Fig. 1. As WiFi transmits at higher power, 802.15.4/ZigBee networks are forced to use channels with little or no interference, which reduces the number of available (i.e., reasonably interference-free) channels to at most four (shown in gray in Fig. 3). In fact, only channel 26 is entirely free from WiFi interference [7]. The resulting bandwidth is obviously insufficient for M2M traffic in heavily populated business and residential areas. Moreover, in scenarios where end-to-end reliability is needed, retransmission rate will increase, resulting in low data rates and vastly increased delays [8]. Communication quality may be further degraded by multipath fading, even on a channel with low interference.

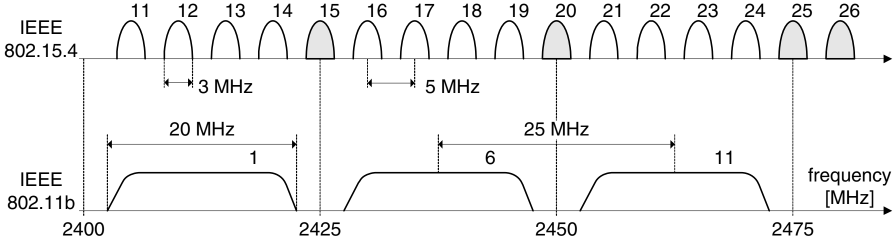


Fig. 3. IEEE 802.15.4 and 802.11b channels. IEEE 802.15.4 channels which experience less interference from IEEE 802.11 ones are shown in gray color.

Mesh ZigBee networks have no beacons and therefore must use non-slotted CSMA-CA, which leads to much lower utilization. Tree-based networks that use slotted CSMA-CA perform better but at the expense of long periods between beacons and short active superframe periods: the former allow only a small number of transmissions in a single superframe (which reduces transmission efficiency), while the latter requires tight synchronization constraints that apply not only to beacons but also to sleep patterns of individual nodes (otherwise nodes that wake-up during inactive period may have to wait too long for the piconet to become active and thus will effectively waste battery power) [9].

As the frame size is limited, little to no data aggregation can be done before packets reach the M2M gateway, which will further degrade transmission efficiency. (The M2M gateway will be discussed in more detail in the next Section.)

A recently proposed extension of the IEEE 802.15.4 standard, known as the IEEE 802.15.4e [10], combines 16 channels in the ISM band with TDMA-scheduled channel hopping. Channels and time slots are jointly allocated in a cognitive manner with respect to network topology and energy consumption. However, not all such combinations are usable because of interference and fading, and channel quality has to be accounted for; however, this is not addressed by the draft standard.

Some authors have investigated the possibility to use IEEE 802.15.1 (Bluetooth) to connect M2M terminals in a piconet, controlled by the piconet master which would aggregate data and send it (through another radio interface) to a LTE network [11,12]. However, neither of these proposals considers terminal grouping, routing, data aggregation, or energy management issues. Moreover, Bluetooth has a serious limitation regarding the number of devices—a Bluetooth piconet contains the master and up to 255 slaves, of which at most 7 can be active at any given time [13]—which renders it virtually unusable for any large-scale M2M network implementation.

3 Enabling Capillary M2M

Inefficiency in capillary M2M stems mainly from the inefficiency of the WPAN protocols used in the capillary network. The logical first step is, thus, to focus on improving the efficiency of those protocols.

3.1 MAC Protocol for Capillary Network

Given the problems with 802.15.4-WiFi interference described above, a feasible approach is to try to exploit not only spectral holes in the ISM band but also the time holes – inactive periods at the WiFi channels. Our preliminary work has shown that idle time in a WiFi channel has sub-exponential probability distribution [14]; transmission slots can be scheduled to occur in those time periods. The MAC protocol would require some cognitive abilities, i.e., nodes that will sense the activity on WiFi channels and analyze the statistics of this activity. Based on the analysis, the occurrence of time holes must be predicted and advertised to M2M nodes so that they can schedule their own transmissions to avoid collisions with WiFi transmissions. However, scheduling of time holes is, by default, less reliable and should be used only in cases when bandwidth of free channels is not sufficient.

3.2 MAC Protocol for Terminals with Multiple Radio Interfaces

The approach outlined above may benefit from the use of nodes with multiple, possibly heterogeneous radio interfaces [11]. A node, possibly mobile, might be equipped with 802.11 interface as well as an 802.15.4 and/or 802.15.1 (Bluetooth) one, and use the one with best connectivity at a given time. For example, it is known that 802.15.1 (Bluetooth) is most resilient to interference due to its use of Frequency Hopping Spread Spectrum (FHSS). We plan to develop a M2M MAC overlay with multiple interfaces which will discover nodes in vicinity and connect with them using the best technology.

3.3 Using TV White Space in the 460-790 MHz Band

Recently the FCC has allowed the unoccupied TV spectrum to be used by unlicensed devices, given that they do not interfere with TV (actually, digital TV) users. Communications in the TV band have certain features, including extended transmission range and non-Line-of-Sight propagation, which makes them attractive for future M2M communications in this band, esp. in less populated areas such as rural Canada. While the existing IEEE 802.22 MAC protocol [15] is not well suited to M2M traffic, on account of the sheer number of M2M terminals and their traffic characteristics, modifications to accommodate for M2M traffic might be possible. Alternatively, a novel cognitive protocol specifically tailored to M2M communications might be developed.

3.4 Design of the M2M Gateway (M2MG)

The cornerstone of the capillary M2M architecture is the M2M gateway (M2MG). M2MG will accommodate M2M traffic from the capillary M2M network organized as a wireless local or personal area network (WLAN or WPAN) with mesh- or tree-based topology, and connect to a wide area network. This network might be a high-performance wired network, or (preferably) a cellular network such as LTE or Wi-MAX. M2MG would thus need both a WLAN/WPAN radio interface and a cellular interface. M2MG will appear to the cellular network as a client that competes with cellular users but also with other M2M gateways.

As M2M data will consist of a large number of short, quasi-periodic messages from M2M terminals, direct transmission of these messages is not feasible. First, they are short and, thus, comparable with the overhead imposed by the existing cellular standards. Second, the sheer number of terminals means that the volume of traffic would be extremely high and direct addressing is infeasible, even with IPv6 addressing scheme.

A much better solution is, then, to aggregate data streams from a group of individual terminals in a given area (e.g., a high rise building or a residential street) and, possibly, of a given type (e.g., power meters or gas meters), before they enter wide area network. Aggregation will be performed by the M2MG, which will also need to keep track of terminal IDs so as to allow the commands from the M2M server to reach their proper recipients. M2M nodes can have local addresses, using a scheme similar to the well known Network Address Translation (NAT) executed at the M2M gateway.

M2MG will also need to prioritize the data, due to the differences in arrival times and the possibility that some data items (e.g., alarms) need to reach the corresponding M2M server faster than others (e.g., ordinary metering data).

Due to its higher traffic volume, a M2MG will need higher priority in accessing the medium than individual cellular devices (e.g., smartphones); appropriate provisions might need to be made in the respective cellular interfaces.

3.5 Sleep Management for M2M Teminals

Many M2M terminals operate on battery power and, thus, need sleep management which can be implemented using a randomized protocol [16]. In this scheme, the duration of the sleeping time is a random variable (e.g., with geometric probability distribution), the average value of which is calculated by taking into account frame collisions and required information throughput from each metering device. Each sleeping node wakes up shortly for each beacon and listens to the beacon frame. M2MG will advertise in each beacon frame the list of nodes for which it has queued downlink frames (command or configuration). If a sleeping node hears its own MAC address, it wakes up and transmits a data request frame after which M2MF will transmit downlink frame.

3.6 Design of Cross-Layer Routing Protocol for Capillary M2M

Routing in capillary M2M is conducted from the M2M terminal (e.g., a smart metering node) to the M2M gateway towards cellular network or Internet. Since the availability of channels for MAC depends on current interference and sleeping status of the uplink node, cross-layer collaboration of routing algorithms in conjunction with MAC and, possibly, even PHY layer protocols is needed to achieve optimum performance. Assuming that links are always available, node sleeping schedules can be tailored to address the performance requirements regarding activity of the network spanning tree. However under sporadic availability of links in frequency and time, network connectivity can not be guaranteed and as a consequence node access delay may have large deviations. In this area, novel reliable and delay tolerant routing algorithm which uses results from MAC and physical layers need to be developed.

3.7 Design of Charging Policies for M2M Traffic

An important component of the capillary M2M framework is an integrated solution for measuring and estimation of resource expenditure, in particular spectrum usage and airtime (these being network operator's most valuable resources) for each message, and subsequent storing this information in a suitable database. Received data is aggregated on a per-service basis and sent to the appropriate M2M server; resource expenditure information is used by the M2M service provider for billing purposes. The actual modalities of billing will depend on the details of service agreements between the network operator (MNO) and the respective service providers. Namely, the service provider might operate as a virtual network operator, leasing resources in bulk from the MNO. Alternatively, the MNO may charge the operator on the basis of actual traffic carried. Either way, individual subscribers (i.e., individuals and commercial entities that house the actual devices) may be charged by the service provider, MNO, or both. Different forms of partnership, the resulting charging policies and their impact on both short- and long-term revenues should be evaluated from the perspective of the MNO as well as from that of the service providers [17][18].

4 Security in Capillary M2M Systems

Most currently accepted security solutions are based on Authentication, Authorization and Accounting (AAA) architecture [19], which is not directly applicable to M2M application scenarios. Namely, many M2M terminals operate under power constraints which preclude the use of full fledged security solutions such as X.805 [20]; instead, low computational complexity algorithms and techniques must be used. We assume that the cellular core network (EPC) is secure, as are the M2M servers which are owned and operated by the MNO and M2M service providers, respectively. What remains to be addressed is, then, the security of other components of the overall M2M system: M2M terminals, M2M gateway, communication between the terminals and M2MG, and the M2M data, including subscriber information.

A survey of security threats focusing on remote provisioning and subscription-related issues is given in [21] where some alternative solutions are identified but not elaborated in great detail; similar survey but much with much reduced scope can be found in [22]. The concept of dedicated module that provides a trustworthy environment (TRE) within the M2M terminal for the execution of software and storage of sensitive data (including keys, credentials, and authentication data) is elaborated in [23]. The paper proceeds to discuss options for validating the M2M terminal state, and advocates a semi-autonomous two-step approach in which internal validation is followed by remote validation by the replying party (e.g., a cellular base station). However, the reverse problem, i.e., the validation of the network by the M2M terminal, is not addressed. Security issues are also discussed in the context of data aggregation [24] but the proposed approach is evaluated only through a proof-of-concept simulation in the context of an 802.15.4-based sensor network.

4.1 Analysis of Security and Privacy Risks

The first step towards a comprehensive security solution in the capillary M2M environment will be a detailed analysis of security and privacy risks in M2M systems. Typical scenarios will be analyzed, for both stationary and mobile M2M, in application areas such as metering, e-health, payment, tracking and tracing, remote control, and safety [2].

4.2 Design and Evaluation of a Compromise-Resilient Architecture for M2M Terminals

The fact that M2M terminals are expected to operate unattended for extended periods of time (and sometimes in an insecure public space) renders them vulnerable to node compromise attacks [25]. Such attacks can be used to launch various internal attacks such as false data injection, selective forwarding, wormhole and Sybil attacks, and thus degrade the performance of the M2M system [21,22]. Therefore, a novel, compromise-resilient architecture is needed, with adequate intrusion and attack detection capabilities which are the critical success factors for M2M technology. The architecture might also include a scheme, similar to neighborhood watch, in which each terminal in capillary M2M will co-operate with its adjacent neighbor devices by establishing a secure communication channel with them, using which they can monitor each other's state, and thus be able to detect node compromise attempts in a timely fashion [26].

4.3 Mutual Authentication of M2M Terminals and M2MG

M2M terminals must be authenticated by the M2MG before they are allowed to attach to the M2MG and, ultimately, to the cellular network; authentication might be performed separately by the network. At the same time, authentication is needed in the reverse direction – M2M terminals must authenticate the network, which will prevent security attacks from bogus terminals and/or bogus networks. Authentication is thus mutual, and must be re-done in regular intervals so as to reduce the likelihood of an adversary capturing and subverting the terminal. For mobile terminals, re-authentication may also be needed in case of handoff to an area controlled by another M2MG. The choice of best solution for mutual authentication and re-authentication will depend on the identity solutions for M2M terminals, e.g., whether an USIM is used or not, and a careful analysis of the tradeoffs incurred by those options is needed. A viable approach seems to be a Rabin-type cryptosystem [27], an asymmetric encryption algorithm in which encryption (or signature verification) operation is extremely fast, while the decryption (or signature) operation is comparably slow and requires a large amount of computation effort [28]. In this manner, the cryptographic burden may be shared between different parties in proportion with their computational capabilities and energy capacity – aspects in which M2M terminals are inferior to the M2MG and other components in the signal path.

4.4 End-to-End Security Protocols for M2M Traffic

In the radio access network (E-UTRAN), all data is sent encrypted [3]; we will analyze the applicability of this scheme to M2M traffic. We will also design and evaluate suitable secure aggregation protocol or protocols, using concealed data aggregation [29], a variation of in-network aggregation in which the aggregation is performed on encrypted values, and only the sink can decrypt the result; thus reducing the number of points along the message path at which an adversary can reach the cleartext message. CDA is based on privacy homomorphism property, namely, an encryption algorithm $E(\cdot)$ is homomorphic if, given $E(x)$ and $E(y)$ and a combination operation *, one can obtain $E(x * y)$ without having to obtain x and y first (i.e., without decryption) [30].

4.5 Security Support for M2M Payment Applications

Payment systems such as point of sale (POS) terminals, automated teller machines (ATMs), and vending machines are an important use case for M2M technology [31]. To support this use of M2M technology, payment service providers may team with MNOs and trusted third parties such as banks, and form a trusted domain that will allow registered M2M terminals to use the service. Later, service provider will be billed by the MNO for the amount of network traffic, while the owner of the M2M terminal will be billed for the payment made in this way. In order to make this scenario a reality, security concerns must be solved. This includes the following: first, mutual authentication between M2M application and M2M terminal, as well as between MNO and M2M terminal; second, message exchanges between M2M application and M2M terminal must be protected; and third, in some cases the identity of the payer must be concealed from the payee and, possibly, from the bank. The problem may further be compounded by the fact that some M2M terminals are mobile; as a result, payments may need to be made during the time that the terminal is roaming. We need to develop a universal authentication and billing architecture, leveraging digital cash (or E-cash) [32] and one-way hash [33] techniques, that will enable M2M terminals to communicate with their subscribed M2M application while roaming throughout the world.

5 Conclusion

This proposal addresses some of the modifications that would enable 4G cellular network such as LTE to support M2M traffic. Cellular M2M approach aims to leverage the significant benefits of cellular networks: virtually ubiquitous coverage, reliable delivery and delay guarantees, and well developed security and charging solutions. It is worth noting that a recent EU FP7 project named EXALTED has identified similar issues with capillary M2M systems.

However, serious challenges need to be overcome before they can be used for M2M traffic. The changes described in this paper appear well positioned to address those challenges. Our future work will focus on investigating the proposed solutions in greater detail, through system design, theoretical analysis, and extensive simulations. An important part of this work will be the implementation of a testbed that will incorporate all the developed protocols and subsystems, including M2MF, MTSS, charging and security solutions.

References

1. The Global Wireless MTC Market, 2nd edn. Berg Insight, Gothenburg (December 2009)
2. Service requirements for machine-type communications (MTC); stage 1, release 11. Technical Report TR 22.368 V11.0.1, 3GPP, Sophia Antipolis, France (February 2011)

3. Palat, S., Godin, P.: The LTE Network Architecture: A comprehensive tutorial. In: Sesia, S., Toufik, I., Baker, M. (eds.) The UMTS Long Term Evolution: From Theory to Practice. John Wiley & Sons (2009)
4. Andrews, J.G., Ghosh, A., Muhamed, R.: Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice-Hall (2007)
5. ETSI Workshop on Machine to Machine (M2M) Standardization, Sophia-Antipolis, France (June 2008), `http://www.etsi.org/website/newsandevents/2008_m2mworkshop.aspx`
6. 1st ETSI TC Machine to Machine (M2M) Workshop, Sophia-Antipolis, France (October 2010), `http://www.etsi.org/website/newsandevents/past_events/2010_m2mworkshop.aspx`
7. Ortiz, J., Culler, D.: Exploring Diversity: Evaluating the Cost of Frequency Diversity in Communication and Routing. In: ACM SenSys, Raleigh, NC (November 2008)
8. Thonet, G., Allard-Jacquin, P., Colle, P.: ZigBee – WiFi Coexistence. white paper and test report, Schneider Electric, Grenoble, France (2008)
9. Mišić, J., Mišić, V.B.: Wireless personal area networks: performance, interconnections and security with IEEE 802.15.4. John Wiley & Sons, Chichester (2008)
10. IEEE 802.15 WPAN^{TM} Task Group 4e (TG4e) report at, `http://www.ieee802.org/15/pub/TG4e.html`
11. Castillo, J.: The survival of communications in ad hoc and M2M networks: The fundamentals design of architecture and radio technologies used for low-power communication NOMOHI devices. In: ITSim 2010, vol. 2, pp. 666–671 (2010)
12. Jung, K., Park, A., Lee, S.: Machine-Type-Communication (MTC) Device Grouping Algorithm for Congestion Avoidance of MTC Oriented LTE Network. In: Kim, T.-h., Stoica, A., Chang, R.-S. (eds.) SUComS 2010. CCIS, vol. 78, pp. 167–178. Springer, Heidelberg (2010)
13. Mišić, J., Mišić, V.B.: Performance modeling and analysis of Bluetooth networks: polling, scheduling, and traffic control. CRC Press, Boca Raton (2006)
14. Mišić, J., Mišić, V.B.: Characterization of idle periods in IEEE 802.11e networks. In: IEEE WCNC, Cancun (2011)
15. Cordeiro, C., Challapali, K., Birru, D., Sai Shankar, N.: IEEE 802.22: the first worldwide wireless standard based on cognitive radios. In: New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005, November 8-11, pp. 328–337 (2005)
16. Mišić, J., et al.: Maintaining Reliability through Activity Management in 802.15.4 Sensor Clusters. IEEE Transactions on Vehicular Technology 55(3), 779–788 (2006)
17. Yaiparoj, S., et al.: On the economics of GPRS networks with Wi-Fi integration. European Journal of Operational Research 187(3), 1459–1475 (2008)
18. Sou, S.I., et al.: Modeling credit reservation procedure for UMTS online charging system. IEEE Transactions on Wireless Communications 6(11), 4129–4135 (2007)
19. de Laat, C., Gross, G., Gommans, L.: Generic AAA architecture. Internet Engineering Task Force Network Working Group, Request for Comment (RFC) 2903 (2000)
20. Recommendation X.805 Security architecture for systems providing end to end communications. ITU-T Lead Study Group on Telecommunication Security (October 2003)
21. Feasibility Study on the Security Aspects of Remote Provisioning and Change of Subscription for M2M equipment; release 9. Technical Report TR 33.812 V1.4.0, 3GPP, Sophia Antipolis, France (June 2009)
22. Du, J., Chao, S.W.: A study of information security for M2M of IOT. In: ICACTE 2010, vol. 3, pp. 576–579 (August 2010)

23. Cha, I., et al.: Trust in M2M communication. IEEE Veh. Technol. Magazine 4(3), 69–75 (2009)
24. Bartoli, A., et al.: Secure lossless aggregation for smart grid MTC networks. In: SmartGridComm 2010, pp. 333–338 (2010)
25. Hartung, C., Balasalle, J., Han, R.: Node compromise in sensor networks: the need for secure systems. Technical Report CU-CS-990-05, Dept. of Comp Sci., Univ. of Colorado at Boulder (January 2005)
26. Lin, X.: CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks. In: IEEE Global Communications Conference, GLOBECOM 2009, Honolulu, HI (2009)
27. Rabin, M.O.: Digital signature and public-key functions as intractable as factorization. MIT Laboratory of Computer Science. Technical Report. MIT/LCS/TR-212 (January 1979)
28. Gaubatz, G., Kaps, J.-P., Sunar, B.: Public Key Cryptography in Sensor Networks—Revisited. In: Castelluccia, C., Hartenstein, H., Paar, C., Westhoff, D. (eds.) ESAS 2004. LNCS, vol. 3313, pp. 2–18. Springer, Heidelberg (2005)
29. Peter, S., Piotrowski, K., Langendoerfer, P.: On concealed data aggregation for WSNs. In: IEEE Consumer Communications and Networking Conference, pp. 192–196 (2007)
30. Rivest, R.L., et al.: On data banks and privacy homomorphisms. In: DeMillo, R., Dobkin, D., Jones, A., Lipton, R. (eds.) Foundations of Secure Computation, pp. 169–180. Academic Press (1978)
31. Study on facilitating machine to machine communication in 3GPP systems, release 8. Technical Report TR 22.868 V8.0.0, 3GPP, Sophia Antipolis, France (March 2007)
32. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology - Crypto 1982, pp. 199–203. Springer (1983)
33. Lamport, L.: Password authentication with insecure communication. Commun. of the ACM 24(11), 770–772 (1981)
34. EXALTED (EXpAnding LTE for Devices), Integrated Project of the European Union's Seventh Framework Programme, http://www.ict-exalted.eu/

Providing QoS in the Integration of RFID and Wi-Fi WLAN

Nargis Khan[1], Jelena Mišić[1], Vojislav B. Mišić[1], and Lutful Karim[2]

[1] Ryerson University, Toronto, ON, Canada
[2] University of Guelph, ON, Canada

Abstract. Radio Frequency Identification (RFID) systems are widely used in various application domains. However, when RFID networks are integrated with WiFi WLANs to extend their short transmission range, interference results due to the use of the same ISM frequency band, and some co-existence mechanism must be found. In this paper, we propose a time-sharing mechanism that allows RFID networks using IEEE 802.15.4 networks to connect to a WLAN that uses IEEE 802.11e EDCA for ensuring priority access for RFID data. To reduce collisions among RFID tags, we use an energy management mechanism based on randomized sleep. Simulation results show that the proposed solution achieves Quality of Service (QoS) by maintaining higher throughput and lower collision probability.

