

**PROJETO DE REDES AD HOC SEM FIO CIENTE DE
TOPOLOGIA**

HEITOR SOARES RAMOS FILHO

**PROJETO DE REDES AD HOC SEM FIO CIENTE DE
TOPOLOGIA**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: ANTONIO ALFREDO FERREIRA LOUREIRO
COORIENTADOR: ALEJANDRO CESAR FRERY ORGAMBIDE

Belo Horizonte
Agosto de 2012

HEITOR SOARES RAMOS FILHO

**TOPOLOGY-AWARE DESIGN OF WIRELESS AD HOC
NETWORKS**

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

ADVISOR: ANTONIO ALFREDO FERREIRA LOUREIRO
CO-ADVISOR: ALEJANDRO CESAR FRERY ORGAMBIDE

Belo Horizonte
August 2012

© 2012, Heitor Soares Ramos Filho.
Todos os direitos reservados.

R175p Ramos Filho, Heitor Soares
Projeto de redes ad hoc sem fio ciente de topologia
/ Heitor Soares Ramos Filho. — Belo Horizonte, 2012
xxvi, 187 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas
Gerais. Departamento de Ciência da Computação.

Orientador: Antonio Alfredo Ferreira Loureiro.

Coorientador: Alejandro Cesar Frery Orgambide.

1. Computação - Teses. 2. Redes de computadores -
Teses. 3. Sistemas de comunicação sem fio - Teses.

I. Orientador. II. Coorientador. III. Título.

CDU 519.6*22 (043)



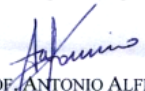
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

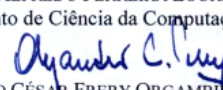
FOLHA DE APROVAÇÃO

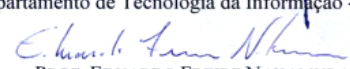
Projeto de redes ad hoc sem fio ciente de topologia

HEITOR SOARES RAMOS FILHO

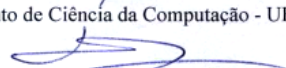
Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

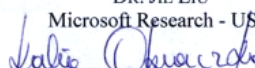

PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Orientador
Departamento de Ciência da Computação - UFMG

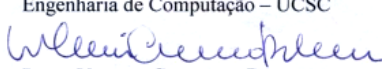

PROF. ALEJANDRO CÉSAR FRERY ORGAMBIDE - Co-orientador
Departamento de Tecnologia da Informação - UFAL


PROF. EDUARDO FREIRE NAKAMURA
Departamento de Ciência da Computação - UFAM / FUCAPI


PROF. GERALDO ROBSON MATEUS
Departamento de Ciência da Computação - UFMG


DR. JIE LIU
Microsoft Research - USA


PROFA. KATIA OBRACKA
Engenharia de Computação – UCSC


PROF. VALMIR CARNEIRO BARBOSA
COPPE - UFRJ


PROF. VIRGÍLIO AUGUSTO FERNANDES ALMEIDA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 17 de agosto de 2012.



Acknowledgments

Foremost, I am grateful to my wife. Karina, thank you for being part of my life, specially during this long and hard journey that culminated with this thesis. We did this together. I have no words to express all love, companionship, support, and everything we are having together. Thanks for everything.

I would like to express my sincere gratitude to my parents that not only gave me birth, but provided me the opportunity of having a solid education. They are my main reference of ethnics, politeness and moral. Thanks for always being such great parents. I also would like to extend my gratitude to all family members that always encouraged me to overcome all challenges posed by this PhD.

I take this opportunity to record my sincere thanks to my advisor and friend Loureiro. I am extremely grateful for his expert, sincere and valuable guidance. He masters the art of encouragement. He is always very positive and optimistic. More often than not he was much more upbeat and confident with my work than me. Thanks for all valuable technical discussions, guidance, advices, and a special thanks for all many opportunities you offered me during this process. Thanks for trusting me for all those challenging missions.

I also would like to thank my co-advisor and great friend Alejandro. We are long-term partners since he was my master's advisor. I have no words to ex-

press all technical support and valuable discussions we had (and still have). Like Loureiro, he masters the art of encouragement. I am really lucky with advisors. Thanks for all support.

I also take this opportunity to thanks my mentor during my internship at Microsoft Research, Jie. For me, this internship was much more than a simple internship mostly because Jie did such a great job supervising me. He set the bar high and guided me to reach great goals. I could not expect to reach such great achievements in a job of only three months. Special thanks to Bodhi, also from MSR, for all valuable collaboration on this work.

My sincere gratitude to Azzedine Boukerche that supervised me during the period I spent in the Paradise Laboratory at the University of Ottawa, Canada. Thanks for accepting me as an exchange student and for supporting me.

During this journey I had the pleasure of meeting and interacting with many people that now I can proudly call friends. They made my journey easier and we spent good times together. Especial thanks for guys from UFMG, Paradise laboratory at uOttawa, and Microsoft Research. All those guys had an intense participation on this journey, some of them I had the opportunity to have some collaboration and some others I had the pleasure to be a friend. I want to record a special thank to the guys I had the opportunity to scientifically collaborate like Richard, Leandro, Eduardo Muccelli, Guidoni, Nakamura, Cristiano, Renfei, Tao, Aman, Bodhi, Ted, and Felipe.

I would like to record a special thank for the guys from the tennis team of Alta Energia in Belo Horizonte. I had a great pleasure playing tennis with them. They made a great effort trying to improve my tennis skills (poor guys).

I also would like to be grateful to the staff of Computer Science department for all support. Special thanks to Renata, Sheila, Tlia and Maristela.

“Anyone who conducts an argument by appealing to authority is not using his intelligence; he is just using his memory.”
(Leonardo Da Vinci)



Resumo

Neste trabalho, estudamos a relação entre métricas topológicas no contexto de redes Ad Hoc sem fio e medidas de desempenho da rede. Neste contexto, estamos interessados na aplicação de diferentes conceitos e métricas relacionadas com a topologia da rede em três modelos de redes distintos: (i) redes de sensores sem fio (WSNs), (ii) redes móveis sem fio (MANETs) e, (iii) redes ad hoc veiculares (VANETs). Esses três modelos cobrem uma grande variedade de topologias, que apresentam diferentes características, desde as WSNs típicas, que não apresentam mobilidade (ou apresentam baixa mobilidade), até redes altamente dinâmicas como as VANETs. As principais contribuições alcançadas são: primeiramente, foi proposto um modelo expressivo de topologias para redes de sensores sem fio, que é apto a descrever uma grande quantidade de estratégias de deposição de nós. Neste contexto, foi proposta uma métrica topológica baseada no betweenness que é capaz de representar o consumo de energia relacionado à tarefa de retransmissão de dados em WSNs. Também foi apresentado um algoritmo distribuído que calcula essa métrica. Esse algoritmo foi utilizado no projeto de um protocolo de roteamento que balanceia o trabalho de retransmissão de dados, aumentando o tempo de vida da rede. No contexto de MANETs, foi desenvolvido um método de localização de nós baseado em GPS que transfere os dados brutos do sinal de GPS para uma plataforma de nuvem, reduzindo o consumo de energia

no dispositivo. Foi demonstrado que, ao aplicar a técnica proposta, foi possível reduzir o consumo de energia em até 80% quando comparado com o GPS tradicional. Para o caso de redes que apresentam alta mobilidade como as VANETs, foi proposta a utilização de técnicas de rastreamento cooperativo para acompanhar as rápidas mudanças de topologia ocasionadas pela alta velocidade dos veículos. Essa solução foi utilizada para aumentar o desempenho de mecanismos de distribuição de vídeo em VANETs.



Abstract

In this work, we study the relationship between topological metrics of Wireless Ad Hoc Networks and the network performance. We are interested in applying different concepts and metrics related to the network topology to three different network models, namely (i) wireless sensor networks (WSNs), (ii) mobile ad hoc networks (MANETs), and (iii) vehicular ad hoc networks (VANETs). These three models cover a wide variety of network topologies, ranging from typically static or nearly static topologies (WSNs) to highly dynamic topologies such as the ones present in VANETs. The main contributions of this work are: firstly, we propose an expressive topology model able to describe a wide variety of deployment strategies for WSNs. We present a topology-related feature estimator derived from the betweenness metric, suitable for representing the energy depletion related to the sensor relay task in WSNs. We developed a distributed algorithm to compute this metric, which was used to design a routing algorithm that aims to make a fair balance of the relay task of nodes in a WSN. For MANETs, we developed a new localization system for Internet capable devices, based on A-GPS technology, which offloads the GPS raw signal data to the cloud. We show that this technique is able to reduce the energy consumption up to 80% when compared to traditional A-GPS. To tackle with the highly dynamic topologies present in VANETs, we proposed the use of a cooperative target tracking solution to track

the quick changes of the topologies due to the high velocity of vehicles and used this solution to improve the performance of a video distribution mechanism over VANETs.