Keywords: RFID, Wi-Fi WLAN, QoS, Collision Probability, Throughput, Integration, Co-existence.

1 Introduction

RFID systems are widely used in many application areas such as manufacturing, health care, public transportation, telecommunications, and logistics and are considered as an alternative of barcode system in the distribution industry and access control. RFID technology is simple and easy to use, but RFID tags have very short transmission range and thus can't be used in many applications. To overcome this limitation, RFID systems are integrated with other networks, most often with wireless local area networks (WLANs) using some variety of the ubiquitous IEEE 802.11 standard [1]; in this solution, the bridging function is typically provided by the RFID readers (stationary or mobile) which are equipped with both the proper interface to connect to RFID tags and IEEE 802.11 interface to connect to the WLAN. In many cases, RFID tags are active and use the well known IEEE 802.15.4 low rate WPAN protocol [2]. Since several of IEEE 802.11 systems work in the same Industrial, Scientific and Medical (ISM) band at 2.4GHz as do IEEE 802.15.4-based networks, their channels overlap, as shown in Fig. 1, and problems arise due to interference. In fact, it has been shown that, under some circumstances, channel overlap may lead to as much as 90% of WPAN frames (!) being destroyed by the interfering WLAN frames [4]. The problem is aggravated by the difference in transmission power; namely, IEEE 802.11 systems typically use much higher power and are thus able to harm RFID transmissions at longer distances. In this paper, we propose a simple solution that eliminates the aforementioned problems and allows RFID systems using IEEE 802.15.4 standard to co-exist with WiFi WLANs. The main features of the solution are as follows.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 424–434, 2012.

- Interference is not dependent on channel selection. Instead, it relies on time multiplexing of RFID and WLAN networks, using the Point Coordination Function (PCF) mechanism [1]. The WLAN Access Point (AP) is aware of the neighboring RFID networks at the Medium Access Control (MAC) layer. RFID readers provide the bridging function.
- To facilitate reliable transfer of data from RFID tags, IEEE 802.11e prioritizing scheme based on Enhanced Distributed Channel Access (EDCA) is used. In this case, different priorities are assigned to the WLAN nodes and RFID readers, with the latter being assigned higher priority to ensure the specified Quality of Service (QoS) for RFID data.
- To reduce collisions among RFID tags when accessing RFID readers, and to prolong the lifetime of RFID tags, an energy management scheme based on randomized sleep management of individual tags is used.
- The proposed solution can easily be adapted to two tier networks with arbitrary number of RFID networks and arbitrary number of tags in each of them.

The performance of the proposed framework, expressed through network throughput and collision probability, is evaluated through discrete event simulation.

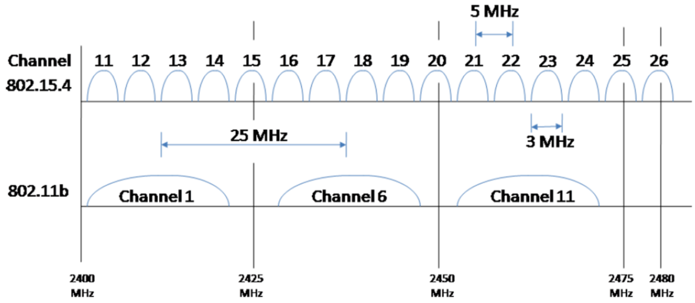


Fig. 1. The IEEE 802.11 and 802.15.4 spectrum usage (from [3])

The rest of the paper is organized as follows. Section 2 presents several existing approaches that solve the interference and co-existence problems that arise when an RFID network is integrated with a WLAN network. In Section 3, we present the proposed network architecture and the associated time-sharing solution that provides QoS while integrating RFID with WLAN network. Section 4 presents the simulation model and results. Finally, Section 5 concludes the paper and outlines the directions for future work.

2 Literature Review

In this section, we present existing approaches that are related to the co-existence and interference problems between WLAN and active RFID networks. The simplest such approach relies on RFID readers integrated with the WLAN access point (AP) [5,6]. However, this is not efficient because it severely limits the RFID network coverage – namely, the short transmission range of RFID tags necessitates that all tags must be very close to the AP.

A number of proposals attempt to use different modulation and channel access schemes such as Frequency Division Multiple Access (FDMA) mechanism [7] or integration of Direct Sequence Spread Spectrum (DSSS) and Code Division Multiple Access (CDMA) [8]. The main problems with such approaches are the scarcity of independent frequencies and codes, respectively, and the necessity to statically allocate those frequencies and codes to tags beforehand. As a result, the schemes are unsuitable for applications with a large number of tags; they are also not well suited to scenarios and/or applications in which a large number of tags can dynamically join or leave the domain.

Some of schemes [7] rely on exclusive use of IEEE 802.15.4 channel 26, see Fig. 1, which is least susceptible to interference from WLAN communications, as the control channel. However, in large scale applications the use of a single control channel jeopardizes reliability of the network: first, because other communication networks could choose the same channel, and second, because the sheer number of nodes that attempt to use that same channel increases the risk of congestion. As a result, this approach can't guarantee the performance required by such applications.

As noted above, interference between 802.15.4 and WLAN networks is usually more damaging to the former. However, [9] demonstrates that 802.15.4 interference has more effect on the uplink communication in a WLAN than on the corresponding downlink communication. Also, the work reported in [10] shows that conflict among channels increases for increasing the number of 802.11b networks.

It is well known that the IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol does not support QoS guarantees for performance indicators such as throughput, delay, and delay jitter, and that the collision probability increases at heavy traffic load. This is the main rationale for attempts to modify the IEEE 802.11 MAC protocol in order to provide priority-based differentiation and, ultimately, QoS support. These attempts have been standardized through the IEEE 802.11e standard that incorporates facilities such as EDCA, which provides different traffic classes with different priorities, and TXOP, which improves efficiency by allowing extended bandwidth allocation. Several performance analyses of the EDCA and TXOP mechanisms have been reported so far [11,12,14], including the coexistence of traditional DCF and EDCA. In general, EDCA has been shown to work better under heavy load than the traditional DCF, mainly because of prioritization achieved through shorter interframe space intervals (AIFS), and are thus able to provide superior performance for QoS-demanding applications.

3 Proposed Solution

3.1 Time-Sharing Co-existence Mechanism

As outlined above, multiple access schemes based on FDMA or CDMA are unable to provide the desired performance and scalability, hence our approach uses a time-division multiplexing approach provided by the Point Coordination Function (PCF) mechanism defined as part of the IEEE 802.11 standard [1]. The PCF-based frame structure that allows the co-existence between the two networks is illustrated in Fig. 2. In this frame structure, the IEEE 802.11 superframe period is interleaved with the IEEE 802.15.4 superframe as follows. At the beginning, the AP periodically broadcasts beacon frames containing information about the duration of the contention-free (CFP) and contention access (CP) periods, which are used to access the medium by RFID and WLAN networks, respectively. During the active portion of the 802.15.4 superframe (which coincides with the contention-free period in the 802.11 network), RFID reader collects data from RFID tags in the 802.15.4 network, while other WLAN nodes are silent as no polling by the AP takes place. An inactive period in the 802.15.4 network starts when a CF-End frame is received at the end of the CFP period; RFID tags may then go to sleep as will be explained below. However, RFID reader switches to its WLAN interface and competes with other WLAN nodes in order to send RFID data to the AP, which will then aggregate the data and forward them tot he application server. After the CP period ends, a beacon frame is sent again that starts the new WLAN superframe.

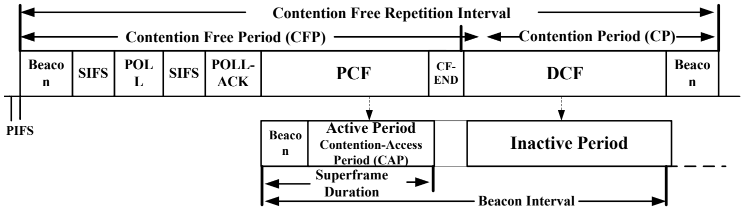


Fig. 2. The frame structure of IEEE 802.11 and 802.15.4 standard

Using this mechanism we can define a two-tiered network architecture which is shown schematically in Fig. 3. In this architecture, the first tier is the RFID network using IEEE 802.15.4 standard wherein RFID tags send data to RFID readers during the active portion of the 802.15.4 superframe. In the second tier, which is a WLAN using IEEE 802.11 with EDCA, RFID readers work as WLAN nodes to deliver their data to the AP. To ensure the QoS guarantees for RFID data, RFID readers are assigned higher priority than ordinary WLAN nodes that may be active in the same coverage area, using the EDCA mechanism.

It is worth noting that this architecture allows for several RFID networks to be integrated with a single WLAN, due to the following. First, the short transmission range of IEEE 802.15.4 networks allows several such networks to co-exist within the coverage area of a single WLAN network. Second, since the WLAN is effectively inactive during the CFP, several IEEE 802.15.4 networks may co-exist in the same space without interference from each other if they use different channels (the 802.15.4 standard allows 16 such channels). Third, collocated IEEE 802.15.4 networks may even work on the same channel, provided the CFP period is long enough to accommodate the active portions of their superframes, shifted in time to avoid any overlap.

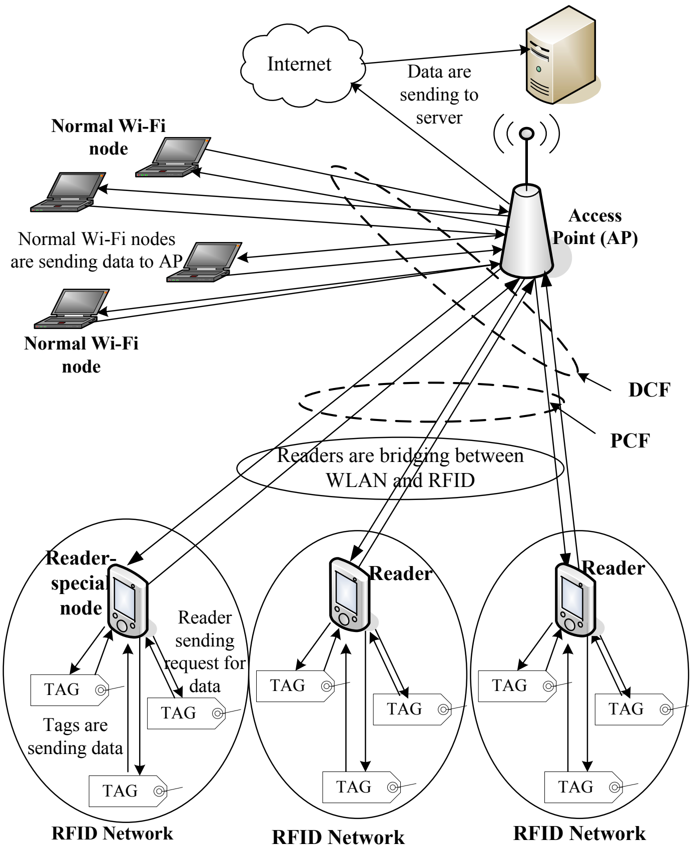


Fig. 3. The proposed two-tiered network architecture

The duration of the active portion of an IEEE 802.15.4 superframe is referred to as the superframe duration, *SD*, while the total duration of the superframe is referred to as the beacon interval, *BI* [2]. The values of these two intervals are determined by two important MAC parameters known as beacon order, *BO*, and superframe order, *SO*, where $0 \leq SO \leq BO \leq 15$, through the following formulas:

$$SD = 2^{SO} \tag{1}$$

$$BI = 2^{BO} \tag{2}$$

both of which are expressed in unit backoff periods of 320μs. The actual calculation of the beacon order and superframe order is performed using Algorithm 1 shown below; the 'actual number of RFID networks' refers to the number of such networks that use the same 802.15.4 channel.

The proposed framework achieves QoS using the IEEE 802.11e EDCA priority assignments (during the contention period) to the nodes when they send data to the AP. (As is well known, plain 802.11b-style DCF does not provide QoS support.) EDCA

Algorithm 1. Calculating the *BO* of RFID Network when the number of RFID networks is known

```
Input: $BI(WiFi)$, $SD$, initial number of RFID Networks
Output: $BO(RFID)$, $MaxNoOfRFIDNetworks$
Assume,
    $n = numberofRFIDNetworks \leftarrow 1$
    $BO(RFID) \leftarrow 0$
    $SO \leftarrow 0$
    $SD \leftarrow aBaseSuperframeDuration \times 2^{S0}$
    $BI(WiFi) \leftarrow 100 timeunit$
    $SD \times 2^{BO(WiFi)} \leftarrow 100 timeunit$
    $BO(WiFi) \leftarrow SD \times 2^{BO(WiFi)}$
    $\Rightarrow BO(WiFi) \leftarrow \lceil \log_2(\frac{BI(WiFi)}{SD}) \rceil$
while $BO(RFID) \leq 14$
    $BI(RFID) \leftarrow 2^{\alpha} \times BI(WiFi)$ where $\alpha \leftarrow \lceil \log_2 n \rceil$
    $BI(RFID) \leftarrow 2^{\alpha} \times 2^{BI(WiFi)} \times SD$
    $BI(RFID) \leftarrow 2^{\alpha + BO(WiFi)} \times SD$
    Again, $BI(RFID) \leftarrow 2^{BO} \times SD$
    $2^{BO(RFID)} \times SD = 2^{\alpha + BO(WiFi)} \times SD$
    $BO(RFID) = \alpha + BO(WiFi)$
    $n \leftarrow n + 1$
end while
$MaxNoOfRFIDNetworks \leftarrow n$
```

provides four access categories with different priorities that are distinguished through different values of parameters such as AIFSN and contention window (CW). For simplicity, we categorize WLAN nodes into two types, RFID readers (acting as bridges) and ordinary nodes. Data from RFID readers have the two highest access categories, AC[3] and AC[2], which ensures that the data sent by a RFID reader will not be affected by transmissions from ordinary nodes. On the other hand, ordinary WLAN nodes are assigned the two lower priority access categories, AC[1] and AC[0].

It is worth noting that the payload of a data frame sent from a RFID tag is much smaller than that sent by an ordinary WLAN node. However, the RFID reader will aggregate a number of such small packets into a larger data packet, therefore the payload of WLAN data frames will not depend much on whether it's been sent from a RFID reader or an ordinary WLAN node.

3.2 More on Medium Access

In the first tier, active RFID tags send data to RFID readers using the IEEE 802.15.4 slotted CSMA-CA medium access mechanism. Whenever tags receive beacon from the polled readers, CAP/active period of the IEEE 802.15.4 superframe starts working. If a tag has data to send it sets the initial parameters value (Retry Count, Contention Window and Backoff exponent) and starts to Backoff Countdown (BC). When the value of BC becomes zero the tag requires to check if the current superframe has sufficient time to send the packet. Otherwise, the tag need to wait until the active period of next superframe. After that the tag needs to sense the state of the medium (idle) in two consecutive slots (i.e. performing two clear channel assessments (CCA)). If the first CCA was successful the tag performs the second CCA. If the medium is found free in both CCAs the tag can transmit the packet, and wait for the acknowledgement. Transmission occurs only at the boundary of the slot. Hence, if packet arrives in the middle of the slot the tags need to wait for the boundary of the slot. A CF-END frame is sent to tags that indicates the end of active period. At the end of active period, the inactive period starts when the tags remain in sleep mode. Tags go to sleep mode to save energy after the PCF and remain in sleep mode based on the sleeping probability, P_{sp}. Thus, tags are mostly in sleep state and check beacon messages after they wake-up.

At the end of PCF, DCF period starts. In DCF, all ordinary WLAN stations and RFID readers compete for the medium to transmit data packets to AP. Data packets are transferred only when the medium is found idle following the IEEE 802.11 DCF medium access mechanism. During the uplink communication of WLAN, data are transferred from a station to AP but not vice versa. EDCA priorities are assigned to the nodes to transmit their data to the AP. CSMA-CA algorithm and DCF bandwidth reservation technique is used in the CP period. If a station has data to transmit it checks the status of medium and also whether it is CP or CFP. If the station senses the medium idle for the duration of inter-frame space of time (AIFS) in the active period of WLAN mode, the station generates a random number. The random number/backoff is decremented if the station receives idle back-off value (i.e., idle medium). Whenever the back-off value reaches zero the station transmits if there is any packet in the queue. Before sending the data packet the station sends the RTS packet and waits to receive CTS packet from the AP. The station sends data packet to AP after receiving CTS. If the AP receives data successfully it replies with an ACK packet that notifies the end of current transmission. At the end of CP period, the WLAN station sets its NAV value according to the period of PCF.

3.3 Energy Management

Due to the EDCA mechanism, there will be few (if any) collisions in the WLAN network. We are referring to collisions of packets sent from RFID readers, on one side, and those sent from ordinary WLAN nodes, on the other; however, there may be collisions between packets sent from different nodes of the same type. At the same time, collisions are possible in the IEEE 802.15.4 network since the standard does not provide any priority and all RFID tags can attempt to access the medium in the same manner. We refer to those collisions as internal collisions in the 802.15.4 network.

To control and, if possible, eliminate such internal collisions and, at the same time, ensure longest possible lifetime for RFID tags which are expected to operate on battery power, we propose an energy management mechanism. Namely, RFID tags need not be awake all the time; they can wake up periodically, send their data, and go back to sleep again. We assume that the sleeping period of the tags is a random variable that follows the geometric distribution; the mean value of the distribution can be adjusted through a single parameter, denoted as P_{sp} (sleeping probability). The corresponding probability generating function (PGF) [15] may be expressed as

$$T_s(z) = \sum_{k=1}^{\infty} (1-P_{sp})P_{sp}^{k-1}z^k = \frac{(1-P_{sp})z}{1-zP_{sp}} \tag{3}$$

while the mean value of sleeping time, $\overline{T_s}$, is calculated as

$$\overline{T_s} = T_s'(1) = \frac{1}{1-P_{sp}} \tag{4}$$

This mean time is expressed in seconds, but a more convenient value is the desired sleeping time in unit backoff periods of 320μs:

$$\overline{T_x} = \frac{\overline{T_s}}{0.00032} \tag{5}$$

Once we know the desired sleeping time in unit backoff periods, we can calculate the mean sleeping time and the sleeping probability, P_{sp}, using (4) and (5). For example, if the desired sleeping time is one hour, the sleeping probability P_{sp} will be 0.999999911.

4 Performance Evaluation

To evaluate the performance of the proposed approach, we have built the simulator of the two-tiered network using the Petri net-based object-oriented simulation engine Artifex by RSoftDesign, Inc. [16].

For the first tier, the IEEE 802.15.4 network, we have assumed the presence of three separate RFID networks, each with a single reader and a number of active tags. The networks operate in the ISM band at 2.4GHz, with raw data rate of 250kbps. The values of superframe order and beacon order are set to $SO = 0$ and $BO = 5$, respectively, which allows up to four such networks to work with a single WLAN in the second tier, using the time-sharing scheme. We assume that each RFID tag has a packet to transmit whenever it wakes up from the sleeping mode. Each tag sleeps for a random time interval, which is obtained as a random value derived from the geometric probability distribution with the parameter P_{sp}. Since sleep times are random, the probability of several tags being awake at the same time is small. In this manner, internal collisions in each of the networks at the first tier are minimized. Note that there can be no collision between the transmissions from different first-tier networks since the active portions of their superframes do not overlap.

4.1 First Tier: The RFID Network

We measure the performance of first-tier networks in terms of collision probability, defined as the ratio of the total collided transmissions over the total transmitted packets. We have conducted two experiments: in the first one, we have varied the average sleeping time (1, 5, 7, 10, and 15 minutes) for a fixed number of tags (80 in each network); in the second, we have varied the number of tags from 20 to 140, in increments of 20, for a fixed average sleeping time of 1 minute. The results are shown in Figs. 4 and 5.

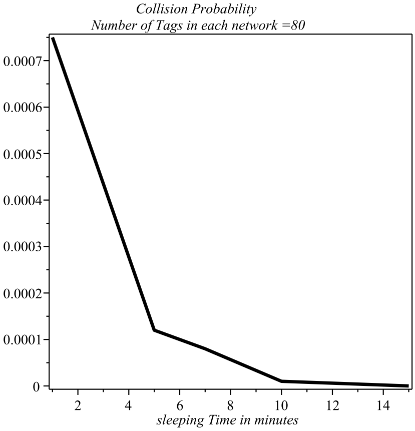


Fig. 4. Collision probability of RFID network over the average sleeping time of tags

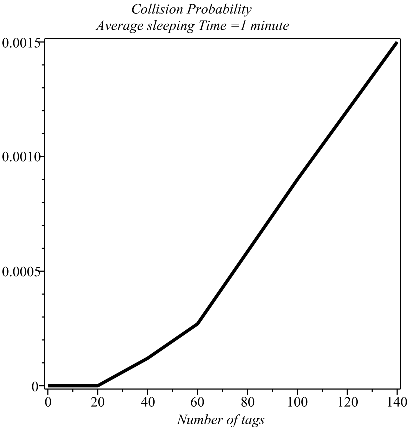


Fig. 5. Collision probability of RFID network over the number of tags

As can be seen from Fig. 4, collision probability decreases with an increase in average sleep time, because longer sleep times reduce the probability that two or more RFID tags will be awake simultaneously. Note that the collision probability can't be totally eliminated since randomization of sleep intervals does not guarantee that the number of active tags will not exceed one at any given moment.

Collision probability also depends on the number of tags in a first-tier network; as can be seen from Fig.5, it increases when the number of tags increases: from nearly zero at 20 tags, to over 0.1% when the number of tags exceeds 110. While this value is not terribly high, the shape of the curve indicates that larger first-tier networks do run the risk of being very much collision-prone; this will lead to deterioration of network throughput and shortening of useful network lifetime.

4.2 Second Tier: WLAN

We measure the performance of the WLAN network in terms of throughput and collision probability. To this end, the number of ordinary WLAN nodes is varied between 4 and 28, while the WLAN also contains the three RFID readers that provide the bridging to first-tier RFID networks. We set the data arrival rate to 40kbps per ordinary node, mean tag sleeping time to one minute, and set the number of RFID tags per first-tier network to 150. Traffic from ordinary WLAN nodes is assigned to lower priority traffic classes, equally split between AC[0] and AC[1] (in EDCA terminology, background and best-effort, respectively), while the traffic from RFID readers is equally split between higher priority, AC[2] and AC[3] (voice and video, respectively). Since data coming from a single RFID tag is of low volume compared to that coming from a WLAN node, we aggregate RFID data from a number of tags up to 100 bytes per packet so that the readers send a larger data packet and thus increase bandwidth utilization and overall network throughput.

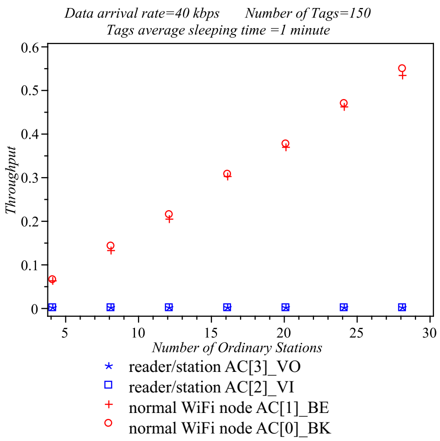


Fig. 6. Throughput of WLAN networks

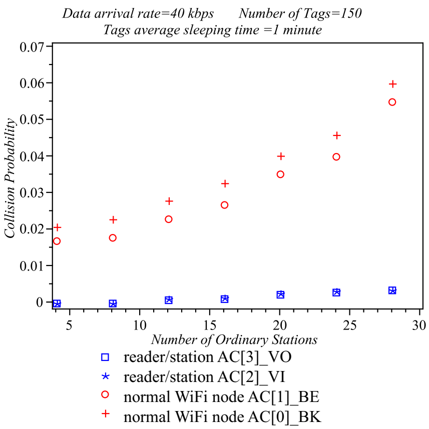


Fig. 7. Collision probability of WLAN networks

As can be seen from Fig. 6, the total throughput of the WLAN increases when the number of ordinary WLAN nodes increases. As expected, the throughput of the RFID readers does not depend on the number of ordinary WLAN nodes, since the EDCA mechanism effectively protects the RFID reader traffic through shorter AIFS intervals and smaller contention window sizes. However, collision probability shown in Fig. 7 increases with the number of ordinary WLAN nodes; it is again very small for the traffic from RFID readers which is given higher priority through EDCA.

5 Conclusion and Future Work

In this paper, we have described a two-tier framework to facilitate the integration of RFID networks using the IEEE 802.15.4 standard with WLAN networks using the IEEE 802.11e standard with EDCA. The framework uses time-sharing to allow for interference-free co-existence between the two technologies which operate in the same ISM band at 2.4GHz. To reduce the probability of collisions in the first, RFID tier, we employ randomized sleep management; to reduce collisions in the second, WLAN tier, we assign higher priority to aggregated traffic from RFID readers. Out results indicate that the framework gives satisfactory results in maintaining low collision probability at both tiers.

Our future work will focus on fine tuning of the mechanism and on the strategies for dynamically adjusting nthe parameters of the network depending on the number of nodes; we will also investigate the possibility of employing mobile readers to improve performance.

References

1. IEEE 802.11. IEEE Sstandard for Iinformation Ttechnology Ttelecommunications and information exchange between systems local and metropolitan area networks specific requirements-Part 11: Wtireless LAN Mmedium Aaccess Ccontrol (MAC) and Pphysical LAyer (SHY) Spacifications. IEEE (2007)
2. IEEE 802.15.4 Sstandard for part IEEE 802.15.4. Wtireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN) IEEE Std 802.15.4. IEEE (2003)
3. Ortiz, J., Culler, D.: Exploring Diversity: Evaluating the Cost of Frequency Diversity in Communication and Routing. In: Proc. ACM SenSys, Raleigh, NC (2008)
4. Sikora, A., Groza, V.F.: Coexistence of IEEE 802.15.4 with other systems in the 2.4 GHz ISM-Band. In: Proc. IEEE Instrumentation and Measurement Technology Conference, IMTC 2005, vol. 3, pp. 1786–1791 (May 2005)
5. Arada Systems LocAir, Active RFID over WLAN, http://www.aradasystems.com/ProductDataSheets/11arada_datasheet_locair.pdf
6. AeroScout WiFi RFID, http://www.aeroscout.com/content/wi-fi-rfid
7. Thonet, G., Allard-Jacquin, P., Colle, P.: ZigBee-WiFi Coexistence. white paper and test report, Schneider Electric, Grenoble, France (2008)
8. Rusch, L.A.: Indoor Wireless Communications: Capacity and Coexistence on the Unlicensed Bands. Intel Technology Journal (Third Quarter 2001)
9. Shuaib, K., Boulmalf, M., Sallabi, F., Lakas, A.: Co-existence of Zigbee and WLAN, a performance study. In: Proc. Wireless Telecommunications Symposium, WTS 2006, pp. 1–6 (April 2006)
10. Tamilselvan, G.M., Shanmugam, A.: Probability of channel collision and per analysis of coexistence heterogeneous networks for various topologies. In: International Conference on Control, Automation, Communication and Energy Conservation, INCACEC 2009, pp. 1–7 (June 2009)
11. Alahmadi, A.A., Madkour, M.A.: Performance evaluation of the IEEE 802.11e EDCA access method. In: International Conference on Innovations in Information Technology, IIT (December 2008)
12. Hwang, G.H., Cho, D.H.: Performance analysis on coexistence of EDCA and legacy DCF stations in IEEE 802.11 wireless LANs. IEEE Transactions on Wireless Communications 5(12), 3355–3359 (2006)
13. Ranjit, K., Srinivas, R., Rajesh, I.: QoS and interoperability issues of 802.11 and 802.11e. Technical report. KReSIT, Indian Institute of Technology Bombay (2006)
14. Rashwand, S., Mišić, J.: Stable operation of IEEE 802.11e EDCA: Interaction between offered load and MAC parameters. Ad Hoc Networks 10, 162–173 (2012)
15. Mišić, J., Mišić, V.B.: Wireless Personal Area Networks: Performance, Interconnection, and Security with IEEE 802.15.4. John Wiley and Sons (2008)
16. RSoft Design. Artifex V.4.4.2. RSoft Design Group, Inc. (2003)

Distributed Distance Sensitive iMesh Based Service Discovery in Dense WSAN

Milan Lukic and Ivan Mezei

Faculty of Techhnical Sciences, University of Novi Sad,
Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia
{milan_lukic,imezei}@uns.ac.rs

Abstract. We investigated performance of localized distance-sensitive service discovery algorithm *iMesh*, which generates information structure in static network, to store the information about nearby actors (service providers). In a network with grid structure, this is achieved by advertising service provider positions in four geographical directions. The propagation of information about remote service providers is restricted by a blocking rule, to reduce the message overhead and provide distance sensitivity. A node requiring service (service consumer) conducts lookup process to obtain information about nearby service providers from the *iMesh* structure. We modified *iMesh* to enable its use in dense networks with topologies other than grid by introducing *iMesh* areas instead of *iMesh* edges. Our simulations compare performance of modified *iMesh* with another localized service discovery scheme (quorum) in dense networks with random topologies. We show that *iMesh* finds the nearest service provider in >95% of cases. It significantly decreases the message overhead compared to quorum, without compromising quality of service discovery.

Keywords: service discovery, localized algorithm, wireless sensor and actor networks, iMesh.

1 Introduction

A typical wireless sensor and actor network (WSAN) consists of relatively large number of small low-cost static sensor nodes which monitor some of the environment parameters (e.g. temperature, humidity, etc.) and usually an order of magnitude smaller number of resource-rich mobile actors. The usual scenario is such that static nodes monitor and detect critical changes of parameters of interest, a situation which will further be referred to as event. Whenever such an event is detected by a sensor node, an appropriate action needs to be taken by a mobile actor device (e.g. robot or unmanned aerial vehicle). A sensing task is performed in distributed manner, which means that there is no central network entity that collects all the data, and the decisions are made locally. There are numerous applications of such distributed tasks in habitat or agricultural monitoring, home automation, object or vehicle surveillance, fire prevention and area exploration,

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 435–448, 2012.

just to name a few. For example, in fire prevention application, a number of static nodes are deployed in a forest environment. The event of interest might be extreme change of temperature caused by fire, which triggers *service discovery* process. Wireless sensor-sensor and sensor-actor communication is used to locate and send service request to a nearby actor, which is equipped with fire extinguisher. Upon reception of service request, the actor moves to event location and performs the necessary action, i.e. extinguishes the fire. Two key issues arise in such an application. First, it is necessary to assure that the nearest service is found to minimize actor movement, and thus energy consumption and response time. This imposes *distance-sensitive* service discovery [1], which implies that service requesting node always finds the closest, or possibly a nearby service. The other issue is to achieve satisfying quality of service discovery with as little communication as possible, which is important for reducing power consumption of battery-powered sensor nodes, and decreasing possibility for message collisions. Both of these issues might be seen as aspects of the same ultimate goal, which is to extend system lifetime by improving energy efficiency. Li et al. [1] introduced a distance-sensitive localized algorithm called iMesh, but it covers only the special case of network topology where static sensors form rectangular grid. In this paper, we propose iMesh extension which can be used in dense networks with random sensor distribution. Algorithm performance in sparse networks, as well as influence of gaps and holes in the network remain as future research direction.

1.1 Problem Statement

We observe the sensing field covered with randomly deployed sensor nodes, due to inability to distribute them in desired manner (e.g. sensors might be deployed from the airplane, or the configuration of terrain dictates the network topology). All of the sensor nodes are aware of their own positions, by using GPS or some other localization system. The mobile actor nodes are also randomly deployed over the sensing area, and their number is significantly smaller than number of sensor nodes. Whenever a static sensor detects some irregularity in sensing values, it is considered as an event which needs to be handled by a mobile actor. We assume that nature and magnitude of occurring events is such that it is enough that each event is serviced by just one actor. The node which senses an event (service consumer) initiates service discovery process. The aim of this process is to locate and send service request to the best (closest) service provider (actor) available. To do so, static nodes communicate with each other and with actors by using single wireless channel. The network is modeled by using unit disk graph (UDG) model. Each network entity can communicate with others that are located within its radio range. Further in the text, we will refer to event-sensing nodes as *service consumers*, and mobile actors as *service providers* (SP).

1.2 Existing Solutions

In the service discovery problem, service providers (sensors, actuators or robots) send location update messages, whereas service consumers (sensors or sink) send search messages to learn latest positions of service providers. The task is to minimize combined update and search message cost, while maximizing success rate of finding target service provider and subsequently routing to it and assigning task. In the literature, many service discovery algorithms have been proposed for mobile ad hoc networks. As an active subject, service discovery has been studied for over a decade in wireless ad hoc networks [5], [6]. Existing solutions can be directly applied to emerging sensor and mobile actuator networks or to wireless sensor and robot networks.

These algorithms can be divided into directory-based and directory-less. Directory-based algorithms use a well structured service directory to store service provider information and facilitate service lookup. They usually require global communication/computation such as clustering and dominating set formation for service directory construction and maintenance. Directory-less algorithms do not maintain any special component but rely on periodical service advertisement and multicasting/anycasting based service lookup. Their execution often involves (limited) flooding operations. Because these existing algorithms may generate large message overhead and/or inconstant per node storage load, they are not suitable for resource constrained WSAN.

Service discovery problem is also known as information brokerage, where the main issue is to exchange information between service consumer and service provider. There are two trivial solutions to the problem. First is to flood the query throughout the network and to wait for responses from service providers. Another trivial scheme is to replicate the service provider data to all nodes in the network, thus enabling service consumers to retrieve information regarding service providers immediately.

All solutions to the given problem are centralized, cluster-based, hierarchical, decentralized, or hybrid. Most existing solutions that use service provider coordination are centralized. One of the service providers (usually actuators, or in some cases sensors), or a central entity, gathers all the information from other actuators and makes a decision. Communication cost for gathering information in case of multi-hop networks is rarely considered. Since usually no details of communication protocols used are given, a complete graph where each service provider is within communication range of any other is assumed indirectly.

Centralized solutions usually define coordination problem as an integer linear programming problem. The main advantage of a centralized solution is that, theoretically, an optimal solution can be found. However, centralized solution features high computation and communication overhead, lack of scalability and slow responsiveness. Moreover, the actual communication cost is usually ignored, especially for large networks. It is further not clear how actuators communicate if the graph is not complete, or if it is disconnected. Centralized solutions also have low fault tolerance if the leader is malfunctioning in any way.

Cluster based hierarchical architecture can be used under certain conditions. Service providers are grouped into clusters where one service provider is chosen to be a cluster head. Lower ranked service providers communicate only to cluster heads, while cluster heads communicate among themselves making decisions. This architecture features good scalability, but also a low tolerance for malfunctioning of a cluster heads.

Decentralized control architectures typically require service providers to take actions based only on the local knowledge. This control approach can be highly robust to failure, since no service provider is responsible for the control of any other service provider. However, achieving global efficiency in these systems can be difficult, because high-level goals have to be incorporated into the local control of each service provider. If the goals change, it may be difficult to revise the behavior of individual service providers. Hybrid control architectures combine local control with higher-level control approaches to achieve robustness, good scalability and the ability to influence actions of the entire team through global goals, plans, or control.

1.3 Steps toward Distributed Distance-Sensitive Solution

A solution of distance-sensitive service discovery problem which always results in finding the closest service provider implies construction of Voronoi diagram with SPs as vertices. Then, the SPs distribute information about them along the edges of their Voronoi polygons. This allows a service consumer to conduct a search in arbitrary direction, and find the closest service when it hits the boundary of its home Voronoi polygon. Although this idea leads to optimal solution, unfortunately it requires global knowledge. In practice, we need to replace the Voronoi diagram with localized planar structure with reasonably good proximity property, which localizes search process to vicinity of service requesting node.

The first idea towards distributed solution is to replace Voronoi diagram with service directory, constructed by well known strip quorum technique [3]. That is, service providers propagate their location along the residing row (eastward and westward). A service requesting node performs search by sending search messages in the orthogonal direction (northward and southward). Their paths cross with strips containing information about service providers, so when the terrain boundary is reached, the information follows the backwards path and returns to node that initiated the search. Although this method requires only local computation, it can generate inconstant storage load on network nodes if service providers are all collinear, and it also makes the localized lookup no longer able to provide the closest/nearby service selection guarantee because the structure bears no proximity property.

Le et al. [1] proposed a modification of Quorum to obtain a planar structure that unlike the Voronoi diagram, can be constructed in a purely localized manner but still possesses required proximity property. The first modification is that service providers propagate their location information in four geographic directions, i.e., north, east, south, and west. The other modification is the use of distance based *blocking rule*, which brings crucial advantage over quorum both in terms of distance sensitivity and number of messages needed to establish information structure. In short, the rule applies whenever two iMesh edges intersect at some node. In such case, the information from closer service provider continues propagation, while the other one is blocked. The paper contains theoretical analysis of the resulting structure called *iMesh* (Fig.1), and shows that iMesh satisfies both constant storage load and required proximity property. Both the theoretical analysis and simulation data are based on assumption that the static sensor nodes are placed in orthogonal grid structure. The transmission distance is such that each node is able to communicate only with its first neighbors in four geographical directions (and thus have 2-4 1-hop neighbors, depending on position of node in the grid). Once mesh structure is established, an event-sensing node which acts as service consumer performs service discovery by means of *cross-lookup* process. It sends search messages that propagate in four directions until they hit iMesh edges, when reply messages containing info about SPs are sent using backwards paths. Otherwise, if a search message hits terrain boundary, it is discarded and no reply is generated.

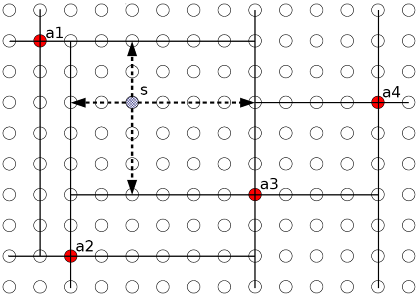


Fig. 1. iMesh construction and cross lookup in a grid network

1.4 Contribution

This research picks up where Li et al. [1] left off. We propose adaptation of the basic iMesh algorithm which can be used in dense networks with random topology. It is relatively easy to perform information mesh construction and lookup in a grid network, because all the propagation paths are either orthogonal or parallel, and what is more important, orthogonal paths must intersect at nodes. In the arbitrary network that might not be the case, so we needed to implement a different mesh construction mechanism to ensure that blocking rule and cross-lookup can be applied properly. We use geographic greedy forwarding for message propagation. When registration messages (which are used for iMesh construction), are propagated through the network, we create *mesh areas* instead of *mesh edges*. That is, all of the nodes that are in 1-hop neighborhood (within transmission radius) of nodes that are retransmitting registration messages store the information about corresponding service providers, regardless of weather they are addressed to transmit registration messages further or not. That way, we create "thick" iMesh areas instead of tiny iMesh edges. We also modified blocking rule [1], which is now applied whenever registration hits any node in another mesh area, and it is not necessary to ensure that registration messages meet at the same node, in contrast to the situation we have in grid network.

The rest of this paper is organized as follows. We present related work in Section 2. Section 3 describes network model, and contains detailed description of algorithms. In section 4 we present our simulation setup and results. Section 5 concludes the paper.

2 Literature Review

Grid Location Service (GLS) algorithm partitions the sensory field into grids and constructs a quad-tree structure over the grids [10]. It uses a hash function, designed on the basis of the quad tree, to match each node (by ID) to a unique subset of nodes called location servers. Every node updates all its location servers with its current location. A node can find the location of any other node by querying one of the location servers of that node. This protocol requires foreknowledge of the sensory field for grid partition. It may generate large message overhead since location updates and location queries travel along zigzag lines. GLS is not distance sensitive service discovery protocol because the length of data retrieval path may not be proportional to the distance of the service provider and service consumer.