List of Figures

1.1	Wireless networks models	2
1.2	Topology-aware ad hoc node components	5
2.1	Outcomes of M^2P^2 for 300 nodes with 1, 10, 10 and 15 H-sensors (in black) and attractiveness 15, 5, 10 and 15, respectively	19
2.2	Two outcomes of network graphs generated by the M^2P^2 model. Dark points are the H-sensors, gray points are the L-sensors and the triangle is the sink node	20
2.3	Comparison of Q-model and M^2P^2	21
2.4	Three different wireless channel models	26
2.5	Correlation between spent energy and measures of centrality, simple gossip routing and URP deployment	32
2.6	Correlograms and scatterplots for gossip routing and URP deployment, 100 and 400 nodes, centered (C) and randomly (R) placed sink	33
2.7	Correlation between spent energy and measures of centrality, random tree routing and URP deployment	34
2.8	Correlograms and scatterplots for tree routing and URP deployment, 100 and 400 nodes, centered (C) and randomly (R) placed sink	35

2.9	Correlation between spent energy and measures of centrality when gossip routing and Q-Model deployment	37
2.10	Correlograms and scatterplots for gossip routing, Q-Model, 100 and 400 nodes, centered placed sink	38
2.11	Correlation between spent energy and measures of centrality, tree routing and Q-Model deployment	39
2.12	Correlograms and scatterplots for tree routing and Q-Model, 100 and 400 nodes, centered placed sink	40
2.13	Coverage and connectivity as a function of the number of H-sensors	45
2.14	Clustering coefficient and the average path length	46
2.15	Energy consumption metrics as function of the number of H-sensors .	49
3.1	Examples of Betweenness and SBet values for two sink positions, center and corner and their respective histograms (the sink is represented by the triangle)	65
3.2	Node A is more central than node B in terms of number of shortest paths to the sink (the pentagon at the center)	66
3.3	An illustrative network with sink (pentagon) and sensors (circles, and lozenges). For each sensor, we have the SBet value within parentheses, and number of paths from the sink within brackets. The circular-shaped sensors have the Relay role, while the lozenges have the Border role.	70
3.4	Average number packets sent per node upon the hop level	74
3.5	Percentage of transmitted messages as a function of the hop distance	76
3.6	Uneven distribution of transmissions for nodes located one hop from the sink	77
3.7	Relay selection decision rule	79
3.8	Behavior of the randomSbetTree algorithm upon varying the parameter T	83
3.9	Analysis of the number of transmissions	85
3.10	Max number of transmissions upon varying the number of nodes . . .	86
3.11	IQR of transmissions upon varying the number of nodes	87
3.12	Relative entropy of transmissions upon varying the number of nodes	88

4.1	The structure of GPS signal	95
4.2	A schematic view of a GPS receiver: analog and digital signal processing	98
4.3	Navigational data: one frame composed by six sub-frames	100
4.4	Instantaneous power consumption for acquisition phase	108
4.5	Instantaneous power consumption for acquisition, tracking and position calculation phases	109
4.6	Solution ambiguity	111
4.7	The flow of CO-GPS backend web service.	112
4.8	The GSP data collector used for experiments	113
4.9	Duty cycling in experimental evaluation. After an idle period (called a gap), the receiver collects a chunk of raw data.	115
4.10	The number of acquired satellites in various experiment settings. . . .	116
4.11	Location error distribution in various experiment settings when single chunk is used for location calculation	117
4.12	Overall location accuracy distribution.	119
4.13	Overall results from 6 locations. The shadow is 100m in diameter. We see that there are bias errors in some cases.	121
4.14	Error due to time drift	122
4.15	Energy savings from CO-GPS mode in two representative scenarios. .	125
5.1	Cooperative target tracking scenarios	132
5.2	Basic components of cooperative target tracking systems	134
5.3	Non-linear relation between sensor and Cartesian coordinations system	139
5.4	Data association problem	141
5.5	Forwarding zone	150
5.6	Example of multi-modal hypothesis for vehicular state estimation . .	153
5.7	Frame loss	159
5.8	Delay	160
5.9	Cost	161



List of Tables

2.1	Simulation scenarios used in the SBet energy analysis	28
2.2	Simulation scenarios used in the M^2P^2 model analysis	42
2.3	Small world characterization of the M^2P^2 model	47
3.1	Description of variables used in Algorithms 1, 2 and 3	67
3.2	The content of <i>Border</i> packet field <i>sonsPaths</i> , and ψ set for each node of the network shown in Figure 3.3	71
3.3	Necessary overhead to calculate the SBet metric	73
3.4	Simulation scenarios used in the randomSbetTree algorithm analysis	81
4.1	Summary of A-GPS assistance for different type of starts	103
4.2	Scenarios of evaluation for CO-GPS	115
4.3	Error statistics	120
5.1	Summary of motion models	138
5.2	Solutions parameters	158



Contents

Acknowledgments	ix
Resumo	xiii
Abstract	xv
List of Figures	xvii
List of Tables	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Goals	5
1.3 Contributions	6
1.4 Outline	7
2 Modeling and characterization of WSNs	9
2.1 Introduction	10
2.2 Related work	12
2.3 Topology model: the M^2P^2 process	15

2.3.1	Definition of the M^2P^2 stochastic point process	16
2.3.2	Relationship between M^2P^2 and Q-model	20
2.4	Topological characterization: the sink betweenness measure	22
2.4.1	Centrality metrics	22
2.4.2	Evaluation models in the SBet energy analysis	24
2.4.3	Evaluation scenarios used in the SBet analysis	28
2.4.4	Results	29
2.5	Evaluation of the M^2P^2 model	38
2.5.1	Coverage and connectivity	43
2.5.2	Small world characterization	44
2.5.3	Energy balancing	48
2.6	A guide to a stochastic planned deployment	50
2.7	Chapter remarks	52
3	Topology-aware design of WSNs	55
3.1	Introduction	56
3.2	Related work	58
3.2.1	Topology-related algorithms	58
3.2.2	Energy hole	60
3.2.3	Load balance in WSNs	61
3.3	Sink betweenness and wireless sensor networks	62
3.4	Distributed algorithm for sink betweenness	66
3.4.1	Node initialization	69
3.4.2	Dealing with the Hello packet	69
3.4.3	Sending the border packet	69
3.4.4	Dealing with the border packet and calculating SBet	69
3.4.5	Analysis	71
3.5	Sink betweenness and energy hole	75
3.5.1	Methodology	75
3.5.2	Evaluation	79
3.5.3	Results	82
3.5.4	Summary of the results	87
3.6	Chapter remarks	89

4	Low energy GPS-based localization in MANETs	91
4.1	Introduction	92
4.2	GPS basics	93
4.2.1	GPS signal	94
4.2.2	GPS receiver	96
4.2.3	Navigation equations	99
4.2.4	A-GPS	102
4.2.5	Coarse time navigation	103
4.2.6	GPS energy	106
4.3	Our proposal	108
4.3.1	Web services	111
4.3.2	Evaluation	113
4.3.3	Acquisition quality	114
4.3.4	Location accuracy	118
4.3.5	Time accuracy	120
4.3.6	Energy consumption	123
4.4	Related work	125
4.5	Chapter remarks	126
5	Cooperative target tracking in Vanets	129
5.1	Introduction	130
5.2	Problem statement	133
5.3	Components of cooperative target tracking systems	133
5.3.1	Motion models	135
5.3.2	Measurements	138
5.3.3	Data association	140
5.3.4	Communication	142
5.3.5	Filtering	144
5.4	Case study: data dissemination in VANETs	147
5.4.1	CTT-based data dissemination algorithm	148
5.4.2	Target tracking mechanism	151
5.4.3	Performance evaluation	155
5.5	Related work	161

5.6	Chapter remarks	163
6	Final remarks	165
6.1	Conclusions and outlook	165
6.2	Publications	168
6.2.1	Periodicals	168
6.2.2	Conferences	169
6.2.3	Under Submission	170
6.2.4	Short Course	171
6.2.5	Awards	171
	Bibliography	173

Introduction

“Great things are not done by impulse, but by a series of small things brought together”

Vincent van Gogh

1.1 Motivation

Wireless networks consist of a set of nodes that communicate through wireless channels. There are different kinds of wireless networks such as wireless personal area networks (WPAN), wireless local area networks (WLAN), wireless mesh networks (WMESH), wireless metropolitan area networks (WMAN), wireless wide area networks (WAN) and cellular networks. Those networks are characterized mainly by the network range. For instance, WPANs typically connect few devices that span a relative small area within a person’s reach. Conversely, cellular networks connect a large number of mobile devices (mobile phones) and span large areas such cities, continents and even devices in different continents. The challenges to interconnect the devices in each of those networks change abruptly because their characteristics are diverse.