Geographic Hash Table (GHT) scheme for data-centric storage is presented by Ratnasamy et al. [11] A node hashes data to a unique location by data type and routes the data to that location by a combined greedy-face routing protocol. The nodes that enclose the harsh location in a planar graph store the data; other nodes may get the data from any of these nodes. The main drawback of this scheme is that it is not distance sensitive. It features the undesired non locality-aware data query, which means that a node near the data source may have to travel a long distance to retrieve the data. Also, it may induce bottleneck spots when some types of data are frequently generated or requested.

Geography based Content Location Protocol (GCLP) utilize content servers which advertise their locations in four directions on a periodic basis [12]. Nodes receiving location advertisements become content location server. If a content location server receives multiple advertisements for a particular resource, it will only forward updates from the content server closest to it. This forwarding policy is an informal definition of the blocking rule generalized, formalized and used for information mesh construction [1]. Due to its periodic location advertisement, GCLP generates large message overhead.

Landmark-based data storage and retrieval scheme called LBIB is presented by Fang et al. [13] This scheme constructs a number of globalized shortest path trees of predefined landmark nodes, each rooted at a landmark node. A data producer hashes data (according to data type) to a certain landmark node and distributes the data using the shortest path tree rooted at that landmark node. A data consumer queries along the same shortest path tree; it gets the data when it hits the storage path or reaches the hash node. This scheme generates storage hot spots and involves many expensive global operations since global topology knowledge is required. It provides no energy efficiency and does not scale. Aydin et al. describe Pseudo-quorum match making service discovery [7]. The network is partitioned into subsets called quorums, where every two quorums intersect and no quorum includes another quorum. To accommodate node mobility and network scale, they proposed that service providers and service consumers systematically forward their advertisement and subscription messages to form pseudo quorums, where they are matched at intersecting nodes. However, successful data retrieval is not guaranteed with this scheme.

Double rulings scheme is proposed by Sarkar et al. [8]. Here, the data about service provider is stored along the curve instead in one or a few isolated nodes (as in e.g. GHT). Service consumers send query along another curve that guarantees to intersect service provider curve thus guaranteeing data retrieval. It is actually an extension of GHT with lower communication costs and is given for static networks.

In hierarchy decomposition scheme (HD) [9], the nodes in the network are classified into a hierarchy of clusters in which each node belongs to one cluster at each level. The data are replicated in the hashed nodes in all neighboring clusters of the service provider at all levels; the data are retrieved by querying the hashed nodes in the clusters in which the consumer resides in an increasing order of levels until a hashed node with the data-of-interest is reached. HD guarantees successful data retrieval; however, it demands a great deal of message and memory overhead to replicate the data since global topology knowledge is required.

Quorum-based location service is introduced in paper [2]. Each node distributes its current position along a "column" in the network. When a node wants to discover the location of another node, it searches along a "row" in the network. This row intersects the columns of all the other nodes, thus ensuring discovery. The weakness of this protocol is that location update and search has to cross the entire network, and network boundary has to be included to guarantee intersection. In addition, if all the nodes are collinear along a column, every node has to store every others location, thus suffering from large storage load.

Summarizing, all these algorithms (but [12]) have some, if not all, of the following weaknesses: requirement for foreknowledge of the network, frequent global computation, inconstant per-node storage load, communication bottleneck spots, and non locality-aware service lookup. On the contrary, the algorithm iMesh used in this paper has none of these drawbacks and combined with auctions may lead to efficient use of WSAN.

3 Preliminaries

3.1 Network Model

We observe distance-sensitive service discovery problem in WSAN in a given moment of time when service providers do not move. Although movement is the essential property of SPs in such network, our goal here is to perform service discovery in a static manner, so that movement of SPs later on is part of service itself, but not service discovery. So, from now on we will treat SPs as static network entities with the same properties as static sensor nodes.

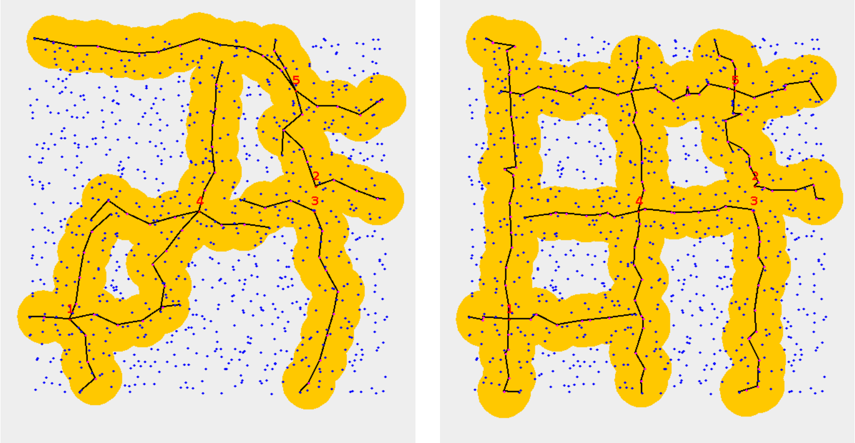
The network consists of set of m static sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$ and n SPs $R = \{R_1, R_2, \ldots, R_n\}$. All the devices are equipped with localization system, and thus aware of their coordinates (x, y) in two-dimensional sensing field. A single wireless channel is used for communication, and the network is modeled as unit disk graph. A network graph $G = (V, E)$, where $V = S \cup R$ is set of vertices which includes both static nodes and SPs, and there is an edge $e = (uv) \in E$ between nodes u and v if and only if Euclidean distance between them is less than or equal to transmission radius r $(|uv| <= r)$. We note $N(u)$ the set of neighbors of node u, i.e. the set of nodes v such that $uv \in E$.

3.2 iMesh Construction

As we mentioned in the introduction, service discovery process can be broken down to two phases. In the first phase, an information mesh structure is created. To do so, it is necessary that each of the nodes send hello message first, to enable creation of neighborhood tables. Figure 2 illustrates mesh construction in a network with 5 SPs and 1000 static sensors. Each SP (numbered nodes in red color) create up to 4 registration messages, depending on their relative positions. For example, SPs 2 and 3 are neighbors (2 is northeast from 3), so SP 2 generates just northbound and eastbound registration messages, while SP 3 generates westbound and southbound messages. A registration message is always passed on from sender node u to single receiver node v which is the furthermost node in general direction of propagation, among sender's neighbors. Black lines in the figure represent propagation paths. We also use the fact that all of the nodes that are neighbors to u $(w \in N(u))$also receive registration message (although it will be retransmitted only by node v), so they store the information about the SP from which the registration originates. This way, we create mesh areas, instead of mesh edges with practically no additional cost. There are two conditions which can stop further propagation of a registration message originating from SP r:

– It cannot be transmitted further, because the terrain boundary is hit.
– Receiver node v is already in mesh area of other SP r'. If $|vr'| < |vr|$, registration message is discarded, otherwise it continues propagation. This is a modification of blocking rule [1].

Of course, whenever a registration from SP r reaches some of the nodes that are already in mesh areas of SP r', if they are closer to r than to r', they need to update their closest service info with coordinates of r.



(a) iMeshC

(b) iMeshD

Fig. 2. iMesh construction

We named the above described protocol *iMeshC*, as the protocols for square grid networks have been named *iMeshA* and *iMeshB* [1]. The obvious shortcoming of iMeshC can be seen in figure 2(a). Mesh paths sometimes tend to twist and bend, due to non-uniform distribution of nodes. Thus, we changed the routing criteria for registration messages. They are no longer transmitted to furthermost node, but to node that deviates the least from straight propagation path. The resulting modification is named iMeshD, which is illustrated in figure 2(b).

In this algorithm, the criterion that the boundary is hit is inability to progress further in desired direction. This is acceptible in our scenario where we observe its performance in dense networks. In sparse networks, this criterion is unable to distinguish holes in topology from network boundaries. Authors of [1] and [3] suggest that registration paths should follow GFG protocol [4]. GFG routes registrations along the outer boundary of the network, and terminates when propagation paths form closed loops. Simulation results shown in section 4 justify our boundary detection criterion in cases when average node degree is high. This lowers number of messages needed for iMesh construction, while our simulations show that we still maintain very high percentage of best service discovery. Performance of service discovery protocols in sparse networks will be a subject of further research.

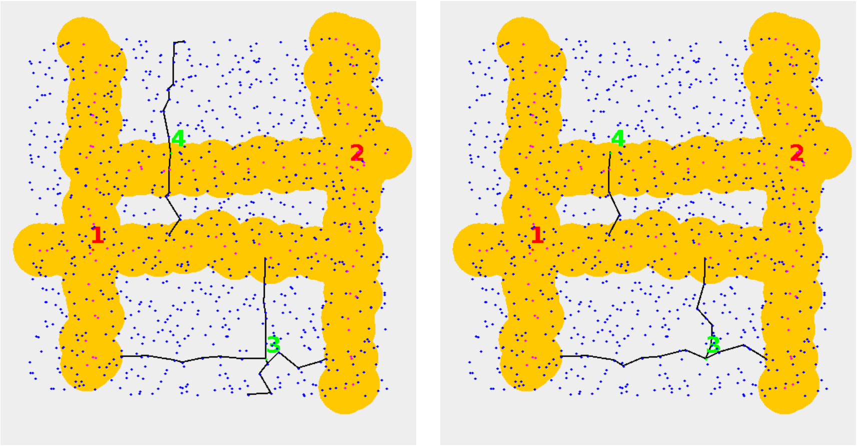
3.3 Cross-Lookup

Once information mesh is established, on occurrence of an event, the service consumer performs search for nearest provider by performing distance sensitive lookup method. It sends search messages in four directions, which propagate in the same fashion as registration messages, until they reach mesh areas, or terrain boundary. In the former case, a receiver node within mesh area generates answer containing info about service provider, and sends it back to node that initiated lookup. Answer is always routed by node closest to the destination. This is done for several reasons:

- The backwards path may differ from the forward path, so it is not necessary to store information about forward path either in nodes or within message payload.
- It may result in smaller hop count.

In the latter case, when search message hits boundary without reaching any of the mesh areas, the request is discarded and no answer is generated.

A service consumer node that is not within any of the existing mesh areas generates 4 search messages. On the other hand, it seems logical that when service consumer is within mesh area, no lookup is needed because it already has information about location of an SP. Still, that may not indeed be the closest SP, as it is shown in figure 3.



(a) cross-lookup search paths (b) cross-lookup answer paths

Fig. 3. iMeshD cross-lookup

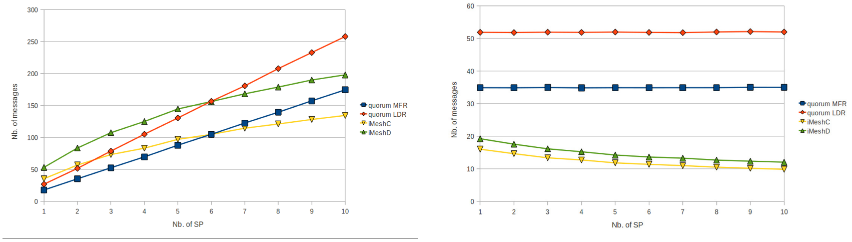
Service requesting node 4 is in the mesh area of SP 2, but it is actually closer to the SP 1. To find out about closer SP, it sends search messages, but only in direction orthogonal to direction of mesh area (in this case to north and south). The northbound search hits the boundary and gets discarded, while the southbound one hits mesh area of SP 1, and generates answer which is sent back to node 4. This way, node 4 gets the information about SP 1 even though it is within mesh area of another SP.

4 Simulations

In our simulations we have compared performance of two service discovery schemes: quorum and iMesh. Having two different routing criteria (most forward routing MFR and least deviation from straight line routing LDR), we have 4 resulting algorithms: quorum MFR, quorum LDR, iMeshC and iMeshD. In each simulation, we deploy randomly $m = 1000$ nodes in a square sensing field of size 1000×1000 length units. Clearly, the density of network depends on number of nodes, and transmission range. We fixed transmission range to 75, which gives us average node degree of about 16. We consider this to be a dense network, but the actual discussion about threshold node degree value after which a network can be considered as dense is beyond the scope of this work and is left for further research. Having fixed number of nodes and transmission range, we performed 10 series of 100 simulations for varying number of SPs from 1 to 10. Each simulation consists of 4 steps:

1. Initialize positions of static nodes and service providers
2. Construct service directory, depending on algorithm
3. Conduct service lookup scheme for every node in the network
4. Change service discovery algorithm and repeat steps 2-3

The values of interest are average number of messages needed to construct service directory, average number of messages per node needed to perform service lookup, and success rate in finding best (closest) service provider.



(a) Message count in service directory construction (total)

(b) Message count in service lookup (per node)

Fig. 4. Average message count in service directory construction and service lookup phase

Results shown in figure 4(a) and table 1 show average number of messages needed to construct service directory. When quorum is used, it grows linearly with number of SP, because each new SP network needs roughly the same number of messages to advertise its position along east-west direction. On the other hand, we observe that if we use iMesh, number of messages is initially higher because registration messages are propagated in 4 directions, but it also grows at a slower

Table 1. Average message count in service directory construction phase

Nb.of SP	quorum MFR	quorum LDR	iMeshC	iMeshD
1	17.54	26.57	35.15	53.00
2	35.06	51.37	56.60	83.24
3	52.32	78.45	73.01	107.25
4	69.40	104.84	83.00	124.78
5	87.64	130.30	96.84	144.37
6	104.69	156.45	104.60	156.07
7	122.35	180.57	114.27	168.11
8	139.27	207.57	120.88	178.66
9	157.12	232.69	127.93	189.73
10	174.33	257.87	134.23	197.83

Table 2. Average message count per node in service lookup phase

Nb.of SP	quorum MFR	quorum LDR	iMeshC	iMeshD
1	34.89	51.87	16.03	19.24
2	34.86	51.80	14.66	17.57
3	34.95	51.92	13.38	16.14
4	34.83	51.85	12.73	15.24
5	34.89	51.96	11.83	14.24
6	34.88	51.84	11.38	13.63
7	34.89	51.77	10.99	13.28
8	34.90	51.98	10.52	12.69
9	35.00	52.10	10.18	12.35
10	34.98	51.97	9.86	12.06

Table 3. Best service discovery success rate (in %)

Nb.of SP	quorum MFR	quorum LDR	iMeshC	iMeshD
1	98.62	98.67	99.95	99.96
2	99.03	98.93	99.49	99.60
3	99.30	99.31	99.13	99.12
4	99.00	99.05	98.76	99.04
5	99.43	99.44	98.65	98.96
6	99.01	99.11	98.09	98.39
7	99.31	99.30	98.19	98.64
8	99.42	99.42	97.88	98.27
9	99.41	99.43	97.96	98.30
10	99.55	99.53	97.77	98.54

pace due to the blocking rule, which brings localization property. In general case, iMeshC consumes less messages in this phase because its routing criteria allows it to make longer hops.

Simulation results for average number of messages in service lookup phase in figure 4(b) and table 2 show crucial advantage of iMesh over quorum, in terms of message efficiency. Both quorum algorithms experience approximately constant number of messages needed to perform service lookup, regardless of the number of service providers. This is due to the property of quorum that search paths always stretch along the entire column. iMesh is localized and distance-sensitive, so the number of messages needed to reach nearest mesh areas in each area decreases with increasing number of service providers. Again, iMeshC needs less messages than iMeshD for the same reason as explained for previous phase.

Finally, we observe the success rate in finding best service (table 3). It shows that all of the algorithms have high success rate (over 97%).

5 Conclusion and Future Work

In this research we introduced and examined generalization of localized distance-sensitive service discovery algorithm iMesh. We performed simulations to compare its performance with well known and widely used service discovery scheme (quorum). iMesh generally shows slightly lower success rate in finding best service when bigger number of service providers is present in the network, but it brings major improvement in reducing message overhead. Also, success rate of iMeshD is slightly better than iMeshC, but it is a matter for discussion if this really justifies its application due to increased amount of messaging, shown in simulation results.

During this work, we recognized some of the subproblems that need to be examined and investigated further. First of all, we need more strict definition of ”dense” network. Further, we need a more sophisticated meshanism for boundary detection, which would be able to distinguish boundaries from holes in topology. As extension of this research, we plan to examine behavior of service discovery algorithms in some specific network topologies, such as when holes and obstacles of various shapes are present in the sensing field. Also as we mentioned in the text, we will investigate performance and behavior of service discovery algorithms in networks with sparse node distribution.

The algorithms introduced in this paper also can be used as a solid starting point for solving more sophisticated and more complicated problems in WSAN, such as task assignment, when single or multiple events occur in the network simultaneously. Our experiences show that service discovery success rate can be improved even further by introducing communication in network consisting of service providers, and applying auction algorithms.

Acknowledgement. This work has been partially financed by the project Innovative electronic components and systems based on inorganic and organic technologies embedded in consumer goods and products, pr. num. TR32016, Serbian Federal Ministry of Science and Education.

Equipment resources to support this work were provided within the FP7 APOSTILLE project, no. 256615.

References

1. Li, X., Santoro, N., Stojmenovic, I.: Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks. IEEE Transactions on Computers 58(9), 1275–1288 (2009)
2. Stojmenovic, I., Liu, D., Jia, X.: A scalable quorum based location service in ad hoc and sensor networks. International Journal of Communication Networks and Distributed Systems 1(1) (2008) (invited paper)
3. Nayak, A., Stojmenovic, I.: Wireless Sensor and Actuator Networks - Algorithms and Protocols for Scalable Coordination and Data Communication. John Wiley & Sons, ISBN-13: 978-0-470-17082-3
4. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. ACM Wireless Networks 7(6), 609–616 (2001)
5. Noor Mian, A., Baldoni, R., Beraldi, R.: A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks. IEEE Pervasive Computing 8(1), 66–74 (2009)
6. Ververidis, G., Polyzos, C.: Service discovery for mobile Ad Hoc networks: a survey of issues and techniques. IEEE Communications Surveys & Tutorials 10(3), 30–45 (2008)
7. Aydin, I., Shen, C.C.: Facilitating match-making service in ad hoc and sensor networks using pseudo quorum. In: IEEE ICCCN (2002)
8. Sarkar, R., Zhu, X., Gao, J.: Double rulings for information brokerage in sensor networks. IEEE/ACM Trans. Netw. 17(6), 1902–1915 (2009)
9. Funke, S., Guibas, L.J., Nguyen, A., Wang, Y.: Distance-Sensitive Information Brokerage in Sensor Networks. In: Gibbons, P.B., Abdelzaher, T., Aspnes, J., Rao, R. (eds.) DCOSS 2006. LNCS, vol. 4026, pp. 234–251. Springer, Heidelberg (2006)
10. Li, J., Jannotti, J., Couto, D.S.J.D., Karger, D.R., Morris, R.: A Scalable Location Service for Geographic Ad Hoc Routing. In: Proc. ACM MobiCom, pp. 120–130 (2000)
11. Ratnasamy, S., Karp, B., Yin, L., Yu, F.: GHT: A Geographic Hash Table for Data-Centric Storage. In: Proc. Intl Workshop on Wireless Sensor Networks and Applications, WSNA, pp. 78–87 (2002)
12. Tchakarov, J.B., Vaidya, N.H.: Efficient Content Location in Wireless Ad Hoc Networks. In: Proc. IEEE Intl Conf. Mobile Data Management, MDM, pp. 74–85 (2004)
13. Fang, Q., Gao, J., Guibas, L.J.: Landmark-Based Information Storage and Retrieval in Sensor Networks. In: Proc. IEEE INFOCOM, pp. 286–297 (2006)

Quorum Based Image Retrieval in Large Scale Visual Sensor Networks

Stojan Milovanovic and Milos Stojmenovic

Singidunum University, Belgrade, Serbia
{smilovanovic,mstojmenovic}@singidunum.ac.rs

Abstract. A recent publication by [SPKK] introduces a framework and set of rules by which object recognition can work on a visual sensor network. Extracted features of the detected object are flooded (with reduced dimensionality at each hop) in the network. The Sensor will match the corresponding feature of the new object with a locally stored one, and send the query on the backward link toward the original detector for matching. Based on their framework we introduce an algorithm which attempts to minimize the number of messages passed within the network when performing an image retrieval task. Extracted features are distributed along a row, while query matching progresses along a column. We compare our results to the algorithm proposed by [SPKK] and achieve fewer transmissions in the retrieval step, and avoid flooding in the pre-processing phase. We expand our algorithm by constructing an information mesh of multiple detections of the same object, to achieve matching with the nearest copy. We also propose a novel feature reduction method, by diving the image into k2 subimages, and extracting features in each subimage. This allows replacing histogram based features with a wide range of other options.

Keywords: visual sensor networks, computer vision, object recognition.

1 Introduction

Recently, visual sensor networks have received attention since they attempt to combine the seemingly non congruent research areas of image processing and ad hoc sensor networks. A philosophical gap exists between the two since they arise from different requirements which need to meet in order to form a visual sensor network. Image processing usually has real time processing requirements, which are more important than memory, storage or power consumption, whereas wireless sensor networks focus on the minimization of power consumption at the expense of a heavy computational load.

The object detection and recognition area research of Computer Vision (CV) field extracts useful information and makes sense of raster imagery. Its goal is to identify objects in images regardless of color, orientation, scale, rotation, position or lighting condition. This is a difficult problem which has only proven successful with certain classes of objects. Usually such systems are very processor, data, and memory

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 449–458, 2012.

intensive which make them good candidates for parallel, powerful processing systems. However, distributed video surveillance applications are situations where each node in the grid is a visual sensor (such as a simple camera) and has limited computational capacity, but can also communicate with other nodes in the grid. Object detection and recognition tasks become distributed problems in this case, and may rely on the entire grid to form a consensus.

Due to the high volume of information, and elevated hardware requirements that are generated in CV tasks such as video surveillance, environment and traffic monitoring, communication between nodes in the network becomes a problem. The transmission and storage requirements of computer vision algorithms, would strain the network, if out of the box algorithms are directly applied to the network. Detecting and recognizing objects that have been previously seen by any of the sensors in the network would involve a great exchange of information between the candidate node and all other nodes that may have seen the same object. [SPKK] proposes one of the following two scenarios:

1. broadcasting the original video content to all nodes, so that each one can locally process queries,
2. broadcasting the unknown object query to all nodes, and wait for a response by the network.

In each case, a substantial amount of overhead would strain the network's resources, so [SPKK] proposes a hierarchical dissemination of information where each node only stores part of the feature vector of the queried object, but the network as a whole contains all of the relevant data to answer any query. They define a flooding based framework that spreads the feature vector out to n hops away from the node which first viewed an object. Later queries travel up the disseminated chain to retrieve the answer, where each node in the chain can either reject the query as a non-match, or allow it to pass one hop closer to the source node which makes the final decision. Their method involved much overhead which must be brought down to a minimum such that network communications are not strained. We incorporate the method proposed by [S1] where the node that first spots a target broadcasts the feature vector of this target to all of its horizontal neighbors. A query is performed by searching for this feature vector through all of its vertical neighbors. This way, full network broadcasting is avoided. Compared to the method proposed by [SPKK], we achieve fewer messages passed in the network for an arbitrary query. In case the sought after object is seen for the first time in the network, its query will come back empty, but will have to traverse the entire height of the network in search of an answer. Since we first send queries towards the closest edge of the network from the query node, and then away from the closest edge, our method can take more hops to result in an answer than the query method proposed by [SPKK]. In our experiments, we determine that when there is a positive outcome in 30% or more of queries, our method requires fewer search hops.

2 Literature Review

The main issue with VSNs is the minimization of message size and reduction in overall network traffic. Uncompressed, or otherwise unedited visual information which is to be transmitted over the network requires high bandwidth, which makes it a natural selection for optimization in grid (mesh) network computing applications such as VSNs. [LDK] propose two methods of compressing and transmitting images in wireless sensor networks that save considerable energy. [YSV] present an energy efficient JPEG-2000 image transmission system over VSNs. [LLC, WA and WA2] articulate compression schemes for visual data that is to be transmitted though VSNs.

The transmission techniques presented above were classified by [CWM] into single, multi hop, and finally end to end categories. [CWM] describe a forward error correction recovery mechanism for multi-path data transmission in VSNs and outline an algorithm for the tradeoff between end to end energy cost and reliability requirements of multi-path data transmission.

An algorithm for obtaining the 'vision graph' of a VSN is described in [CDR], where two nodes in such a graph are deemed adjacent if their cameras have predominantly overlapping fields of vision. This case is preferable when the 3D structure and position of objects is a desired outcome, but it increases data traffic between nodes, and therefore overall network throughput and processing load. [DR] propose a method of auto calibrating such network based cameras based on belief propagation. Here, camera node neighbors communicate directly and match scene points in order to perform calibration.

Apart from data compression and transmission algorithms, surrounding topics include data security, embedded visual systems and P2P VSNs. [LKZ] introduce a low complexity method of providing secure data transmission over VSN, which protects against eavesdropping attacks. [ABL] propose a system of traffic monitoring where individual cars and their license plates can be isolated. Arth and Bischof [AB] progress further in this field by developing an object recognition system based on an interest point detection linked to a vocabulary tree for real time surveillance. Their system is implemented on a DSP embedded device. In [PCPGM, QKRBS], the authors applied a multi-agent framework to the management of a surveillance system using a VSN. [FBBS] propose a distributed network of smart cameras for real-time tracking. They discuss the benefits of a distributed surveillance network compared to a host centralized approach. In [GB], the authors proposed a technique for tracking objects across spatially separated, uncalibrated, non-overlapping fields of view.

[SPKK] studies the problem of determining whether any of the (distant) nodes in the network has previously seen the same or a similar object compared to the newly acquired one at one of the nodes. Thus, it deals with knowledge distribution (feature distribution) in visual-sensor networks.

They propose a novel method for the distribution of features across a network of visual sensors, the hierarchical feature-distribution scheme (HFDS). Along with the HFDS, the candidate specifies four requirements, that have to be fulfilled by any recognition method, to ensure that the results of a recognition or matching in a distributed architecture will be the same as those in a non-distributed architecture.

Abstraction (requirement 1) provides a function that translates level N features into more abstract higher level N+1 features with reduced dimensionality, and reduced storage needs (requirement 2). There exists a metric which provides a measure of similarity between two feature vectors at each level N (requirement 3), which converges, meaning that the measure at level N+1 is not larger than the measure at level N. The main idea is that if there is no match at a higher level N+1 then there is no match at lower level N (requirement 4).

[SPKK] discusses how one can map four basic pattern (object) recognition methods onto the distributed visual sensor network using HFDS. Those four basic methods are: template matching, histogram matching, subspace methods, and a random projection method. For each of those methods [SPKK] proves that they fulfill the four requirements, described above. This ensures, that they will be, recognition-wise, equally successful in a distributed scenario, as in the non-distributed scenario. A few selected possibilities to map state-of-the-art methods for representation of visual samples on the distributed structure are: histogram of oriented gradients (HOG), pyramid of histograms of orientation gradients (PHOG) and covariance region descriptor (COV).

[SPKK] selected the publicly available COIL-100 image database to test the proposed hierarchical feature-distribution scheme. It contains images of 100 different objects, each one rotated by 5 degrees, 72 images per object. Simulation was performed on rectangular 4-connected grid networks.

Three feature distribution methods were simulated. In ‘flooding at match’, the captured image is stored locally, and each object search task is performed by flooding the lowest level 1 feature vector in the whole network. The node with the previously captured image will perform matching and respond to the node that detected the new object. Flooding means that each node receives a copy of the feature vector. In ‘flooding-at-learn’, the captured lowest level 1 feature vector is flooded to the whole network. Therefore the node that detected the new image already has the knowledge of previous encounters of that object and can match immediately. [SPKK] proposes a third method, M-hier, hierarchical distribution scheme, the original feature vectors are flooded as follows. The detecting node is the only one with the highest level 1 feature vector. Its horizontal and vertical neighbors receive level 2 feature vectors from it. These neighbors in turn forward level 2 feature vectors to its horizontal and vertical neighbors in an expanding direction. This process continues until reaching the highest defined level H. Afterwards flooding will continue by expanding level H features further to the remaining nodes in the network. During flooding, the coordinates of the source node can also be propagated in addition to the feature vector. When a new copy of an object is detected, it is compared with locally available feature vectors, at the level where that feature is available. For those that match, the highest level 1 feature of the tested object are sent to the original source by backward links. The source node then can decide if there is a match. Comparison is included in the communication load on the network. It can be simplified by counting each transmitted feature vector of length L as load L (this is then proportional to message size). Please see Figure 1 for a depiction of the M-hier algorithm. The red, encircled node is the source from which the feature vector is propagated throughout the network.

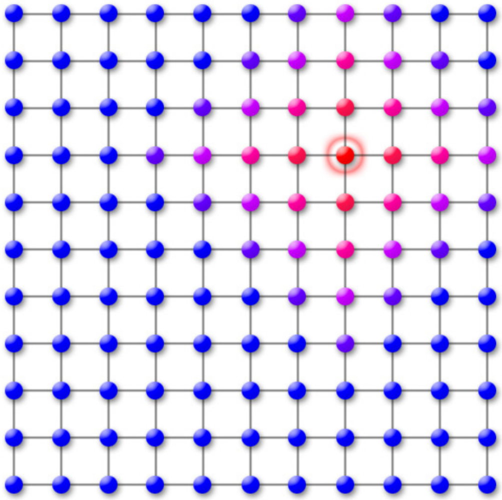


Fig. 1. M-hier feature vector network dissemination

Experiments in [SPKK] are performed using histogram matching only. The number of bins is a power of two, and feature vectors at level N+1 histograms are obtained using the mapping which combines adjoining bins from level N. That is, sum of data in bin 1 and 2 at level N produces datum in bin 1 at level N+1, the sum in bins 3 and 4 produces datum in bin 2 etc. This object detection method has some limitations. First, it is a 'whole image' matching. Images contain mostly the main object and little background. Extraction of objects from larger images is not covered here, and can be done by separate image processing techniques. This limitation will be also applied in our work, which will instead concentrate on the network scalability issue.

The other limitation is that the correctness of object matching itself is not questioned here. Each judgment is assumed correct. Therefore there is no impact of threshold T on the performance, as ground truth is not established (only later in some real experiments to some limited extent). Similarly, this will also not be a focus of our investigation – we will mainly deal with the matching algorithm itself and its communication overhead.

The main remarks is that proposed M-hier algorithm is not sufficiently scalable. It is still based on flooding the whole network, which consumes bandwidth despite reducing the level of information. In the search phase, the lowest full size feature vector is still communicated between newly and previously detected locations.

[SLJ] overcomes message flooding deficiencies, by proposing a quorum-based location service. The destination node registers its location along a 'column' to form an update quorum. The source node makes a query along a 'row' to form a search quorum. The destination location is detected at the intersection between the update and search quorums. The overhead of each routing task, including location service, is O(√n), where n is the number of nodes in the network. In Figure 2, we depict the

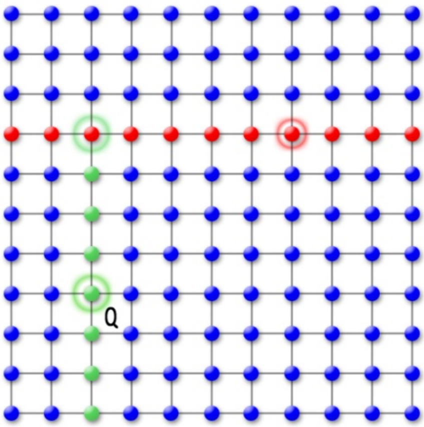


Fig. 2. [SLJ] full feature vector horizontal dissemination, vertical query from node Q

horizontal feature vector spread and vertical query method proposed by [SLJ]. The full feature vector is spread horizontally throughout the entire grid, and all queries are performed vertically.

3 Contribution

We address the scalability issue with the work of [SPKK], and correct it using the scheme proposed by [SLJ]. Essentially, we avoid the flooding strategy employed in SPKK to diffuse the feature vector of the target image throughout the network, and also shorten the hop count of the query message in order to get a result.

3.1 Quorum Based Image Retrieval

The feature vector can be any array of features which follow the rules set out in [SPKK]. Choosing the most accurate feature vector for general object detection is a research area of its own, and not a focal point of this paper. We focus on the overall hop count, and minimizing message traffic in the network. For our purposes, we selected edge orientation histograms as the main feature vector. The feature vector of each image is transmitted to each horizontal node, and queries are done vertically as proposed by [SLJ]. We modify the query algorithm so that the query node first notes its location relative to the edges of the network, and performs the search up or down first depending on its proximity to the border of the network. This way, fewer messages are passed in the network, at the expense of time required to get a result.

Queries performed in [SPKK] can only be answered at the source node, which means that each query must travel to it and back in order to be answered. In the worst case, there can be at most $2\sqrt{n}$ hops required to reach the node which contains the full feature vector, and another $2\sqrt{n}$ for the answer to reach the query node, where n is the number of nodes in the network.

3.2 Feature Extraction from Sub-Images

We determine that images can be divided into k2 subimages by dividing rows and columns into k parts. A feature extraction method with d dimensions can be applied on each of subimages. This together gives k2d-dimensional feature representation. Feature reduction is then obtained by reducing 4 subimages into a single image (then k=1,2,4,8,…). The four properties given in [SPKK] can be proven for a wide range of specific feature extraction methods. This way, HLAC, SIFT, Viola's Haar wavelets etc. can replace the simple histogram based features.

3.3 Q-Hier Based Feature Distribution

The direct improvement of M-hier [SPKK] is then Q-hier as follows. The feature vector of the detected object is distributed in its row only, instead of the whole network. Each search is then performed in the column of the query node, by transmitting the lowest level 1 feature vector. A match can be determined at the node which intersects the query column and the feature row. If there is no match, the search stops. In case of a match, the lowest level 1 feature is forwarded toward the original source, and can be tested similarly along the route, stopping with the first failure, or reaching it for final test (if it is the only node with the originally stored lowest level feature vector then only that node can make a positive decision). Compared to M-hier, row distribution may be unnecessary in case of the first mismatch. But flooding the whole network is avoided.

Note that if we have only one level of feature vectors (H=1) then Q-hier is simply a quorum based scheme. Since we will only simulate rectangular networks, it is then the basic row-column variant of it. Its superiority over M-hier for H=1 is then already demonstrated in the original papers on the quorum scheme [SLJ]. We implement this scheme in our experiments.

3.4 iMesh: Multiple Image Copies

Next, [SPKK] assumes that one image is stored in only one node, throughout the process. This does not address the third, fourth etc. appearance of the same image. The node that discovered an object for the second time in the network can also serve, together with the original node, in matching for further appearances. If several copies already exist then iMesh from [LSS2] can be used. Again, for H=1 there is no difference in algorithm and performance gains compared to [LSS2].

4 Results and Discussion

We constructed a test set of 100 arbitrary images that were used to verify our theoretical results. The [SPKK] algorithm was compared with our own work on a 100 x 100 node grid. We chose to compare our Q-hier scheme with H=1 to their M-hier algorithm. The experiment was set up to choose a random image from the test set, and compute its feature vector based on its histogram of edge orientations. This vector was then diffused in the network via the schemes proposed by the competing algorithms. Random nodes were then chosen in the network that issued queries based on feature vectors computed from the other images in the test set. The hop count of retrieving an answer to a query was counted and compared between the two algorithms. It was observed that [SPKK] outperformed our algorithm in terms of hop count when the number of occurrences of the queried image not being in the network was very large. As the number of positively answered queries approached a threshold of 90%, our algorithm produced better results. This means that once the network becomes aware of its surroundings, our algorithm tends to outperform the scheme proposed by [SPKK]. We implemented the variant where only one row of the network contains the feature vector, which decreases message traffic, but increases the hop count of typical queries.

The input to the algorithm is a set of 100 images: (I1, I2, ... I100), and a grid network of size n x n. The expected output is the average number of hops required to determine if image I has been previously seen in the network. For each iteration of the experiment, a random node is selected as the source node in the network. The edge orientation histogram (or any variant of a feature vector) is spread horizontally to all nodes in the same row as the source node. Random nodes are then selected in the network and query images are assigned to them. Each query image is converted to its corresponding feature vector, and queries are processed vertically through the network. Feature vectors are compared using correlation to determine whether the query image is present in the network.

4.1 M-Hier-H, Q-Hier-H, M-Hier-B, Q-Hier-B: Feature Level Distribution

Determining the best level distribution remains to be investigated. This is not resolved even in the original solution, because the initial node can calculate all levels and immediately flood the highest level to the whole network. This can be defined as method M-hier-H. There will be a reduction in communication cost, and no modest savings in the search phase, since tests at higher level would trigger more contacts to the source that eventually prove false. Similarly Q-hier-H can be defined, which restricts communication in rows and columns.

Further options include dividing these levels in different ways. For example, a balanced method may divide the number of rows (assume C=R for simplicity) R by the number of levels H, and reuse each level R/H times. This may define two new algorithms M-hier-B and Q-hier-B, respectively.

Acknowledgment. This work was partially supported by the following grant: "Digital signal processing, and the synthesis of an information security system", TR32054, Serbian Ministry of Science and Education.

References

[AB] Arth, C., Bischof, H.: Real–time object recognition using local features on a dsp–based embedded system. Journal of Real–time Image Processing 3(4), 233–253 (2008)

[ABL] Arth, C., Bischof, H., Leistner, C.: TRICam - an embedded platform for remote traffic surveillance. In: Conference on Computer Vision and Pattern Recognition Workshop (2006)

[CDR] Cheng, Z., Devarajan, D., Radke, R.J.: Determining vision graph for distributed camera networks using feature digests. EURASIP Journal on Applied Signal Processing 2007(1) (2007)

[CWM] Charfi, Y., Wakamiya, N., Murata, M.: Trade-off between reliability and energy cost for content–rich data transmission in wireless sensor networks. In: 3rd International Conference on Broadband Communications, Networks and Systems, pp. 1–8 (2006)

[CWM] Charfi, Y., Wakamiya, N., Murata, M.: Challenging issues in visual sensor networks. IEEE Wireless Communications 6(2), 44–49 (2009)

[DR] Devarajan, D., Radke, R.J.: Calibrating distributed camera networks using belief propagation. EURASIP Journal on Applied Signal Processing 2007(1), 221 (2007)

[FBBS] Fleck, S., Busch, F., Biber, P., Strasser 3d, W.: surveillance - a distributed network of smart cameras for real-time tracking and its visualization in 3d. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, p. 118 (2006)

[GB] Gilbert, A., Bowden, R.: Incremental, scalable tracking of objects inter camera. Journal of Computer Vision and Image Understanding 111(1), 43–58 (2008)

[LDK] Lecuire, V., Duran-Faundez, C., Krommenacker, N.: Energy-efficient image transmission in sensor networks. International Journal of Sensor Networks 4(1), 37–47 (2008)

[LKZ] Luh, W., Kundur, D., Zourntos, T.: A novel distributed privacy paradigm for visual sensor networks based on sharing dynamical systems. EURASIP Journal on Advances in Signal Processing 2007(1) (2007)

[LLC] Lu, Q., Luo, W., Wang, J., Chen, B.: Low-complexity and energy efficient image compression scheme for wireless sensor networks. Computer Networks 52(13), 2594–2603 (2008)

[LSS2] Li, X., Santoro, N., Stojmenovic, I.: Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks. IEEE Transactions on Computers 58(9), 1275–1288 (2009)

[PCPGM] Patricio, M., Carbo, J., Perez, O., Garcia, J., Molina, J.M.: Multi–agent framework in visual sensor networks. EURASIP Journal on Advances in Signal Processing 2007(1) (2007)

[QKRBS] Quaritsch, M., Kreuzthaler, M., Rinner, B., Bischof, H., Strobl, B.: Autonomous multicamera tracking on embedded smart cameras. EURASIP Journal on Embedded Systems (2007)

[SLJ] Stojmenovic, I., Liu, D., Jia, X.: A scalable quorum based location service in ad hoc and sensor networks. International Journal of Communication Networks and Distributed Systems 1(1), 71–94 (2008); invited paper

[SPKK] Sulic, V., Pers, J., Kristan, M., Kovacic, S.: IEEE Transactions on Circuits and Systems for Video Technology 21(7), 903–916 (2011)

[WA] Wu, H., Abouzeid, A.: Energy efficient distributed image compression in resource-constrained multihop wireless networks. Computer Communications 28(14), 1658–1668 (2005)

[WA2] Wu, H., Abouzeid, A.A.: Error resilient image transport in wireless sensor networks. Computer Networks 50(15), 2873–2887 (2006)

[YSV] Yu, W., Sahinoglu, Z., Vetro, A.: Energy efficient JPEG 2000 image transmission over wireless sensor networks. In: IEEE Global Telecommunications Conference, GLOBECOM 2004, pp. 2738–2743 (2005)

From Real Neighbors to Imaginary Destination: Emulation of Large Scale Wireless Sensor Networks

Bogdan Pavkovic[1], Jovan Radak[2], Nathalie Mitton[2], Franck Rousseau[1], and Ivan Stojmenovic[3]

[1] Grenoble Informatics Laboratory (LIG), University of Grenoble, France
{firstname.lastname}@imag.fr

[2] INRIA Lille - Nord Europe, France
{firstname.lastname}@inria.fr

[3] SITE, University of Ottawa, Canada
ivan@site.uottawa.ca

Abstract. The ultimate test for many network layer protocols designed for wireless sensor networks would be to run on a large scale testbed. However, setting up a real-world large scale wireless sensor network (WSN) testbed requires access to a huge surface as well as extensive financial and human resources. Due to limited access to such infrastructures, the vast majority of existing theoretical and simulation studies in WSN are far from being validated in realistic environments. A more affordable approach is needed to provide preliminary insights on network protocol performances in large WSN. To replace large and expensive realistic testbeds, we introduce a novel approach to emulation. We propose a specifically designed experimental setup using a relatively small number of nodes forming a real one-hop neighborhood used to emulate any real WSN. The source node is a fixed sensor, and all other sensors are candidate forwarding neighbors towards a virtual destination. The source node achieves one forwarding step, then the virtual destination position and neighborhood are adjusted. The same source is used again to repeat the process. The main novelty is to spread available nodes regularly following a hexagonal pattern around the central node, used as the source, and selectively use subsets of the surrounding nodes at each step of the routing process to provide the desired density and achieve changes in configurations. Compared to real testbeds, our proposition has the advantages of emulating networks with any desired node distribution and densities, which may not be possible in a small scale implementation, and of unbounded scalability since we can emulate networks with an arbitrary number of nodes. Finally, our approach can emulate networks of various shapes, possibly with holes and obstacles. It can also emulate recovery mode in geographic routing, which appears impossible with any existing approach.