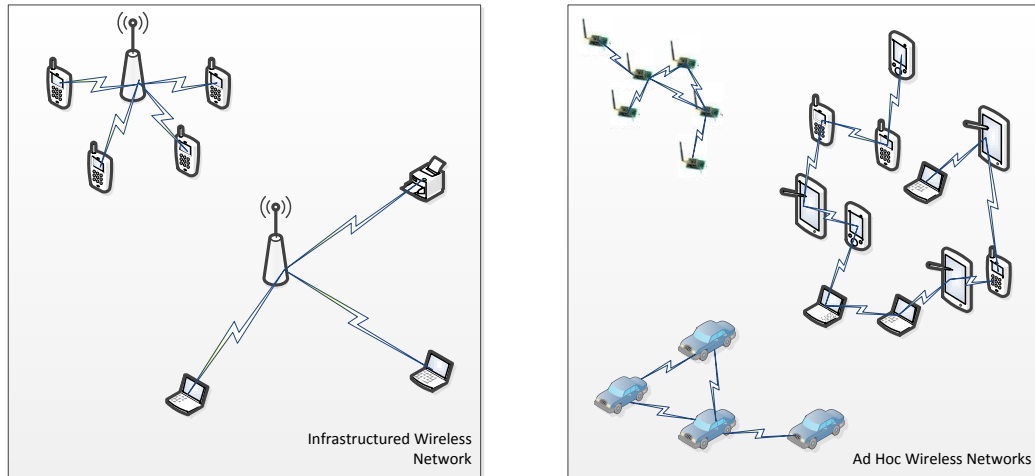


Figure 1.1: Wireless networks models

Some wireless networks require an established infrastructure to work properly. This is the typical case of WLANs where desktop computers, laptops, printers, and other devices connect to an access point to share services. The same situation occurs in cellular networks where the cell towers act as an access point, and are responsible for connecting a limited number of devices within a limited area. Access points may be interconnected, so, devices that are connected to a different access points may be able to communicate. Those networks are heavily dependent on the infrastructure. For example, in the case of a failure in an element of the infrastructure, all devices attached to it are not able to communicate even if they are in the range of other devices.

Conversely, other wireless networks are formed by independent devices that are free to associate to any other device in their range. The devices can act as traffic generators, traffic consumer, or data forwarder, for instance. Communication often occurs in multihop fashion. Those networks are called *ad hoc networks* and have been a hot research topic with many opportunities to be explored [Ramanathan and Redi, 2002; Kiess and Mauve, 2007]. Figure 1.1 illustrates some typical examples of wireless networks that rely on an existent infrastructure, and also wireless ad hoc networks. On the left side, we can observe that mobile

phones and WLAN devices are connected by an access point. On the right side, we can observe three well-known wireless ad hoc networks: wireless sensor networks (WSN), mobile ad hoc networks (MANET) and vehicular ad hoc networks (VANET). Some devices can also work in a hybrid mode, i.e., they work attached to an access point but are able to communicate in ad hoc mode. In this thesis, we are mostly interested in wireless ad hoc networks.

In such networks, the nodes are usually randomly deployed in the workspace. For instance, wireless sensor nodes are typically launched onto a region of interest. MANETs are formed by nodes that move around and form random topologies. The same situation happens in VANETs where vehicles are connected by chance while moving around. Thus, stochastic point processes [Baddeley, 2006; Baccelli, 2009] theory describes the location of a number of points in a region of the space, and is a natural way of representing the random nature of ad hoc networks nodes' location. Nodes are able to communicate only when they are into the communication range of each other. Thus, the network connectivity induces a particular case of random graph [Erdős and Rényi, 1959], namely geographic random graphs.

The topology induced by network connectivity plays an important role in the design and the operation of wireless ad hoc networks. Many properties like coverage, connectivity, lifetime and network congestion are directly influenced by the way nodes are placed in the workspace. For instance, in a WSN scenario, Younis and Akkaya [2008] suggest that the deployment can be optimized in function of area coverage, network connectivity, network longevity and data fidelity.

Moreover, Hoydis et al. [2009] present a study on the effects of the topology on local throughput capacity of medium access protocols in the context of ad hoc networks. They concluded that the way nodes are deployed has strong impact on the local throughput, which is related to network capacity and performance. They point to the need of more complete studies involving different deployment strategies beyond the usual random deployment.

Celebi and Arslan [2007] proposed a location-aware engine architecture for cognitive wireless radio and networks. Their studies unveil that location information can be used to optimize the network performance. Hoydis et al. [2009] referred to that work stating that its results suggest that topological neighbor-

hood information can be used to improve performance. They also stated that further work is called for, i.e., there are open research venues involving the relationships between topology and network characteristics. Their studies are only related to the relationship of throughput and topology.

Haenggi et al. [2009] present a study of the modeling of random nodes location by, for instance, a Poisson point process. They argued that stochastic geometry and random graph theory are indispensable tools for the analysis of wireless networks, and that such tools lead to analytical results on a number of important problems. For instance, they apply those techniques to model and quantify interference, connectivity, outage probability, throughput, and capacity of wireless networks deployed as Poisson point process.

Perillo and Heinzelman [2009] proposed several coverage-aware routing protocols to route traffic around sparsely deployed regions so that the coverage remains high for a long lifetime. Their proposal intends to increase the overall lifetime of the network by avoiding the sparsely sensed regions. A carefully study of that work reveals that routing metrics based on coverage properties are, actually, based on topology metrics (node density).

The aforementioned studies illustrate the importance of the topology in wireless ad hoc networks. This thesis is situated in this context, and is focused on the following research subjects: (i) the identification of topological information/metrics relevant to the design and operation of wireless ad hoc networks, (ii) the estimation of the topological information/metric of interest, and (iii) the design of topology-aware algorithms that take advantage of topological information to improve the network performance.

We envision that a topology-aware ad hoc node presents the components shown in Figure 1.2. As we can observe, there is a topology inference module responsible for collecting data from both sensors and network packets to gather the necessary information to be used by any element of the network protocol stack (application, transport, routing, medium access and physical layers).

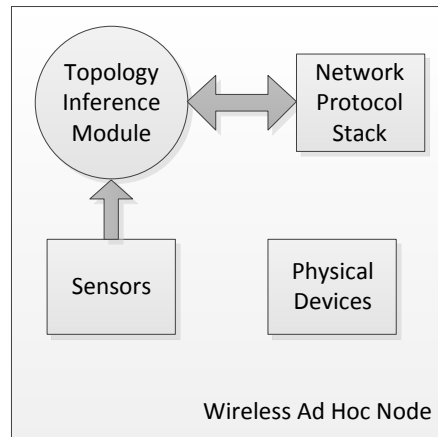


Figure 1.2: Topology-aware ad hoc node components

1.2 Goals

The performance of wireless ad hoc networks is heavily influenced by the way nodes are organized. Thus, topological awareness is a key issue that can be used to improve the performance of those networks. In this context, this work aims to improve the performance of wireless ad hoc networks considering their topology. We started by providing a novel topology model that is able to help understanding the influence of the topology on the design of WSNs. Based on this study, we proposed a new topology metric useful in the design or in the operation of WSNs that is further applied in the design of a novel routing protocol that improves the lifetime of a WSN. Mobile networks pose new challenges on the topology awareness as the topology changes quickly. We observed that node's location is a basic but also an important topology feature. Thus, to tackle mobile networks, we firstly introduced an energy efficient GPS-based algorithm to estimate the node's location and secondly a target tracking scheme that provides up-to-date information of the node's location. Thus, some topological properties can be estimated even for mobile networks with quick topology changes. This study covered a study of the topology from static networks, like traditional WSNs, to high mobile networks such as vehicular networks.

1.3 Contributions

The main contributions of this thesis are:

M²P²: a topological model for WSNs: that network is a special type of an ad hoc network where autonomous devices cooperatively monitor a set of observable phenomena. We propose the Multilevel Marked Point Process M²P², an expressive model able to represent a wide variety of WSNs scenarios, from totally random to planned stochastic node deployment in wireless sensor networks. This model can be easily included in any simulation platform for WSNs.

SBet: a novel centrality metric tailored for WSNs: a centrality metric derived from betweenness, namely Sink Betweenness, SBet for short. We showed that SBet is more suitable for representing WSN characteristics and that it is highly correlated with the energy spent in the relay task of a node in a WSN. Thus, both, M²P² and Sbet are powerful tools on the design space of WSNs. A designer, will be able, for instance, to evaluate how many nodes should be deployed, how many low- and high-end nodes (nodes with more powerful battery and communication radius) should be deployed to increase the network lifetime. SBet can also be used to design topology-aware algorithms that benefits from the knowledge of this metric in order to improve the network performance.

CO-GPS: a low energy GPS-based localization for MANETs: location is one of the most basic topological features, however, it is also one of the most useful for ad hoc networks. With the advent of mobile devices, location based applications has emerged as a new trend. A wide variety of applications have used the location information in order to improve the user experience, thus, we propose a highly power-conserving kind of GPS solution, namely Cloud-Offloaded GPS, tailored for mobile devices. We showed that CO-GPS decreases the energy consumption of GPS location tags by moving most part of energy- and CPU-demanding tasks to the cloud.

A cooperative target tracking module for VANETs: a module that accommodates all necessary features for tracking targets in vehicular networks. Target tracking plays a key role for vehicular ad hoc networks (VANETs) due to the fact that a wide variety of envisioned applications rely on the ability of detecting, localizing and tracking objects (for instance, other vehicles) surrounding a given vehicle. Thus, vehicles are able to manage the highly dynamic topologies present in this kind of network by tracking the state (localization, velocity, acceleration, etc) of surrounding nodes.

CTTDD: a location-aware multimedia data dissemination for VANETs: a data dissemination scheme for VANETs, namely Cooperative Target Tracking based Data Dissemination algorithm (CTTDD), that takes advantage of the target tracking module. In our approach, the nodes only store the states of the neighbors that are directly involved in the data communication process. Our algorithm uses the neighbors' location and near-future predictions to drive the relay selection process.

1.4 Outline

This document is organized as follows. Chapter 2 presents the M^2P^2 topology model for WSNS, and also proposes the Sink Betweenness metric (SBet). Chapter 3 presents a distributed algorithm to estimate the SBet metric for all nodes in a WSNs. We also propose a routing algorithm that uses the SBet metric in order to improve the lifetime of WSNs. Chapter 4 presents CO-GPS, our solution for localizing energy-constrained devices by using a low energy GPS-based localization solution. In Chapter 5, we propose the target tracking module for VANETs and the CTTDD, a data dissemination algorithm that takes advantage of the target tracking module to improve throughput and decreases the delay of multimedia data transmission in VANETs. Finally, in Chapter 6, we provide our final thoughts, an outlook of this work, and the publication list related to this thesis.

Energy-aware topology modeling and characterization of wireless sensor networks

“All our knowledge has its
origins in our perceptions”

Leonardo da Vinci

Heterogeneous wireless sensor networks were proposed to address some fundamental limits and performance issues present in homogeneous Wireless Sensor Networks (WNS). The use of a set of high-end sensors may lead to increase some WSNs capabilities in different ways. Thus, the network can be comprised, for instance, of two different set of nodes, namely low- and high-end nodes (L- and H-sensors, respectively). High-end nodes usually present higher battery capacity and higher transmission ranges (more powerful radio), while low-end nodes are regular WSN nodes. Questions as, for instance, how many high-end sensors should be used and how to plan their deployment need a proper assessment. In this work, we propose a novel modeling solution that is able to represent a wide variety of scenarios, from totally random to planned stochastic node deployment in heterogeneous sensor networks. This model encompasses homogeneous and heterogeneous networks showing characteristics of small-world networks and can address the energy hole problem. We show that using only about 3 % of high-end sensors and deploying nodes by using the

slightly attractive model herein defined, we observe improved characteristics of the network topology, among them: (i) low average path length, (ii) high clustering coefficient and (iii) fairly relay task distribution among the sensors. We also provide a comprehensive guide of how to deploy nodes to improve the lifetime by diminishing the energy hole effect by using topological metrics. Moreover, we evaluate a topological metric, namely Sink Betweenness, suitable for characterizing the relay task of a node. We show that this measure is highly correlated with energy consumption in a wide variety of typical scenarios, while classical measures, such as Betweenness, Closeness, degree, Eccentricity, for instance, do not exhibit this desirable property. Sink Betweenness can be used in adaptive algorithms to alleviate the energy hole effects. The Sink Betweenness can also be used in other applications such as to build a routing infrastructure that favors the data fusion process [Oliveira et al., 2010].

2.1 Introduction

Wireless Sensor Networks (WSNs) are ad hoc wireless networks consisting of spatially distributed autonomous devices that cooperatively monitor environmental conditions such as temperature, pressure, and pollutants, among other applications. WSNs have been studied in various application areas (e.g., health, military, home) [Akyildiz et al., 2002; Culler et al., 2004] where human presence is not possible nor desired [Cui et al., 2006; Younis et al., 2006].

The sensors scattered in a sensor field have the capability of collecting and aggregating data, and routing them to a base station (also called sink node) [Akyildiz et al., 2002]. The sink also connects the WSN with other networks such as the Internet.

Node deployment, and the consequent induced topology, plays an important role in the design of wireless sensor networks. Many important properties such as coverage, connectivity, data fidelity, and lifetime are directed influenced by the way nodes are placed in the sensor field.

Most WSN models in the literature assume that the network is comprised of homogeneous nodes, i.e., all sensors have the same capabilities in terms of energy, processing, memory, and communication. However, Yarvis et al. [2005]

show that homogeneous ad hoc networks suffer from fundamental limitations and, hence, exhibit poor network performance such as end-to-end success rate, latency and energy consumption. Another class of WSN models assumes that there are different sets of nodes, each one with different capabilities. For instance, suppose we have two sets of nodes: the first one comprised of a small number of powerful high-end sensors (H-sensors), and the second one of a large number of low-end sensors (L-sensors). In this case, we have a Heterogeneous Sensor Network model [Yarvis et al., 2005].

Wu et al. [2008] show that the lifetime of a uniformly deployed WSN is strongly limited by the sensors at the first hop from the sink, a problem known as “energy hole”. This problem follows from the relay task that is more concentrated on nodes that are placed close to the sink node, when data collection algorithms are used. The energy hole problem is also present in heterogeneous networks. In this case, it appears in the neighborhood of each H-sensor and the sink. The authors conclude that just randomly increasing the number of nodes cannot desirably prolong the network lifetime when a totally random deployment is used. They show that the entire network lifetime can be improved by spreading more nodes nearby the sink.

An important task in the development of energy-aware solutions for WSNs is the design of efficient techniques for the creation of heterogeneous networks topologies with specific properties. Complex networks [Newman, 2003] can be used to model a network that has certain non-trivial topological features such as heavy-tailed degree distribution, high clustering coefficient, community structure at different scales, and evidence of a hierarchical structure. The two most well-known examples of complex networks are those of scale-free and small-world. In a scale-free network, a vertex degree obeys a power law distribution, while a small-world network has a high clustering coefficient and a small path length [Newman, 2003]. Small-world networks present interesting characteristics w.r.t. data communication in a computer network [Helmy, 2003]. To create a network with small-world features, the designer should add a small number of long-range links, called shortcuts.

In this work, we introduce a novel deployment model for WSNs. Some topologies that may be represented by this model exhibit desirable characteris-

tics for WSNs. We show that a proper planned stochastic deployment leads to improve the network performance by means of shorter average path length and higher cluster coefficient (which can improve the fault-tolerance properties). Besides, an appropriated deployment can properly address the energy hole problem. We also introduce and evaluate a new metric, namely Sink Betweenness [Oliveira et al., 2010; Ramos et al., 2011a], which is able to characterize the energy hole problem and can be used in the design of WSNs algorithms.