Keywords: emulation, simulation, routing, wireless sensor networks.

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 459–471, 2012.

1 Introduction

A plethora of theoretical results and practical applications have emerged from the wireless sensor networks (WSN) research domain. One of the main issues in WSN is experimentation on real testbeds. The vast majority of existing testbeds consist of several dozens of sensor nodes. They satisfy the need for experimenting with centralized algorithms in small scale deployments. However experimental evaluation of network layer protocols intended for large scale WSN is still unfordable to most researchers due to several issues. The cost of buying a large number of sensors to start with, most researchers normally do not have such resources. Then the need to deploy them physically in vast environments. Finally, providing appropriate human resources for their maintenance. Thus, the most popular way for validating and comparing algorithms and solutions is through carefully driven simulation [15], using different types of software simulators such as ns-2, OPNET, WSNet. There is a handful number of existing large scale sensor network deployments, such as Senslab [13], Wisebed [20], GreenOrbs [8], allowing researchers to test their solutions in a real environment. Using these testbeds for validation and comparison of protocols raises two main challenges: *scalability* (how protocols perform on larger networks?), and *pattern* (sensors are usually placed in a regular structure and with certain density which limits possible patterns for investigation).

A compromise between simulation and real testbeds is emulation. Emulation combines elements of real environment experimentations with some assumptions normally taken during simulations. It generally has realistic parameters which are directly incorporated (by software) into the architecture being used [14,17,6]. Sometimes all nodes are real. Virtual nodes might also be added [3]. In Wisebed project [20], simulated node corresponds to either existing or virtual node. Links between virtual nodes, between a real and a virtual node, and even links between two real nodes are simulated (parameters are sent to a base station that makes simulated decisions and returns communication results back to the real participating nodes) [20].

A solution based on using smaller networks to emulate the behavior of the large scale networks has been proposed [9]. Up to 50 sensors, all within a 1-hop neighborhood, are deployed. The real source node is placed at the center of a real 1-hop neighborhood, while the destination is virtual and placed outside this neighborhood. The process of emulation is performed as follows. *(i)* The message is being held at the source node S, all available nodes serve as actual one-hop neighbors, and S chooses the best forwarding node B among them, according to the routing algorithm. *(ii)* Node B is remapped to source node S and the position of the virtual destination is recalculated, translating it by the vector $-\overrightarrow{BS}$. In order to provide a realistic variability of the signal propagation conditions, the virtual destination is also rotated around S. The goal is to change the set of potential forwarders experienced by the source node S between the successive steps as in real routing. We will refer later in the text to this as randomness of 1-hop neighborhood. *(iii)* Node S is again source node (after remapping from node B) and the process is repeated until the virtual destination falls within 1-hop neighborhood or the routing algorithm fails.

With this setup, we can provide realistic and accurate results using real sensor nodes and real wireless links among them, while providing scalability for experimentation scenarios with a virtually unlimited size of the emulated network. Each sensor is provided with its own geographic location (by software following actual measurements), and accurate location for all of its neighboring sensors. The local neighborhood of the source node is fixed and a subset of the same nodes are forwarding candidates at each hop. Emulation is only due to enforced mapping of large set of virtual nodes to a small set of real nodes.

In this paper we propose a different approach to the emulation of large scale wireless sensor networks. The main goal is to achieve better randomness of neighborhood structure, and control network density at the desired level. We use pseudo-randomness instead of full randomness in the location of neighbors. Our 43 nodes are deployed in a hexagonal pattern. We also "return" the message from the forwarding neighbor back to the initial source node by our additional software links, so that the next step may be carried out. However we do not need to rotate the virtual destination node. Instead, the original full size simulated network is translated so that the source node is the origin, and its neighbors from simulated network are all rounded to their nearest physical node from the actually deployed hexagonal network. Consequently, only a subset of the nodes forming the real hexagonal network are considered as forwarding candidates, and this set changes at every routing step. We have evaluated our system on greedy geographical algorithm (GARE) and cost over progress greedy algorithm (COP_GARE) [9] as well as XTC algorithm [19].

In Section 2 we give more details on emulation of sensor networks and related work. The description of our emulation setup with the details on the theoretical background are given in Section 3. We present results obtained with this emulation framework and evaluate their accuracy in Section 4. Finally we summarize and give conclusions in Section 5.

2 Related Work

The ultimate test for many protocols would be to run them on an existing real-world large scale wireless sensor network testbed [20,13,1]. The largest WSN testbed in the world is GreenOrbs which consists of 1 000+ deployed sensors [8]. Physical testbeds for WSN systems tend to be small in scale, expensive to maintain and time consuming to set up, mainly due to the huge amount of human resources needed [3]. They also lack in flexibility, offering only a fixed topology and limited heterogeneity and programmability. Repeatability is most of the time impossible since many relevant operating parameters are beyond user control.

2.1 Different Approaches to Emulation

Different approaches have been proposed for emulation so far. The main goal is to overcome the shortcomings of simulation while staying away from the complexity of real world experimentation and providing a certain level of repeatability. It duplicates the functionality of one system in terms of another system, offers greater fidelity than simulation and greater flexibility than physical testbed. Nevertheless, in the literature, the term emulation has been endorsed by different approaches.

Emulation of large scale networks has been studied in a variety of contexts [2,10,5]. Few different types of approaches are proposed. The main goal in all approaches is to overcome the deficiencies of simple simulation either with environment emulation — in which the characteristics of the real nodes are built-in and executed in simulator — or using network emulation — in which each node communicates with a real node in order to obtain more accurate results. Fall [4] distinguishes network emulation — where real traffic is exchanged between real and simulated nodes — and environment emulation — where real implementations are embedded in a controlled environment.

One of the first definitions of emulation was proposed by Ke *et al.* [7] by using a combination of simulated and real sensors. They have added three simulator modules, real time scheduler, network objects and tap agents, to achieve better cohesion with the hardware and to better simulate it in the ns-2 simulator. Two real machines are attached to the endpoints of the network producing the traffic that is later injected to emulated nodes by the ns-2 server. Satisfactory results were obtained for the emulation of 10 to 120 mobile nodes. However the experiments could not be carried out in real time.

A similar concept can be found in SensorSim [12], where the authors propose hybrid simulations. They combine real sensor readings (geophone, microphone and infrared detector) with simulated nodes in order to get more realistic event detection in military scenarios.

Object-oriented representation of sensors, communication channels and physical media (mobility model, power model, etc.) has been used in J-Sim [14]. In this approach, a virtual simulation environment is integrated with a small number of real hardware devices to facilitate performance evaluation of real-life devices at a large scale. Application specific models are developed using an object-oriented model, subclassing the simulation framework. The environment is thus well controlled and hardly tunable to fit any other deployment context.

Physical layer emulation has been proposed using an FPGA based DSP engine [6]. RF signal propagation in a physical space can be emulated for a wide variety of wireless devices. This work focuses on channel emulation and provides satisfactory results for higher layer performance evaluation of real systems in a controlled propagation environment.

Coulson *et al.* [3] have described a virtual testbed model for flexible experimentation in WSN by seemingly integrating physical, simulated and emulated sensor nodes and radios in real time. It was also demonstrated as part of the WiseBed [20] project. Their main approach to emulation is to add simulated nodes, emulated nodes, virtual links between two nodes (each endpoint could be a real, emulated or simulated node) and support different inter-node connectivity patterns in a physical testbed. Virtual link between two physical nodes can be created by connecting each of them to a base station, connected via Internet, and simulating the link performance in real time under different parameters from those currently present in the testbed. A base station connected to a physical node can similarly run a virtual link between a physical and a simulated node. However, the concept of emulated node and virtual links involving such nodes, has not been properly defined or exemplified.

2.2 Emulation Using a Real One-Hop Neighborhood

Our current emulation framework is based on an idea introduced in previous work [9]. A set of real nodes is placed randomly using the MIN-DPA algorithm [11] in a 1-hop neighborhood of the designated central node S, within a circle of radius r. The routing destination D is situated further away from the 1-hop neighborhood, and is a virtual node. This setup is depicted in Fig. 1.

At each step, node S chooses a new forwarding node (*e.g.* node B) according to the routing algorithm being used. Then the packet being routed is delivered to the selected next hop over the real radio link — with medium access, propagation and interference. When using greedy routing, the packet makes progress towards the virtual destination D, if at least one neighbor is closer to D than S. The coordinates of the virtual destination D are updated by $-\overrightarrow{SB}$ and correspondingly, node B, the new holder of the packet in transit, is translated to the original source node S. This translation allows to reuse the same physical nodes as forwarding candidates in the next step. In other words, the virtual destination D can be considered closer, at the new position D', to same 1-hop neighborhood, and mobile. The same procedure is repeated until the position of the virtual destination D falls inside the 1-hop neighborhood, *i.e.* it can be reached directly. In this final step, routing towards the nearest physical neighbor to D is performed.

One problem with this approach is that the layout and size of the 1-hop neighborhood of S does not change between successive routing steps, which is not realistic. To overcome this to a certain extent, D' is further rotated by a random angle to a new position D'' as shown in Fig. 1. Hence the actual candidate neighborhood subset is rotated at each step, while preserving the distance to the source. This results in more variability in the 1-hop neighborhood, providing more randomness in the emulated neighborhood along the path. Obviously, we cannot expect full randomness of the 1-hop neighborhood with such a rotation.

The other problem with this emulation is that the neighborhood density is fixed at the very beginning, at deployment stage, and is not changed or controlled later during emulation. Further, while rotating the virtual destination increases the randomness of the neighborhood, it does not allow to control an experiment with obstacles and deal with sparse networks. When no neighbor is closer to destination than the current node, recovery mode is called upon. Recovery mode cannot be emulated with this approach, because the rotation of the neighborhood will interfere with the mandatory traversal of particular faces of the Gabriel graph.

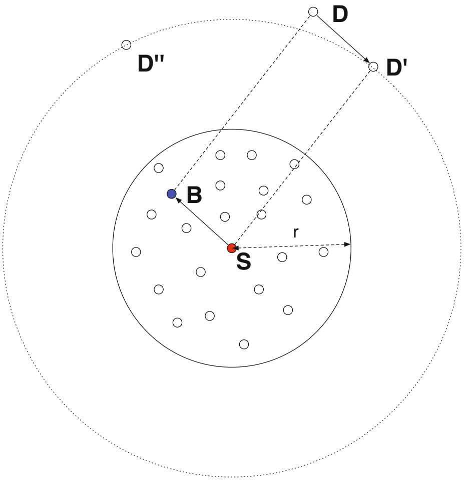


Fig. 1. One-hop neighborhood of S, translation and rotation of D

3 Mapping the Simulated Network to a Hexagonal Neighborhood

Our emulation approach improves upon the one proposed in our previous work [9], and differs from the other emulation approaches existing in the literature. We use realistic radio communications and aim to use the real radio links, without simulating them. This makes our approach fundamentally different from the other aforementioned emulation methods.

Our emulation framework uses a small wireless sensor network, up to 50 sensor nodes, to study the behavior of large scale wireless sensor networks. Instead of the random neighborhood structure described in 2.2, we deploy a set of sensors in a grid-like structure based on regular hexagons. Randomness and different densities of the 1-hop neighborhood are achieved at each step by using only a subset of the nodes in this regular structure.

Our 1-hop environment consists of 43 WSN430 sensor nodes placed in a hexagonal grid P as shown in Fig. 2 (dotted part named *physical hexagonal grid* P). The origin S is positioned at the center of this structure. At each emulation step, only node S forwards the routed packet over a real radio link. The positioning of the sensor nodes following a hexagonal pattern ensures that in two successive steps, each node lying at the intersection of two successive neighborhoods, translates to one of the nodes of the real network grid, due to central and axial symmetry of the proposed structure.

Emulation using this regular hexagonal structure can be explained in the following way. Suppose that we have a network H with nodes generated randomly or by following any desirable distribution. This network is composed of a large number N of nodes to simulate, *e.g.* $N > 1\,000$, see Fig. 2. The source U and

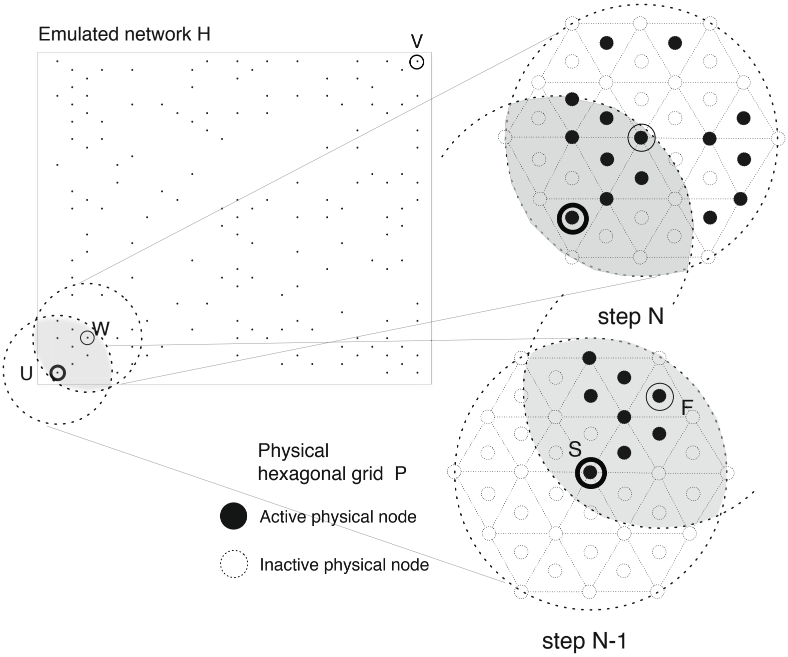


Fig. 2. Mapping the simulated neighborhood of the simulated source node to the real nodes surrounding the real source node of the physical network in two consecutive routing steps

destination V are set for a routing task. We translate U to the center S of our physical network P. Both networks have the same transmission radius — if necessary, additional scaling can be performed to achieve that. Destination V is translated to virtual destination D. Each neighbor W of the source node U is translated to the virtual node W' in the neighborhood of S. This virtual node is mapped to physical node F by rounding its coordinates to the hexagonal grid, that is, it is mapped to its nearest physical node. The set of neighbors of U is mapped to a set of neighbors of S, marked by filled discs in Fig. 2. This means that some neighbors of S from P are activated, while others are deactivated for the next routing step. We then use the physical links to activated nodes to select the forwarding node F among them, whose origin is simulated node W from H.

Then W becomes the new source node. We translate H further by vector $-\overrightarrow{UW}$, which then moves also the destination to a new position. This new neighborhood will map to a new set of activated nodes around S. Note that the set of previous and current source neighboring nodes overlap, and that this intersection results in a neighborhood structure that is preserved but translated, which is represented as a shaded region in Fig. 2. This procedure is repeated until we reach destination D or the routing algorithm fails.

The main advantage of our method, even compared to the full size experimentation, is the ability to work with arbitrarily dense networks, by placing the appropriate number of nodes in the 1-hop neighborhood. Furthermore, the source node S can have a different neighborhood across successive steps, by using only a subset of the regular structure as candidate forwarders. Although highly regular, this structure is offering significant variety in node placement and distance to central node S. This approach allows to emulate a wide range of densities. However at too high densities, several simulated nodes could map to the same physical node, which poses the limit on this emulation.

The ultimate advantage of our method is its unlimited scalability. It can provide virtually unlimited network scenarios. We will obviously sacrifice some accuracy and fidelity, but are likely to gain much more insight compared to pure simulations. Our method can be applied on existing testbeds, from 50 to 1 000 nodes, to emulate the performance of even a million sensors network.

4 Experimental Results

4.1 Experiments Using the Basic Emulation Setup

This section contains experimental results obtained using the basic emulation idea from previous work [9]. The experiment was done using a physical testbed consisting of 43 WSN430 sensor nodes [21]. The first phase of the experiment serves to gather statistics on link qualities, used directly afterwards for evaluating the routing algorithms. During this phase, each node transmits a batch of 128 messages, one node at a time, while the other nodes are measuring the number of successfully received messages from each node separately. Each node transmits 1 024 messages in 8 series. Statistics on link qualities from each separate node are gathered in the central node S (referent node). This allows to compute $ETX(uv)$ (expected transmission count) for the whole 1-hop neighborhood link set and use it as part of the routing metric.

We calculate $ETX(uv)$ for each given link uv as:

$$ETX(uv) = \frac{1}{p(uv) \cdot p(vu)}$$

where $p(uv)$ and $p(vu)$ are message reception probabilities based on measures over 1024 sent messages between nodes u and v. For our case where nodes are static, the reception probability does not change significantly over time (standard deviation in the number of received messages in separate series was 4.61 %). This ETX measure is used during the routing phase. We have evaluated the performance of three different geographical routing algorithms, XTC [19], GARE (greedy algorithm in real environment) and COP_GARE (cost over progress greedy algorithm in real environment) [9].

GARE is a localized greedy routing algorithm where the current node makes routing decision based on its position, the position of the destination, and its 1-hop neighbors. We calculate locally the Relative Neighborhood Graph (RNG) using $\frac{ETX(uv)}{|uv|}$ as a weight function for link uv. The longest edge in each triangle is not considered as a candidate for forwarding. Among RNG links, the one which provides the largest progress towards the destination is selected. If there is no RNG edge with positive progress, the next node is chosen using the same criteria of minimal distance to destination, among the remaining edges.

COP_GARE is based on the COP algorithm [16] adjusted by using ETX as a weight function. Among the neighbors with positive progress, we choose the one with the minimum ratio $\frac{ETX(uv)}{|ud|-|vd|}$, where progress $|ud| - |vd|$ is the difference between current distance from the destination $|ud|$ and possible distance $|vd|$.

Similarly to GARE, XTC [19] is based on the RNG structure, but it uses only ETX as a weight function. This algorithm prefers the most reliable and at the same time the shortest links.

The success rate was 100 % for all algorithms due to the high density of the 1-hop neighborhood (the node degree of the source node was 41 at every step of the emulation) and the uniform placement of the sensor nodes which allowed the routing algorithms to have positive progress at every step.

All the routing algorithms demonstrated their expected behavior. XTC had the highest total number of hops and retransmissions, because it systematically selected short RNG edges. GARE algorithm shows overall improvement in energy consumption (smaller number of hops), using longer links with slightly worse ETX. COP_GARE outperformed GARE and XTC since it did not restrict to the RNG neighborhood subset. We present performance results of the three routing algorithms averaged over 100 runs in Table 1.

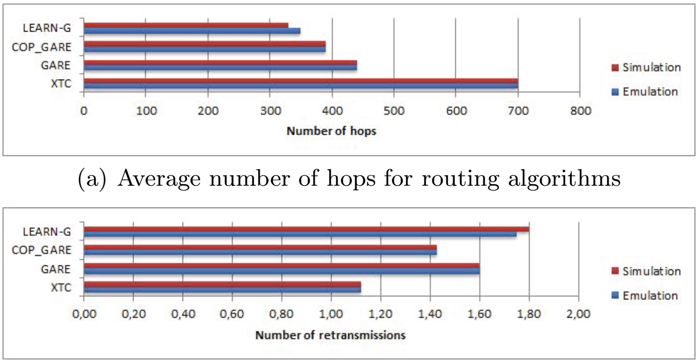
Table 1. Performance comparison with randomized graph orientation

Routing algorithm	**Total No. of transm.**	**Total No. of hops**
XTC	6 196	5 820
GARE	4 948	3 103
COP_GARE	2 162	1 536

4.2 Emulation Using a Hexagonal Grid Neighborhood

This new emulation setup is introduced to avoid using a rotation of the virtual destination, and to use only a subset of the available neighbors instead of the full set, as elaborated above.

The large scale simulated network was generated using an adapted version of the MIN-DPA algorithm [11]. MIN-DPA algorithm generated connected graphs with no crescent holes. In a nutshell, MIN-DPA calculates first an approximate transmission range r such that the expected node degree is equal to d (desired density). It then places n nodes sequentially, in n rounds. The i-th node is placed based on the positions of previous $i-1$ nodes. Approximate degrees d_j of nodes already placed (based on r) are calculated. Proximity constraint is satisfied if node i is not isolated from the previous nodes based on the approximate range r and it is no closer than d_{min} to any of the previous nodes. The next node is placed in the neighborhood of node with minimal d_j value.



(a) Average number of hops for routing algorithms

(b) Average number of retransmissions per hop

Fig. 3. Comparison of the results for emulation and simulation

We performed two simulations, and compared same algorithms over the same network scenarios to show proof of concept of our approach and to emphasize its benefits. One simulation does not use any concept of emulation. It simply runs same algorithms on the full network. The results are labeled by "Simulation" in Figure Fig. 3. The second simulation follows our emulation setup. The actual emulation (physical testbed) was not carried. Instead, the whole process, as described here, was simulated. Thus we simulated the emulation process. Emulation was simulated using the placement of 43 sensor nodes in a hexagonal grid in a 1-hop neighborhood of radius $r = 5\,m$. The results are labeled with "Emulation" in Figure Fig. 3. Both "simulation" and "emulation" were carried based on the same physical layer models. In that sense, the comparison made shows the pure impact of our emulation, and results are more realistic then if the comparison were made with testbed simulation, where physical layer and its impact would be unpredictable, and comparison with "simulation" is difficult. Overall, Figure Fig. 3 shows small difference between "simulation" and "emulation" in all cases.

We proceed by evaluating the performances of different geographical routing algorithms based on ETX measurements. As in 4.1, we implemented XTC, GARE, COP_GARE and additionally LEARN-G. LEARN-G is a variant of the LEARN algorithm [18] with an additional greedy step. In LEARN the source node S, aiming for destination D, chooses the neighbor N in the restricted neighborhood, such that $\sphericalangle NDS \leq \frac{\pi}{3}$, that has lowest energy mileage, given as $Energy(SN)/|SN|$ where $Energy(SN)$ is the energy needed to send a message from S to N. If there is no such neighbor LEARN-G switches to greedy that does not impose restrictions on the angle.

For each run we have 4 source nodes, situated in the 4 different corners of the network. The total number of nodes in this network is 1 012 with an average node degree of 10. Routing is performed across the diagonals to reach the destination situated in the opposite corner of our simulated network. We measured the total number of hops needed to reach the destination and the average number of retransmissions needed at each step.

We obtained a 100 % success rate for all the routing schemes (all of them are greedy with no face recovery phase) since all generated topologies were of sufficiently high density to allow greedy routing to make advance at each step.

The results are shown in Fig. 3 and are similar to the one presented in previous work [9]. XTC shows the largest number of hops, while LEARN-G has the smallest number of hops, since it is favoring longer links.

The performances of GARE and COP_GARE are situated between LEARN-G and XTC, since they are both using mid-sized edges. The biggest difference between simulation and emulation is for the case of LEARN-G, see Fig. 3(a): it has a slightly smaller number of hops per route for the case of simulation which can be explained by the fact that the algorithm was not constrained to the 1-hop neighborhood, as it was the case with emulation, but it could have also used neighbors outside the 1-hop neighborhood. The situation with the number of retransmissions is also similar, the smallest number of retransmissions is for XTC, GARE has the largest number, among the 3 algorithms used in [9], while COP_GARE performs in between having the lowest overall energy consumption. LEARN-G is using the longest links thus requiring the highest number of retransmissions. The difference in the number of retransmissions between emulation and simulation comes from the already mentioned difference in the links used.

5 Conclusion

Emulation improves the quality of results over simulation by taking into account more accurate environmental data or/and more realistic and complex models to represent sensor nodes or radio propagation. The main novelty in this paper is to imitate the behavior of a real large network deployment by using just small subset of the sensor nodes that would be required. With our previous work [9] we have given insight on the basics of the emulation and as well a proof of the feasibility of the real world implementation. In this paper we go a step further: we improve the emulation so it more closely resembles real world scenarios. Emulated networks can have an arbitrary number of nodes and a desired node distribution and density. Human effort and code complexity remain reasonable, still allowing large scale experimentation.

Our simulations, using the principle of small testbed of 43 WSN sensor nodes show that we can retrieve valuable and repeatable simulation results for a specific type of problems. We have created an environment that is using the same principle as emulation explained in this paper, but with higher consistency of retrieved results. Future work will include experimental results with emulation on a real testbed. Mobility could also be included in our scenarios.

Our emulation approach can also be applied on already deployed large scale sensor testbeds, like GreenOrbs [8], the largest sensor network in the world which consists of 1 000+ deployed sensors. Would this network scale to 1 000 000+ nodes? What changes should be made to the CTP protocol to allow much larger deployments? The bottlenecks of existing large scale deployments could be studied and remedied using an emulation approach, as advocated here, on already large WSN, *e.g.* GreenOrbs would be an ideal platform.

This approach can be used for many other network layer studies, such as broadcasting, geocasting and multicasting for example.

Acknowledments. This work was partially supported by NSERC CRDPJ 386874 - 09 (Reliable and secure QoS routing and transport protocols for mobile ad hoc networks), NSERC Discovery grant of I. Stojmenovic, the French National Research Agency (ANR) project ARESA2 under contract ANR-09-VERS-017, ANR project IRIS under contract ANR-11-INFR-016, and the European Commission project CALIPSO under contract 288879.

References

1. Burin des Roziers, C., Chelius, G., Ducrocq, T., Fleury, E., Fraboulet, A., Gallais, A., Mitton, N., Noél, T., Vandaele, J.: Using SensLAB as a First Class Scientific Tool for Large Scale Wireless Sensor Network Experiments. In: Domingo-Pascual, J., Manzoni, P., Palazzo, S., Pont, A., Scoglio, C. (eds.) NETWORKING 2011, Part I. LNCS, vol. 6640, pp. 147–159. Springer, Heidelberg (2011)
2. Canonico, R., Di Gennaro, P., Manetti, V., Ventre, G.: Virtualization Techniques in Network Emulation Systems. In: Bougé, L., Forsell, M., Träff, J.L., Streit, A., Ziegler, W., Alexander, M., Childs, S. (eds.) Euro-Par Workshops 2007. LNCS, vol. 4854, pp. 144–153. Springer, Heidelberg (2008)
3. Coulson, G., Porter, B., Chatzigiannakis, I., Koninis, C., Fischer, S., Pfisterer, D., Bimschas, D., Braun, T., Hurni, P., Anwander, M., Wagenknecht, G., Fekete, S.P., Kröller, A., Baumgartner, T.: Flexible experimentation in wireless sensor networks. Commun. ACM 55(1), 82–90 (2012)
4. Fall, K.: Network Emulation in the Vint/NS Simulator. In: Proceedings of ISCC 1999, pp. 244–250 (1999)
5. Grau, A., Herrmann, K., Rothermel, K.: Efficient and scalable network emulation using adaptive virtual time. In: International Conference on Computer Communications and Networks, pp. 1–6 (2009)
6. Judd, G., Steenkiste, P.: Using emulation to understand and improve wireless networks and applications. In: NSDI (2005)
7. Ke, Q., Maltz, D.A., Johnson, D.B.: Emulation of multi-hop wireless ad hoc networks. In: Proceedings of the Seventh International Workshop on Mobile Multimedia Communications, MOMUC 2000. IEEE Communications Society (October 2000)
8. Liu, Y., He, Y., Li, M., Wang, J., Liu, K., Mo, L., Dong, W., Yang, Z., Xi, M., Zhao, J., Li, X.-Y.: Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs. In: 2011 Proceedings IEEE INFOCOM, pp. 873–881 (April 2011)
9. Lukic, M., Pavkovic, B., Mitton, N., Stojmenovic, I.: Greedy geographic routing algorithms in real environment. In: MSN, pp. 86–93 (2009)

10. Maier, S., Herrscher, D., Rothermel, K.: Experiences with node virtualization for scalable network emulation. Computer Communications 30(5), 943–956 (2007); Advances in Computer Communications Networks
11. Onat, F.A., Stojmenovic, I., Yanikomeroglu, H.: Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks. Pervasive and Mobile Computing 4(5), 597–615 (2008)
12. Park, S., Savvides, A., Srivastava, M.B.: Sensorsim: A simulation framework for sensor networks. In: Proceedings of MSWiM (August 2000)
13. Senslab. Very large scale open wireless sensor network testbed, http://www.senslab.info/
14. Sobeih, A., Hou, J.C., Lu-Chuan, K., Li, N., Zhang Ning, H., Chen, W.P., Tyan, H.Y., Lim, H.: J-sim: a simulation and emulation environment for wireless sensor networks. IEEE of Wireless Communications 13(4), 104–119 (2006)
15. Stojmenovic, I.: Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions. IEEE of Communications Magazine 46(12), 102–107 (2008)
16. Stojmenovic, I.: Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio. IEEE Network 20(1), 21–27 (2006)
17. Sventek, J., Maclean, A., McIlroy, R., Milos, G.: Xenotiny: Emulating wireless sensor networks on xen. Technical report, University of Glasgow, Department of Computing Science (2008)
18. Wang, Y., Song, W.-Z., Wang, W., Li, X.-Y., Dahlberg, T.A.: Learn: Localized energy aware restricted neighborhood routing for ad hoc networks. In: The 3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON (2006)
19. Wattenhofer, R., Zollinger, A.: Xtc: A practical topology control algorithm for ad-hoc networks. In: IPDPS (2004)
20. WISEBED. Wireless Sensor Network Testbeds, http://www.wisebed.eu/
21. WSN430. Kit developer's guide, http://perso.ens-lyon.fr/eric.fleury/Upload/wsn430-docbook/

Enhancing TCP Congestion Control for Improved Performance in Wireless Networks

Breeson Francis[1], Venkat Narasimhan[2], and Amiya Nayak[1]

[1] School of Electrical Engineering and Computer Science,
University of Ottawa, Ottawa, Canada
{bfran097,nayak}@uottawa.ca
[2] Les Enterprises Norleaf Networks,
Gatineau, Quebec, Canada
narasim@norleaf.ca

Abstract. Transmission Control Protocol (TCP) designed to deliver seamless and reliable end-to-end data transfer across unreliable networks works impeccably well in wired environment. However, when introduced to wireless networks, TCP performance degrades significantly due to the unpredictable nature of wireless environment and the subsequent loss of packets. TCP congestion window (*cwnd*) is considerably reduced because of the ensuing congestion control mechanisms thereby affecting TCP performance critically. In this paper, we propose changes to improve TCP performance by keeping congestion window as large as possible during the various phases of congestion control mechanism. We evaluate the performance using OPNET simulations and show how it fares against the widespread TCP Reno.

Keywords: Wireless TCP, fast retransmission, fast recovery, retransmission timeout, congestion window, duplicate acks.

1 Introduction

Transmission Control Protocol (TCP) forms the backbone for transferring data across the Internet, providing end-to-end congestion control and continuous reliable data delivery services. TCP congestion control mechanism [1] makes sure of end-to-end data delivery without causing congestion en route to destination. TCP makes use of sequence numbering, congestion window and retransmission timer mechanisms to achieve sequential, congestion less and reliable services. TCP sender assigns sequence number for every packet sent and expects an acknowledgement for the same before proceeding with further data transfer. Congestion window (*cwnd*) is used to perform congestion control, which keeps track of the number of packets that can be sent by the sender without being acknowledged by the receiving side. Basically, congestion window decides whether TCP sender is allowed to send packets at any particular instance. TCP accomplishes reliable data delivery by deploying retransmission timer mechanism which detects packet loss and retransmits them. If an

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 472–483, 2012.

acknowledgement is not received before the expiry of retransmission timer, TCP retransmits the packet and triggers congestion control.

Alternate trigger for congestion control mechanism is duplicate acknowledgement (duplicate ACK) arrival at TCP sender. TCP receiver sends a duplicate ACK if the packet is received out of order. When the TCP sender receives duplicate ACKs beyond a certain threshold, it assumes a packet loss and fast retransmission and fast recovery mechanisms are triggered.

During congestion control, TCP Reno reduces the congestion window and abstains from sending packets to avoid further escalation of congestion. When TCP Reno encounters an RTO (Retransmission Time-out), it enters slowstart by reducing congestion window to 1 and slowstart threshold (*ssthresh*) is set to half of existing congestion window. During slowstart phase, *cwnd* is increased additively for every acknowledgement received till *ssthresh* is reached. Then it enters congestion avoidance phase where *cwnd* is linearly increased by 1 maximum segment size (MSS) for every round trip time till a packet loss is detected.

However, as compared to wired networks, in wireless networks a packet loss is *not always* due to congestion, rather they are caused by the inherent unreliable nature of the wireless transmission medium. Several techniques have been proposed to handle the volatile nature of wireless networks [7, 15]. In wireless networks, a packet loss may be due to poor channel conditions, high error rate, high link latency, handoffs, large round trip time etc. Congestion window, which decides the rate of packet transmission, should be handled intelligently during these scenarios. The usual congestion control mechanism, which reduces the congestion window, will only help in decreasing the TCP throughput rather than mitigating the congestion. Mechanisms should be used to keep *cwnd* as high as possible, while keeping the congestion under control. In this article, we present an approach towards a TCP sender side modification during slowstart and fast retransmit and fast recovery phases to keep the congestion window at higher levels so as to utilize the available bandwidth efficiently. Since *cwnd* is retained at higher level it provides an enhanced performance in the wireless scenarios and the simulation results show that the modified TCP performance is considerably healthier than TCP Reno mechanism.

The rest of this paper is organized as follows: Section 2 reviews some of the related work, Section 3 describes the proposed mechanism, Section 4 shows the simulation and performance results and Section 5 concludes the paper with discussions and future work.

2 Related Work

Various schemes [4, 8, 11, 12] have been proposed in wireless networks to improve TCP throughput and to handle congestion indication in such a way that TCP throughput is retained high. Lai et al. in [6] proposed an innovative TCP variant, known as TCP for non-congestive loss (TCP-NCL). TCP-NCL describes a new serialized timer approach in which a new timer, congestion decision timer, is started instead of initiating congestion control mechanisms immediately after a retransmission timeout. If the corresponding acknowledgement is received before congestion decision timer expires, TCP sender continues with the normal data transfer without invoking congestion control mechanisms.

Yongmei et al. in [16] suggest mechanisms to differentiate packet losses due to congestion from wireless losses. In the case of wireless losses, the authors suggest to modify congestion window based on bandwidth-delay product, which indicates the present network load. Bandwidth-delay product is derived from the rate of acknowledgements received and round trip time. Chen et al. [2] propose a receiver-aided mechanism in which the TCP receiver monitors the contention state of the end-to-end connection and the TCP sender is informed about it via the acknowledgement mechanism. TCP receiver uses end-to-end delay as contention to decide network congestion level and the same is notified to TCP sender, which decides the congestion window size based on it.

The authors in [13] suggest that the entire lifetime of a TCP connection is allocated into a finite number of slots and a constant TCP congestion window is used during these slots where the network scenario is assumed to remain unchanged. The beginning of a new slot triggers window recalculation and the congestion window would be set according to the connection's available share in that slot. Elrakabawy et al. [4] proposed a new TCP congestion control algorithm in wireless networks, called TCP with adaptive pacing (TCP-AP). This algorithm measures the change of round-trip delay and determines the network congestion status based on the delay. TCP-AP estimates the appropriate congestion window size according to varying round-trip delays, and the algorithm also introduces rate control during the congestion window adjusting to avoid outburst flows.

Some study has gone into TCP congestion window overshooting problem. Nahm et al. [9] proposed a modified congestion window adaptation mechanism, in which, instead of increasing additively, TCP congestion window is allowed to increase every round trip time (RTT) by a certain fractional rate. Papanastasious et al. [10] proposed a slow congestion avoidance (SCA) scheme. During the congestion avoidance phase in traditional TCP mechanism, the *cwnd* value linearly increases by one maximum segment size (MSS) for every round trip time. In the SCA scheme, authors have defined a new variable *ca_increase_thresh* indicating the number of ACKs received before TCP starts congestion window adaption. *cwnd* value is kept unchanged till the number ACKs received is smaller than *ca_increase_thresh*. Once the number reaches *ca_increase_thresh*, the *cwnd* value is increased by one MSS. This way, the *cwnd* value increases slowly mitigating the congestion window overshooting problem.

Authors in [3, 14] discuss about routing mechanisms for achieving fault-tolerant connectivity in rapidly changing wireless environment.

3 Proposed Mechanism

The main notion of the proposed mechanism is to keep the congestion window as high as possible during congestion control. There are mainly two scenarios when congestion window is reduced. One is during a retransmission timeout (RTO) and the other is when the TCP sender receives a threshold number (usually set to three) of duplicate ACKs.

When an RTO occurs, TCP sender enters slowstart phase, where *cwnd* is reduced to 1 and *ssthresh* is set to half the value of existing *cwnd*. *cwnd* is increased additively till *ssthresh* is reached. When *cwnd* reaches *ssthresh*, it is further increased linearly till a packet loss is detected. The first idea of the proposed mechanism is to set *ssthresh* and *cwnd* to half the existing *cwnd* when a retransmission timeout occur, with a minimum of 2 * maximum segment size (MSS). This keeps the *cwnd* value considerably high so that a sudden decrease in TCP throughput is not observed. By reducing the *cwnd* to half, we reduce the probability of further congestion occurrence at the same time keep its value high enough for efficient use of the available bandwidth. The algorithm for the above mentioned concept is as given below and corresponding flowchart is presented in Figure 1.

```
if (retransmission timeout)
{
  ssthresh = (send_max - send_unacked) / 2;
  if (ssthresh <= 2.0 * send_mss)
  {
      ssthresh = 2.0 * snd_mss;
  }
  cwnd = ssthresh;
}
```

where *send_max* is sequence number of the latest packet sent, *send_unacked* is the sequence number of first unacknowledged segment and *send_mss* is the maximum segment size for outgoing segments.

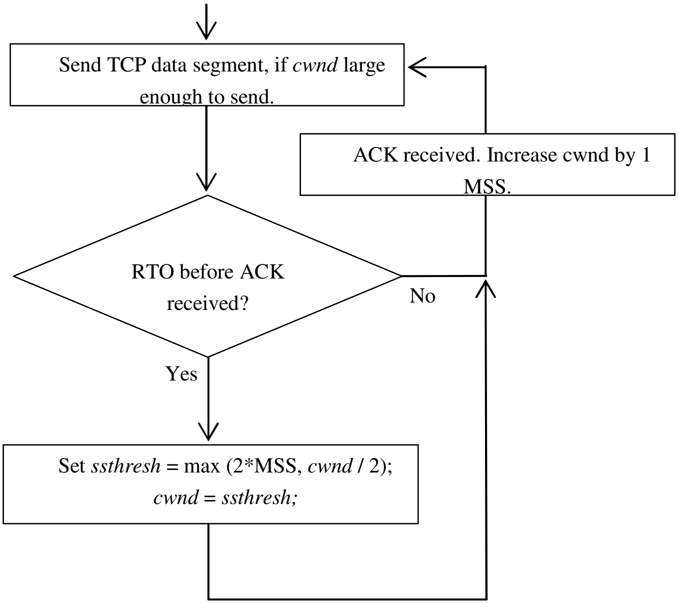

Send TCP data segment, if cwnd large enough to send.
ACK received. Increase cwnd by 1 MSS.
RTO before ACK received?
No
Yes
Set ssthresh = max (2*MSS, cwnd / 2);
cwnd = ssthresh;

Fig. 1. Flowchart for RTO handling

After receiving a threshold number of duplicate acknowledgements (usually set to three), TCP sender enters fast retransmit and fast recovery phase. During Fast Retransmit, the oldest segment in the retransmission buffer for which an acknowledgement is not received will be resent immediately even though no timeout has occurred. After fast retransmission the congestion window is reduced to half of the existing congestion window. Also each duplicate packet is considered an ACK for an already sent packet and congestion window is thus incremented for each duplicate ACK received and a packet is sent if the new congestion window size permits. The second idea of the proposed mechanism is to set *cwnd* to three fourth of the existing *cwnd* instead of half during fast retransmission. This keeps the *cwnd* value in the proximity of where it was before entering fast retransmission and fast recovery phase and avoids sudden decrease in TCP throughput. The algorithm for the above mentioned concept is as given below and the corresponding flowchart is presented in Figure 2.

```
if (threshold duplicate acks received)
{
    ssthresh = (send_max - send_unacked)* 3 / 4;
    if (ssthresh <= 2.0*snd_mss)
    {
       ssthresh = 2.0*snd_mss;
    }
    cwnd = ssthresh;
}
```

where *send_max* is sequence number of the latest packet sent, *send_unacked* is the sequence number of first unacknowledged segment and *send_mss* is the maximum segment size for outgoing segments.