Next sections are organized as follows. Section 2.2 discusses the related work that motivates this research. Section 2.3 introduces the M^2P^2 deployment model herein proposed. Section 2.4 presents a new topological metric that is useful for characterizing the energy consumption due to the relay task in WSNs scenarios. Section 2.5 assesses and characterizes some of the topologies that can be described by the M^2P^2 model. Section 2.6 presents a comprehensive guide to a planned deployment that can be described by the M^2P^2 model. Finally, Section 2.7 presents some concluding remarks and future directions for this work.

2.2 Related work

Younis and Akkaya [2008] present a comprehensive survey on strategies and techniques for node placement in WSNs. In that survey, they propose a classification for different deployment methods. The first criterion is whether a node is static or mobile. In case of static nodes, they consider two deployment strategies: controlled and random. A controlled deployment is usually appropriate to indoor applications whenever the designer is able to specify the placement of all nodes, whereas a random location is usually pursued for applications where the designer is not able to exactly place the sensor nodes. The latter scenario assumes that sensors will be randomly placed. For instance, they could be dropped by a helicopter or an airplane. They also suggest that the deployment can be optimized in function of: (i) area coverage, (ii) network connectivity, (iii) network longevity, and (iv) data fidelity, while nodes can assume the following roles: sensor, relay, cluster-head, and base station.

The induced topology has influence on the most important characteristics of the WSNs. Despite this fact, studies involving different topologies are seldom

found in the literature (see, for instance, Frery et al. [2010]). Actually, uniform random placement (URP) is the most used deployment strategy in WSN simulations [Younis and Akkaya, 2008; Wang et al., 2008]. Let us define the deployment model as a stochastic point process, i.e., a probability law able to describe the location of a number of points in a region of the space. For the sake of simplicity, let us assume that we are interested in stochastic point processes on the compact window $W = [0, \ell]^2 \subset \mathbb{R}^2$, where ℓ is the side length of the sensor field. A fixed number of n points obeys a URP distribution on W if they are placed uniformly and independently of each other. A sample from such process can be built by observing outcomes from $2n$ independent identically distributed random variables $X_1, \dots, X_n, Y_1, \dots, Y_n$, obeying the uniform law on $[0, \ell]$, say $x_1, \dots, x_n, y_1, \dots, y_n$, and then placing the n points on coordinates $(x_i, y_i)_{1 \leq i \leq n}$. Younis and Akkaya [2008] state that the URP assumption can be unrealistic or even undesirable for WSNs scenarios.

Hoydis et al. [2009] present a study on the effects of the topology on local throughput capacity of medium access protocols in the context of ad hoc networks. They conclude that deployments based on the URP hypothesis have strong impact on the local throughput, which is related to network capacity and performance. They use cluster-based point processes [Baddeley, 2006] to conclude that simulations and analytical calculation which are done by using simple URP hypothesis can lead to incomplete findings and insights. They point to the need of more complete studies involving other point processes beyond the URP model.

Haenggi et al. [2009] present a study of the modeling of random node location based on, for instance, a Poisson point process. They argue that stochastic geometry and random graph theory are indispensable tools for the analysis of wireless networks, and that such tools lead to analytical results on a number of concrete and important problems. For instance, they apply those techniques to model and quantify interference, connectivity, outage probability, throughput, and capacity of wireless networks deployed following the URP model. They use techniques based on stochastic geometry and on the theory of random graphs, including point process theory, percolation theory and probabilistic combinatorics, to state the importance of the cross-disciplinary dialogues among engineering, computer science and applied probability areas. They also present some possible

future research issues involving those subjects.

It is of paramount importance to assess the behavior of protocols and algorithms for WSNs considering different topologies and scenarios. For the sake of illustration, the energy hole is a fundamental problem that limits the lifetime of WSNs. A similar behavior is perceived in heterogeneous networks, however, the energy hole is observed in nodes that are either close to the sink or the H-sensors.

Mohapatra [2005]; Li and Mohapatra [2007] present the first mathematical model towards the characterization of the energy hole problem. The authors considered sensor nodes distributed following the URP law (see Section 2.2) in a circular region divided in concentric coronas. They observed the impact of four factors: node density, hierarchical deployment, source bit rate and traffic compression. Based on these observations, they shown that simply adding more nodes in the network does not solve the problem, which using hierarchical deployment and data compression can mitigate it, and that increasing the bit rate leads to worse results.

Olariu and Stojmenovic [2006] present a strategy to mitigate the energy hole problem, considering the URP deployment in which the nodes' energy consumption satisfy the relation $C = d^\alpha + c$, where C is the energy consumed, $\alpha \geq 2$ is the power attenuation d is the Euclidean distance between sender and receiver nodes, and c is a device-dependent positive constant. In this work the authors show that for $\alpha = 2$ no routing strategy can avoid the energy hole problem. On the other hand, they argued that for $\alpha > 2$ suboptimal solutions can be reached.

Liu et al. [2007] propose a different approach to the energy hole problem; they consider nonuniform node deployment. They then derive a placement function based on the distance to the sink, in hops. An extension of this idea is presented by Wu et al. [2008], who show that nearly balanced energy depletion is possible by increasing the density in geometric progression from the outer to the inner coronas. Based on this fact, they propose a nonuniform node distribution strategy: the Q-Model (see Section 2.3.2).

Differently from the aforementioned approaches, in this work, we present a novel deployment model for WSNs that leads to a planned but not totally controlled (i.e., not deterministic) topologies that presents a small-world behavior and properly addresses the energy hole problem. We assess a wide variety of

scenarios that can be described by this model in terms of (i) coverage, (ii) connectivity, (iii) small-world characteristics and (iv) energy hole behavior. We show that with a planned stochastic deployment the addition of only 3% of H-sensors the generated topology improves the network performance by means of a better average path length (shorter paths), a higher cluster coefficient (that can improve the fault tolerance properties) and reduces the energy hole problem. We also show that betweenness is able to characterize the energy hole problem and can be used to guide the network design and deployment.

2.3 Topology model: the M²P² process

A stochastic point process is a probability law that describes the location of a number of points in a region of the space. For the sake of simplicity, let us assume that we are interested in stochastic point processes on the compact window $W = [0, \ell]^2 \subset \mathbb{R}^2$, where ℓ is the side length of the sensor field. A fixed number of n points obeys a binomial distribution on W if they are placed independently of each other. A sample from such process can be built observing outcomes from $2n$ independent identically distributed random variables $X_1, \dots, X_n, Y_1, \dots, Y_n$, obeying the uniform law on $[0, \ell]$, say $x_1, \dots, x_n, y_1, \dots, y_n$, and then placing the n points on coordinates $(x_i, y_i)_{1 \leq i \leq n}$.

If the number of n points is the outcome of N , a random variable following the Poisson distribution with parameter (mean) $\lambda > 0$, i.e., $\Pr(N = k) = e^{-\lambda} \lambda^k / k!$ for every $k \in \mathbb{N}_0$ and $N(\omega) = n$, $\omega \in \Omega$ an arbitrary event, points are placed according to a binomial point process on W , we then have a Poisson point process with intensity λ on W . This distribution process is regarded to as one of the basic tools in the theory and practice of point process since it describes complete randomness.

Several properties stem from the aforementioned constructive definition provided for Poisson point processes, some of them being equivalent definitions as, for instance, the following two:

PPP1 The number of points in every compact set $A \subset W$, denoted by $C(A)$ for “counts”, follows a Poisson distribution with mean $\lambda \mu(A)$, where $\lambda > 0$ is

called “intensity” and $\mu(A)$ is the area of A .

PPP2 If $A_1, A_2, \dots, A_m$ are disjoint subsets of W , then $C(A_1), C(A_2), \dots, C(A_m)$ are collectively independent random variables.

The connection between the Poisson and binomial processes is established by the *conditional property*: if a Poisson process on W has intensity λ and knowing that $C(W) = n$, then the distribution of the number of points $C(A)$ in any $A \subset W$ follows a binomial distribution, i.e., $\Pr(C(A) = k \mid C(W) = n) = \binom{n}{k} p^k (1-p)^{n-k}$, $0 \leq k \leq n$, where $p = \lambda\mu(A)/\mu(W)$.

An important generalization is obtained by varying the intensity λ suitably on W . In order to do so, we define the bounded positive function $\lambda: W \rightarrow \mathbb{R}_+$, called “intensity function”, and replace property PPP1 above for the following:

PPP3 The number of points in every compact set $A \subset W$, denoted by $C(A)$, follows a Poisson distribution with mean $\beta = \int_A \lambda(u) du$.

We then have a (possibly inhomogeneous) Poisson point process, which reduces to the basic Poisson process whenever β is the area of $A \subset W$, i.e., when the intensity is constant. In this last case, the n points are *conditionally* independent, given the function λ .

2.3.1 Definition of the M^2P^2 stochastic point process

In the WSN context, such inhomogeneous process can be used to specify some areas with more concentration of points (sensors) by using the intensity parameter in an appropriated manner. For instance, we will build a stochastic point process suitable for describing the deployment of inhomogeneous WSNs by choosing the intensity function in such a way that the energy hole phenomenon is alleviated. For this, the intensity function will concentrate more points in regions near both the sink and the H-sensors, and less in other regions.

We start this process by first placing m H-sensors on W , and then deploying the remaining $n - m$ sensors “close” to them. Denote the coordinates of the m H-sensors by $\mathbf{h} = \{(hx_1, hy_1), \dots, (hx_m, hy_m)\}$ (how these sensors are placed will be described later).

Without loss of generality, in the following, we consider the intensity function λ , which has increased but constant intensity around selected spots:

$$\lambda(x, y) = \begin{cases} a, & \text{if } d((x, y), (hx_i, hy_i)) \leq r_c, 1 \leq i \leq m, \\ 1, & \text{otherwise.} \end{cases} \quad (2.1)$$

where $a \geq 1$ (the attractiveness parameter), d is any distance measure, and r_c is the communication radius of the low-end sensors (L-sensors). In the remaining of this work, we employ the Euclidean distance but any suitable distance measure may be used to enhance realism. Denote such process by $\Lambda(n - m, a, \mathbf{h})$.

Notice that a stochastic point process defined by an intensity function λ as the one in Equation (2.1) has overall mean intensity given by $\int_w \lambda$. If $a > 1$, then it is more likely to have points around the m coordinates where there is an H-sensor; if A_1 belongs to the area of influence of an H-sensor and A_2 does not, but $\mu(A_1) = \mu(A_2)$, on average there will be a more sensors in the former than in the latter subset. As defined, two or more H-sensors that are arbitrarily close will behave as a single H-sensor for the deployment of L-sensors, since the Λ process favors the occurrence of the latter as a function of the distance to the former.

Note that if $\lambda(x, y) = \lambda$, the inhomogeneous Poisson point process becomes the basic Poisson point process, i.e., it reduces to complete randomness. Samples from inhomogeneous Poisson point processes can be conveniently obtained by using the `rpoispp` function available for the R package [R Development Core Team, 2009] in the `spatstat` library, being the intensity function the only mandatory parameter.

In heterogeneous WSNs, H-sensors are useful to provide long-range shortcuts and diminish the number of hops to reach the sink node. They have a high-powerful radio that is able to communicate in long-range distances and a high-capacity battery that increases their lifetime. Those features make the H-sensors more expensive than the other nodes. It is desirable that the deployment of the H-sensors be made in such way that it diminishes the amount of H-sensors close to each other and, thus, decreases the total amount of H-sensors required to create the appropriate shortcuts. In real-world deployments it can be done by launching the H-sensors far from each other in a convenient distance. The re-

sulting topology is planned but does not require a deterministic placement of the nodes. Accordingly, we use a stochastic repulsive deployment of the H-sensors as described below.

The SSI (Simple Sequential Inhibition) stochastic point process [Baddeley, 2006] is a convenient model for the repulsive deployment of sensors. This process is defined on a window W by the maximum number of m points and an inhibition distance d . The first of the m points is placed in W obeying a binomial process. At each subsequent iteration, a new point is placed in W and it is accepted only if all other previous points lie further than d , otherwise it is rejected. The procedure stops either when the m points have been placed or when a maximum number of iterations is reached. Clearly, if $d > \ell/m^{1/2}$ it will be impossible to place all the points in $W = [0, \ell]^2$. Smaller inhibition distances do not guarantee that there will be all the m points, unless d is negligible. This is the process that places at most m non-overlapping disks of radii $d/2$ on W . There are richer repulsive point processes, where there is no strict inhibition as, for instance, the Strauss process [Baddeley, 2006]; the SSI will suffice for our purposes, and it will be denoted by $H(m, 2r)$, for *hardcore*.

We are now ready to define the Multilevel Marked Point Process M^2P^2 .

Definition 2.3.1 ($M^2P^2(m, n, a, r_c, r_i)$ on $W \subset \mathbb{R}^2$) Consider a number $m \geq 1$ of H-sensors over a total of $n > m$ sensors, the intensity $a \geq 1$ of L-sensors on a circle or radius $r_c > 0$ centered at each H-sensor (r_c is the communication radius among L-sensors) and inhibition radius $r_i > 0$ among H-sensors. Thus, M^2P^2 is a compounded process of m samples of $H(m, 2r_i)$ (the H-sensors) and $n - m$ samples of $\Lambda(n - m, a, \mathbf{h})$ (the L-sensors), $\mathbf{h}$ is the set of m coordinates of the H-sensors.

Firstly, take a sample from an $H(m, r_i)$ process with exactly m points: the coordinates of the m H-sensors. Secondly, return the outcome of an inhomogeneous binomial point process through the intensity function λ defined in Equation (2.1) using as $\mathbf{h}$ the m coordinates obtained in the first step, and take a sample of $n - m$ points by using $\Lambda(n - m, a, \mathbf{h})$.

Figure 2.1 shows four outcomes of the M^2P^2 process with 300 nodes inside an area of 100×100 square units: 1 (first), 10 (second and third), and 15 (forth) H-sensors (dark points) with attractively deployed L-sensors around the

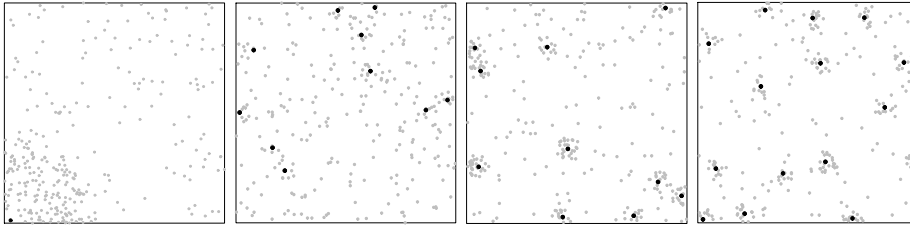


Figure 2.1: Outcomes of M^2P^2 for 300 nodes with 1, 10, 10 and 15 H-sensors (in black) and attractiveness 15, 5, 10 and 15, respectively

H-sensors with attractiveness parameters 5 (first), and 15 (third and forty). The leftmost outcome shows a homogeneous network ($m = 1$) where the darker point represents the sink node. The other figures show three heterogeneous WSNs. If $a = 1$, no attractive behavior around the sink and the H-sensors is taken.

The M^2P^2 process can be extended in two ways, namely the deployment of the H-sensors and the deployment of the simple nodes (L-sensors). It is also immediate to generalize it to higher dimensions.

A sample from the M^2P^2 process is just a set of marked points. The connectivity radii among L- and H-sensors, r_c and r_{ch} respectively, induce a network topology.

Figure 2.2 shows two outcomes of network graphs induced by the M^2P^2 model. Black points represent H-sensors, gray points represent L-sensors, the triangle represents the sink node. Edge colors follow the node type. There are 1000 sensors, being 30 H-sensors and 970 L-sensors deployed in a 1000×1000 sensor field. The communication radii are $r_c = 50$ and $r_{ch} = 300$ for L- and H-sensors, respectively, and $a = 5$. Observe that heterogeneous WSNs like the ones present in Figure 2.2 can be seen as an union of homogeneous WSNs, where the H-sensors act like sinks and are able to communicate in a different wireless channel acting as an overlay network (or even can be extended to communicate using another technology such as Internet, cable, etc.). Those kind of networks can also be seen as a multi-sink approach, considering the sink a more powerful element. This model is more likely to be used in large scale WSNs as it scales like a hierarchical WSNs. In such networks, the energy hole problem happens in the L-sensors. We are disregarding the energy hole in the H-sensors as we assume they present a powerful battery or they can be plugged to a continuous energy

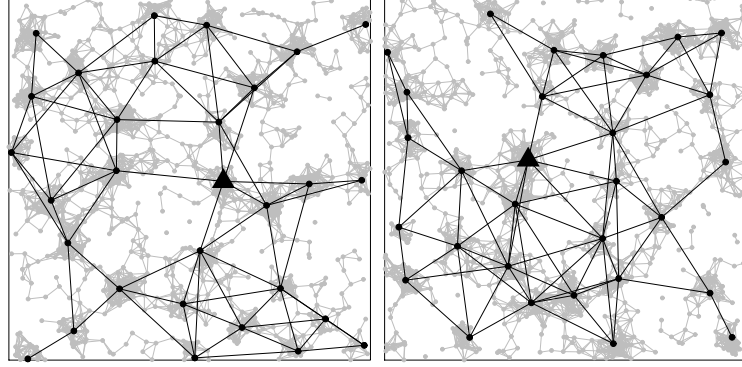


Figure 2.2: Two outcomes of network graphs generated by the M^2P^2 model. Dark points are the H-sensors, gray points are the L-sensors and the triangle is the sink node

source.