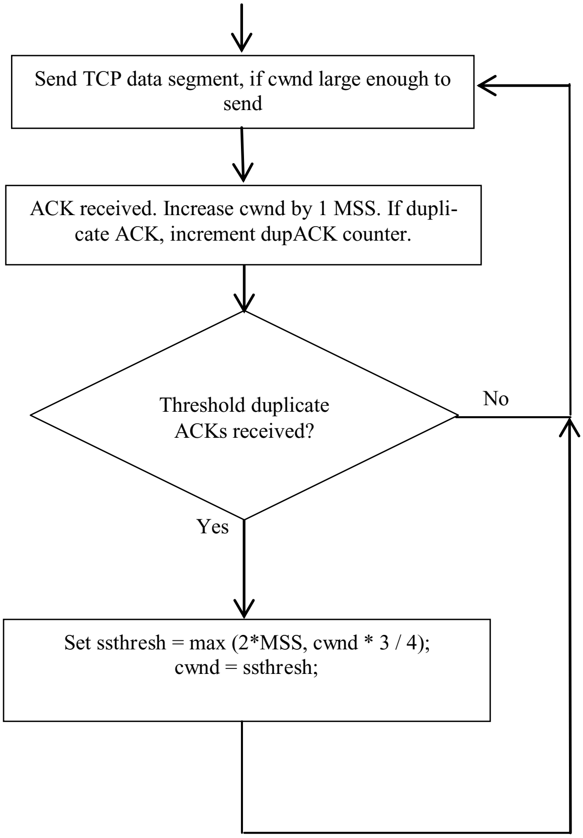

Send TCP data segment, if cwnd large enough to send
ACK received. Increase cwnd by 1 MSS. If duplicate ACK, increment dupACK counter.
Threshold duplicate ACKs received?
No
Yes
Set ssthresh = max (2*MSS, cwnd * 3 / 4);
cwnd = ssthresh;

Fig. 2. Flowchart for Duplicate ACK handling

4 Simulation and Analysis

In this section we describe simulation environment and parameter setting done for performance comparison between TCP Reno and modified TCP with the above mentioned changes. Simulation is carried using OPNET simulator. At MAC layer, standard 802.11g with data rate of 24 Mbps with 256 KB buffer size is used for all the simulations. The maximum number of retransmissions at MAC layer is set to 4 for packets larger than 256 bytes (Long_Retry_Limit) and 7 for other packets (Short_Retry_Limit) as specified in IEEE 802.11 MAC standard. Nodes are using AODV as the routing protocol. At transport layer, TCP Reno with duplicate ACK threshold of 3 and an initial retransmission timeout of 2 seconds is used. File Transfer Protocol (FTP) is used for all the data transfer over TCP. Best effort type of service is used for QoS purpose. Table 1 shows the parameters used for OPNET simulator. Different scenarios with multiple hops and single hop wireless connections are simulated and corresponding TCP throughput and congestion window size is compared.

Table 1. OPNET simulator parameters

Parameter	Value
MAC data rate	24 Mbps
MAC buffer size	256 KB
MAC Long Retry Limit	4
MAC Short Retry Limit	7
Routing Protocol	AODV
TCP Version	Reno
TCP Duplicate ACK Threshold	3
Initial RTO	2 seconds

Single Hop Scenario:

In single hop scenario, two nodes are talking to each other over wireless medium directly with congestion introduced at the sender side. FTP traffic is sent from sender to receiver using both the standard TCP Reno and our proposed TCP as transport layer protocol and compared against each other. From the results we observed that the average throughput is increased when the congestion window is kept at values nearer to where it was before congestion occurred. By using the proposed mechanism, congestion window is retained at higher values and thereby higher TCP throughput is achieved.

Figure 3 and Figure 4 show the TCP throughput and congestion window comparison, respectively, between TCP Reno and TCP with proposed changes, in single hop scenario. TCP throughput obtained using our proposed change is considerably healthier and during our simulations we observed that on an average 20 – 25 % throughput increase is achieved. Figure 4 shows that when proposed TCP is used the congestion window is retained very close to the values before congestion started and simulation results show that on an average, congestion window size is around 30 % higher than that achieved while using TCP Reno.

Multi Hop Scenario

We used a multi hop chain topology with four wireless nodes with congestion introduced at the sender side. Again, FTP is used as the application protocol with TCP Reno and proposed TCP at the transport layer. All the nodes where using same TCP version during simulation. The proposed mechanism aids keeping the congestion window at higher values and thereby higher TCP throughput is achieved. From the results we observed that although the average throughput is increased it is considerably less than single hop scenario.

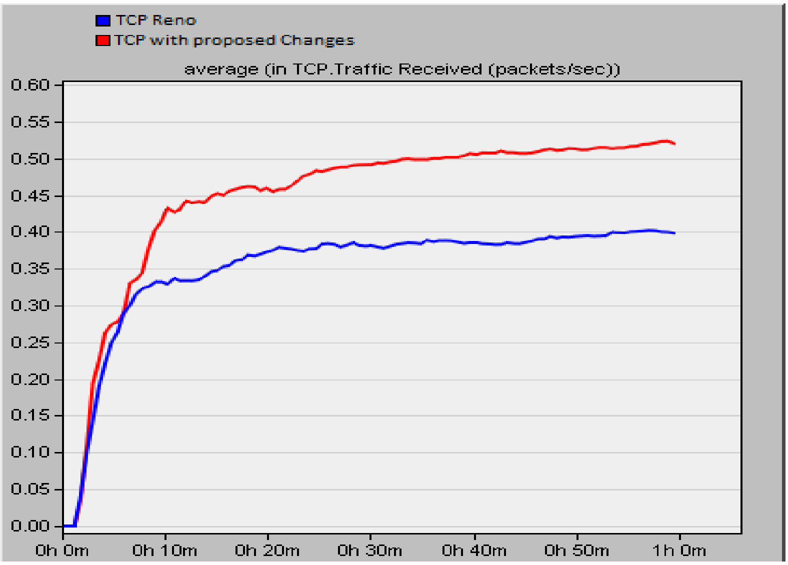


Fig. 3. Single hop throughput comparison

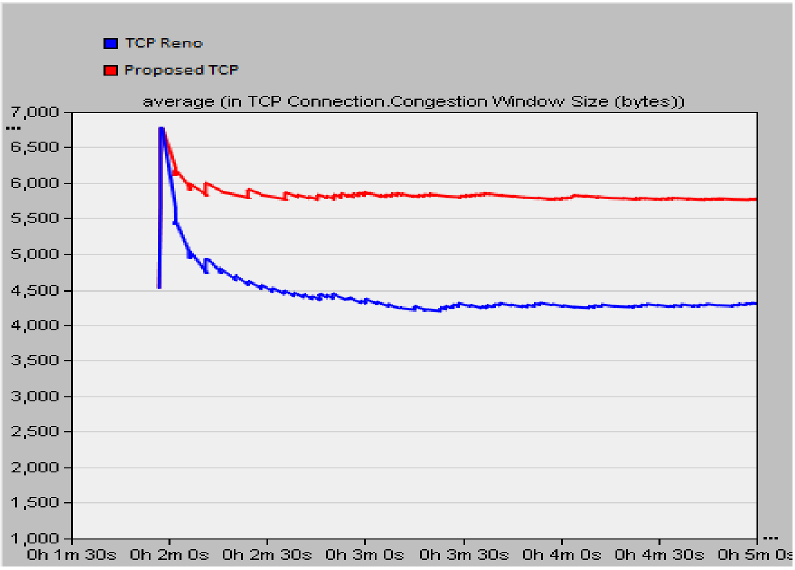


Fig. 4. Single hop congestion window comparison

Figures 5 and 6 show the TCP throughput and congestion window comparison, respectively, between TCP Reno and TCP with proposed changes, in multi hop scenario. During our simulations, proposed TCP in multi hop scenario achieved on an average 35 – 40 % higher throughput than TCP Reno and congestion window is around 45 – 50 % higher than observed with TCP Reno.

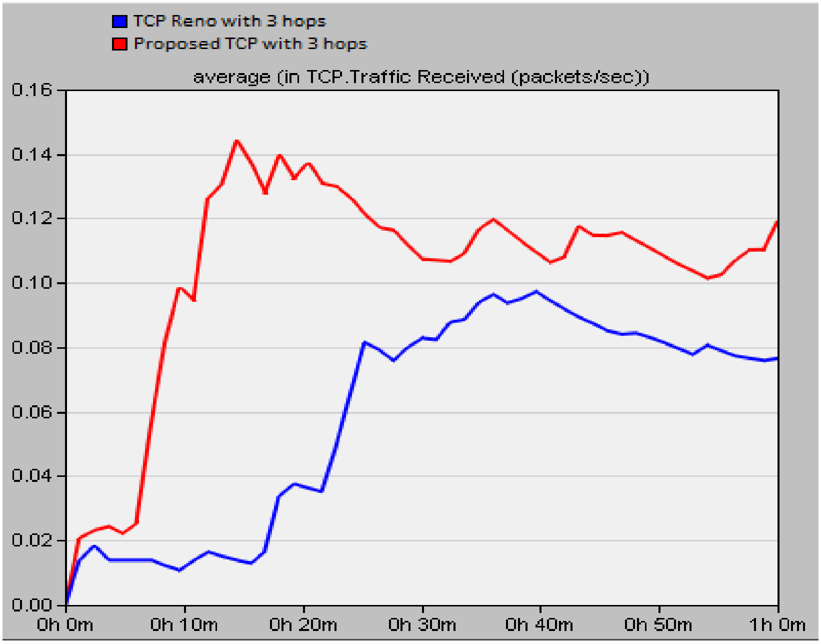


Fig. 5. Multi hop throughput comparison

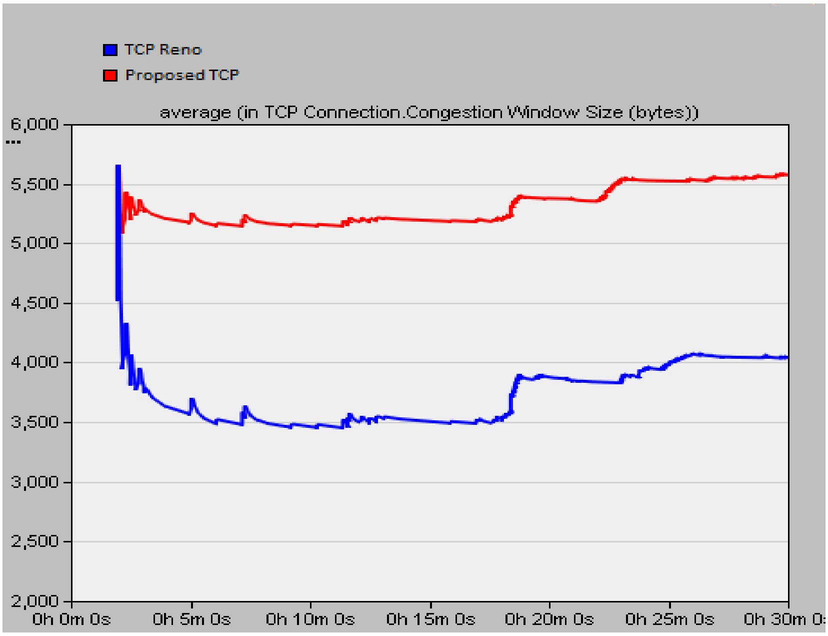


Fig. 6. Multi hop congestion window comparison

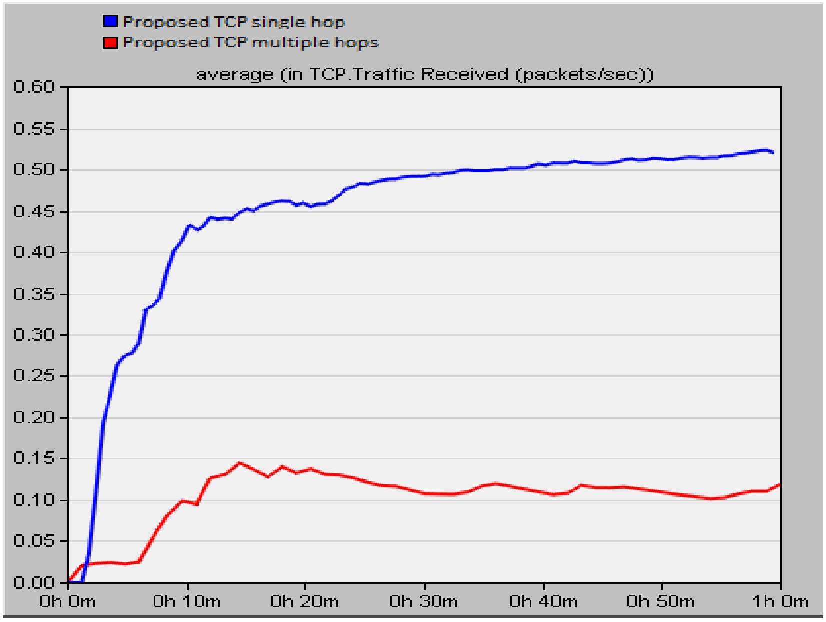


Fig. 7. Single hop vs Multi hop throughput comparison

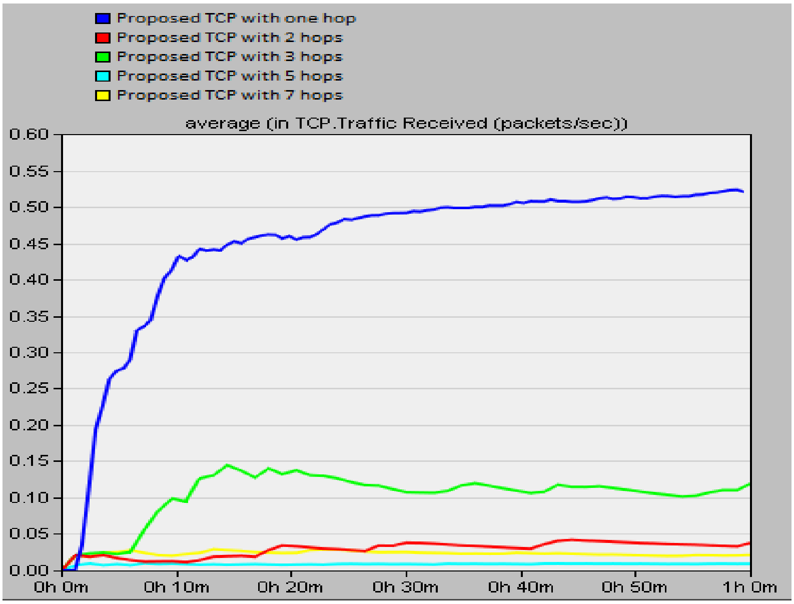


Fig. 8. Comparison of throughput with multiple nodes

It was observed that as the number of hops increased, the congestion increased due to link layer contention (related timers DIFS, SIFS, PIFS) and increased control traffic (RTS / CTS). As shown in Figure 7, the throughput observed is significantly less in multi hop scenario when compared to single hop scenario.

Figure 8 shows a comparison of multi hop TCP throughput with different number of hops. We have compared proposed TCP throughput against 1, 2, 3, 5 and 7 hops. The results show that as the number of hops increases, link layer contention plays a big part in determining the throughput achieved.

5 Conclusions and Discussions

In this paper, we have proposed sender side TCP modifications to improve TCP performance in wireless networks by incorporating two schemes. We have appraised the performance of the proposed schemes with extensive simulations using OPNET. The simulation results have confirmed that the proposed schemes have resulted in significant performance improvement over TCP Reno. TCP Reno was chosen as the base TCP version due to its widespread use and better handling of duplicate ACKs. TCP Tahoe does not support fast recovery and TCP New Reno wrongly triggers fast recovery when 3 or more packets are reordered. The important feature of the proposed scheme was to retain congestion window as near as possible to the value when congestion occurs. Changes were made to handle congestion window calculation differently during retransmission timeout and duplicate acknowledgement arrival. Also, the proposed changes were limited to sender side only so that it is easy to incorporate in existing wireless networks.

The proposed mechanism works particularly well in scenarios where packet drops in wireless medium is significantly high. Since the congestion window is maintained high during the packet drops, the throughput achieved is considerably higher than other TCP variants. However, due to increased link layer contention in multi hop scenario, only increasing the congestion window would not suffice.

During our simulation it was also observed that as the number of nodes involved increased, the throughput reduced significantly due to link layer contention. Our future study will be directed to improve TCP throughput in accordance with link layer contention. In addition, our future work would involve the comparison of the proposed mechanism in this paper with the mechanisms mentioned in [8], [11] and [12].

Acknowledgments. This work was partially supported by the grant, "Reliable and secure QoS routing and transport protocols for mobile ad hoc networks", NSERC CRDPJ 386874-09.

References

1. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. RFC2581 (April 1999)
2. Chen, C., Wang, H., Wang, X., Li, M., Lim, A.O.: A Novel Receiver-aided Scheme for Improving TCP Performance in Multihop Wireless Networks. In: Proc. Int. Conference on Communications and Mobile Computing, pp. 272–277 (2009)

3. Du, J., Kranakis, E., Nayak, A.: A Geometric Routing Protocol in Disruption Tolerant Network. International Journal of Parallel, Emergent, and Distributed Systems 25(6), 489–508 (2010)
4. Elrakabawy, S.M., Klemm, A., Lindemann, C.: TCP with adaptive pacing for multihop wireless networks. In: Proc. 6th ACM Int. Symp. Mobile Ad Hoc Networking & Computing, pp. 288–299 (2005)
5. Francis, B., Narasimhan, V., Nayak, A., Stojmenovic, I.: Techniques for Enhancing TCP Performance in Wireless Networks. In: 9th Workshop on Wireless Ad hoc and Sensor Networks (in 32nd International Conference on Distributed Computing Systems) (2012)
6. Lai, C., Leung, K., Li, V.O.K.: Enhancing Wireless TCP: A Serialized Timer Approach. In: Proc. IEEE INFOCOM (2010)
7. Lin, X., Stojmenovic, I.: Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks. Journal of Parallel and Distributed Computing 63(1), 22–32 (2003)
8. Long, W., Zhenkai, W.: Performance Analysis of Improved TCP over Wireless Networks. In: Proc. 2nd Int. Conference on Computer Modeling and Simulation, pp. 239–242 (2010)
9. Nahm, K., Helmy, A., Kuo, C.J.: Cross-layer interaction of TCP and ad hoc routing protocols in multihop IEEE 802.11 networks. IEEE Trans. on Mobile Computing 7, 458–469 (2008)
10. Papanastasious, S., Ould-Khaoua, M.: TCP congestion window evolution and spatial reuse in MANETs. Wireless Communications and Mobile Computing 4, 669–682 (2004)
11. Prasanthi, S., Chung, S.: An Efficient Algorithm for the Performance of TCP over Multihop Wireless Mesh Networks. In: Proc. 7th Int. Conference on Information Technology, pp. 816–821 (2010)
12. Rai, I.A., Hellen, T.: On improving the TCP performance in asymmetric wireless mesh networks. In: Proc. Int. Conference on Communications, Computing and Control Applications, pp. 1–6 (2011)
13. Roy, R., Das, S., Ghosh, A., Mukherjee, A.: Modified TCP Congestion Control Algorithm for Throughput Enhancement in Wired-cum-Wireless Networks. In: Proc. 4th Swedish National Computer Networking Workshop (2006)
14. Stojmenovic, I., Nayak, A., Kuruvila, J.: A cross layering approach to scalable routing and broadcasting in ad hoc and sensor Networks. IEEE Communications Magazine 43(3), 101–106 (2005)
15. Stojmenovic, I., Simplot-Ryl, D., Nayak, A.: Towards scalable cut vertex and link detection with applications in wireless ad hoc networks. IEEE Networks 25(1), 44–48 (2011)
16. Yongmei, L., Zhigang, J., Ximan, Z.: A New Protocol to Improve Wireless TCP Performance and Its Implementation. In: Proc. 5th Int. Conference on Wireless Communications, Networking and Mobile Computing (2009)

Author Index