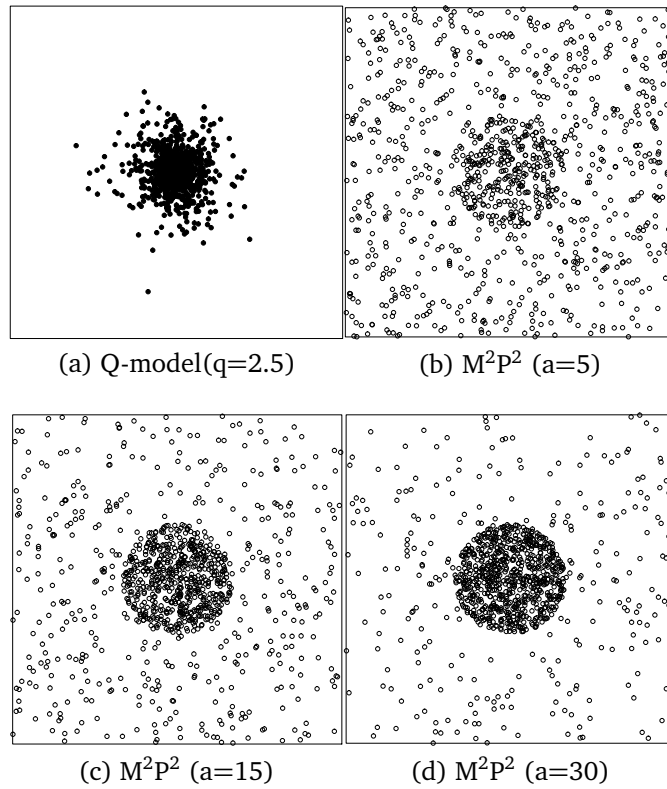
2.3.2 Relationship between M^2P^2 and Q-model

The Q-model was defined by Wu et al. [2008] and aims at alleviating the energy hole effect by increasing the density of nodes closer to the sink. The authors propose a geometric law for the density of nodes of the form $f(r) \propto (rq^r)^{-1}$, with $q > 1$ (hence the name of the model), and r the distance to the sink. This model concentrates more nodes close to the sink when q increases. Following the authors, in our studies we used $q = 2.5$. Figure 2.3 illustrates different outcomes of Q-model under $q = 2.5$ and M^2P^2 under three different values of a . In this plot, there are 1000 nodes in a square field of side 300 m. Observe that M^2P^2 tends to spread the nodes all around the field even though when a increases those spread nodes becomes less dense.

Although Figure 2.3 shows that Q-model produces topologies quite different from the ones produced by M^2P^2 , Q-model can be view as a instance of M^2P^2 . For this, let us modify the Equation 2.1 to

$$\lambda(x, y) \propto (rq^r)^{-1}. \quad (2.2)$$

M^2P^2 reduces to Q-model when λ is defined as Equation 2.2 and there is

Figure 2.3: Comparison of Q-model and M^2P^2

no H-sensor (only the sink node). For multiples H-sensors, this model will generate multiple Q-model outcomes around each H-sensor. On the other hand, we can observe that a discrete random variable $X: \Omega \rightarrow \mathbb{N}$ is degenerate in $k \in \mathbb{N}$ if $\Pr(X = k) = 1$. Thus, if $f(r) \propto k$, and the domain of $f(r)$ is the circle centered at the sink position with radius r_{ch} , this model will generate outcomes equivalent to M^2P^2 ones. Thus, as M^2P^2 is more general than Q-model in the sense that it generates either homogeneous or heterogeneous topologies, the Q-model can be seen as an instance of the M^2P^2 model with λ appropriately defined. In Section 2.6 we show that the M^2P^2 is suitable for real deployments and we present a guide to a stochastic but planned deployment based on it.

2.4 Topological characterization: the sink betweenness measure

There are many measures proposed in the literature for characterizing and representing complex networks [see, for instance, Luciano et al., 2007].

Consider a network whose topology is represented by the graph $G(V, E)$, where $V = \{v_1, \dots, v_n\}$ is the set of $|V| = n$ nodes, and E is the set of edges.

Depending on the underlying communication model, WSNs can be represented by directed or undirected graphs. Thus, let us define the in- and out-neighborhoods of node v_i as $N_i^{\text{in}} = \{v_j : e_{ji} \in E\}$ and $N_i^{\text{out}} = \{v_j : e_{ij} \in E\}$, respectively. The neighborhood of a vertex v_i is $N_i = N_i^{\text{in}} \cup N_i^{\text{out}}$. The in- and out-degrees of a vertex are defined as $k_i^{\text{in}} = |N_i^{\text{in}}|$ and $k_i^{\text{out}} = |N_i^{\text{out}}|$, respectively. The degree of a vertex v_i is defined as $k_i = |N_i|$. Edges may be weighted, i.e., there may be a function $W : E \rightarrow \mathbb{R}$ which associates a real-valued weight w_e to every $e \in E$.

In the WSN context, we aim at finding strong relationships among topological metrics from the complex network theory and network metrics. For instance, in this work we investigate the relationship between energy depletion and topological metrics that define the centrality of a node in a specific context. Next section presents some centrality concepts, including a new metric firstly introduced in our previous works Oliveira et al. [2010], and Ramos et al. [2012], which is able to describe the energy hole effect.

Distributed inference of topological metrics is always desirable for the design of topology-aware algorithms. In this context, energy-efficient distributed inference is a challenge to be addressed.

2.4.1 Centrality metrics

The classification of the nodes by its structural importance was introduced by Bavelas [1948] and Leavitt [1951]; those ideas amount to as the first centrality index for connected graphs, the *Bavelas's Index*.

Centrality indices [Koschützki et al., 2005; Luciano et al., 2007] aim at estimating the importance of a vertex [Yiwei et al., 2006], that means, to rank it by its topological importance [Wasserman and Faust, 1994]. Vertices positioned

in central areas generally possess higher structural importance than the border ones. Whenever data flows across the network, those central vertices are natural and significant information brokers. Usually, the vertex importance increases with its participation in the paths of a graph [Luciano et al., 2007]. Consequently, the importance of a computational element for a network, or a person to a social network, can be calculated based on its topological centrality.

There are several indices of centrality based on different graph features such as distance between vertices, *Closeness* [Beauchamp, 1965; Sabidussi, 1966], degree, *Eccentricity* [Hage and Harary, 1995], neighbourhood importance, *Eigenvector* [Bonacich, 1972], *Hub Score*, *Authority* [Kleinberg, 1999] and *Page Rank* [Brin, 1998]. Another widely used concept in indices of centrality is the graph shortest path; for example, the *Shortest-path Betweenness Centrality* [Freeman, 1977, 1979] calculates the centrality of vertex i based on the proportion of the number of geodesics (shortest paths) between any pair of vertices that falls on i by the total number of geodesics in the graph.

Locating and counting geodesics is difficult with large networks [Freeman, 1979], and computational resources are limited in WSNs. The most efficient centralized algorithm to calculate Betweenness has running time $O(nm + n^2 \log n)$ for weighted graphs, and $O(nm)$ for unweighted graphs, where n and m are the number of vertices and edges respectively.

The Betweenness of node v is defined as:

$$B(v) = \sum_{s=1}^n \sum_{t=1}^n \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (2.3)$$

where σ_{st} is the number of shortest paths from s to t , $\{s, t\} \in V$, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through $v \in V$, $s \neq v \neq t$ and $s \neq t$.

In WSN scenarios, communication typically takes place between sensor nodes and the sink node, and vice versa. In order to consider this characteristic, we adopt a new centrality metric, namely Sink Betweenness (SBet) [Oliveira et al., 2010; Ramos et al., 2012], which considers only the shortest paths that

include the sink as one of the terminal nodes. It is defined, for every $v \in V$, as

$$\text{SBet}(t) = \sum_{i \in \psi_t} \frac{\sigma_{ts}}{\sigma_{is}}, \quad (2.4)$$

where s is the sink, σ_{ts} is the number of shortest paths from t to the sink, σ_{is} is the number of shortest paths from i to the sink, $\psi_t = \{i \in V \mid t \in SP_{i \rightarrow s}\}$, and $SP_{i \rightarrow s}$ is the set of all shortest-paths from a node i to the sink, so ψ_t is the set of nodes that contains t at least in one of their shortest-paths.

For the sake of simplicity, in this work we consider that WSNs can be represented by non-weighted graphs. In some scenarios, it is more appropriated to use of weighted graphs, and both Betweenness and Sink Betweenness can be easily modified to support such feature.

2.4.2 Evaluation models in the SBet energy analysis

In the following section we present an assessment of the performance of using centrality metrics to capture the energy consumption behavior in a variety of WSNs scenarios. We showed in Ramos et al. [2011a] that other metrics commonly used in complex networks theory fail to represent the energy consumption in WSNs scenarios, thus, in this work we only study the betweenness and the SBet centrality metrics. For this, we estimate the correlation between the spent energy of the nodes and the centrality metrics we are interested in. We used Spearman's rank correlation because it is robust and is recommended if the data does not necessarily come from a bivariate normal distribution [Bonett and Wright, 2000].

We evaluated the performance of these two centrality metrics in a variety of WSNs scenarios, varying (i) the deployment model, (ii) the wireless channel model, (iii) the interference model, and (iv) the routing protocol, when an application of continuous data collection is used, i.e., when all nodes send their data toward the sink continuously.

Regarding the deployment model, we adopted two different ones, the URP model, defined in Section 2.2, and the Q-model, defined in Section 2.3.2. The Q-model was tailored to alleviate the energy hole effect and was shown to be

an instance of the herein defined M^2P^2 model for homogeneous WSNs (see Section 2.3.2).

The wireless channel model describes how the signal propagates in the physical channel. The signal can be diffracted, reflected and scattered. Zuniga and Krishnamachari [2004] argue that these effects lead to two characteristics: (i) the exponential decay of the signal strength, and (ii) the signal strength at distance d follows a log-normal distribution. Thus, the log-normal shadow path-loss model provides accurate channel models. This model describes the path loss at a distance d as:

$$PL(d) = PL(d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma_1}, \quad (2.5)$$

where d is the transmitter-receiver distance, d_0 is a reference distance, n is the path loss exponent, and X_{σ_1} is a zero-mean Gaussian random variable with standard deviation σ_1 , which amounts for the shadowing effects. In this work, we consider only stationary environments, i.e., X_{σ_1} is not a function of time.

The simplest channel considers isotropic communication radii, leading to perfect circular regions. This model, which is termed UDG (Unit Disk Graph), is known to be inaccurate and may not capture real-world conditions. In this work, we start our analysis using this model by doing $\sigma_1 = 0$ in order to better understand the effects of the communication model in the energy hole problem. Empirical studies show that $\sigma_1 = 4$ is able to capture realism introducing shadowing effects [Zuniga and Krishnamachari, 2004]. Observe that when $\sigma_1 > 0$ the communication region may adopt an arbitrary shape.

The log-normal shadowing model as shown in Equation (2.5) is affordable for specifying bidirectional links, i.e., a node u is able to communicate with a node v and vice-versa if they are inside the communication region of each other. Considering the two directions of the link as independent yields a larger variance than in real world experiences, thus, Tselishchev et al. [2010] propose the use the log-normal as shown in Equation (2.5) and then adding and subtracting a zero-mean Gaussian with standard deviation σ_2 . As consequence, if $\sigma_2 > 0$, the links are not necessarily bidirectional. Thus, the greater this standard deviation is, the more likely the links are non-bidirectional. The value of σ_2 should be set

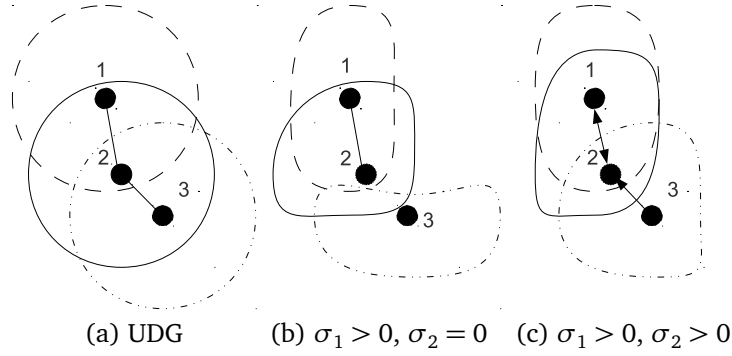


Figure 2.4: Three different wireless channel models

to zero when UDG channels are considered. Tselishchev et al. [2010] empirically devised $\sigma_2 = 1$ as a typical value.

Figure 2.4 depicts the tree situations described above. In Figure 2.4a nodes 1 and 3 can communicate to node 2, once they are inside their communication regions. In Figure 2.4b nodes 2 and 3 are not able to communicate with each other due to shadowing effects, and in Figure 2.4c the link between nodes 2 and 3 is unidirectional as a consequence of $\sigma_2 > 0$.

Another important aspect related to the wireless channel is the interference model. In this work we considered two options. The first is a simplistic ideal channel without collisions, i.e., all packets arrive to the receiver radio interface without errors, even when there are interferences in channel. The second situation is more realistic and considers an additive interference model where transmissions from other nodes are treated as interference by linearly adding their effect at the receiver.

Media Access Control (MAC) protocols play an important role in network communication since they are responsible for transforming a raw transmission facility, namely the physical layer, into a link responsible for node-to-node communication [Forouzan, 2007]. Usual MAC protocols provide a wide variety of well-known features such as framing, addressing, flow control, error control and media access control. In the WSNs context, energy efficiency is a goal that should be pursued in the design of all of these features. In this work, we consider a simple MAC protocol that intends to alleviate collisions in order to trying to keep a

high delivery rate in the sink node. Observe that in high intensive traffic conditions, the energy hole problem might be not well-characterized due to a high packet loss rate even far from the nodes close to the sink. Thus, nodes close to the sink will not receive the lost packets and will not deplete energy relaying them. These saturated environments are usually avoided by the network designer due the network performance degradation and was avoided in our simulations as well. In our assessment, MAC protocol performs carrier sense, exponential backoff scheme and random off-set transmissions in order to alleviate collisions. For the sake of simplicity, we chose a 100% duty-cycle once this factor does not show strong influence on the results.

WSNs are data-driven networks that usually produce a large amount of information that is routed, often in a multi-hop fashion, to a sink node, which works as a gateway to the application. Given this scenario, routing plays an important role in the data gathering process. In this work, we are interest only in providing routing techniques representative of WSNs usual scenarios to show that the metric herein proposed is able to characterize the energy hole problem.

We used two simple routing schemes, namely simple random tree and simple gossiping. The former creates a tree infrastructure by using a flooding starting in the sink node. Thus, each node receiving the setup packet stores its neighbors. Every time a node is ready to transmit a packet, it will randomly choose one of its neighbors that is along the one possible shortest path to the sink.

The second routing scheme is a variation of gossip routing algorithms [Hedetniemi et al., 1988; Niculescu and Nath, 2003], where nodes neither need to setup a parent nor to store their neighbors. The sink node starts a flooding and every node who receives it sets its distance, in hops, to the sink. Thus, every time a node is ready to transmit, it does not send to a specific node but rather broadcasts the packet. Every node that receives this packet and is closer to the sink node than the sender, is able to relay the packet. In order to avoid excessive transmissions and congestion, nodes retransmit packets with a probability proportional to the size of its vicinity. In our simulations, we configured the probability of retransmission to $1/k_i$, where k_i is the node degree¹. For our scenarios, this probability suffices to provide a delivery rate near to 1. Observe

¹Observe that it will be the in-degree for realistic channels

Table 2.1: Simulation scenarios used in the SBet energy analysis

Parameter	Value
sink node	1 (center-most or a random node)
network size	$n \in \{100, 200, 300, 400\}$ nodes
deployment model	URP and Q-Model
simulation time	3000 s
data rate	1 packet/min
sensor field	$100 \times 100 \text{ m}^2$
collision model	no-collision and additive
routing model	random tree and simple gossip
app. model	continuous data collection
σ_1	$\{0, 4\}$
σ_2	$\{0, 1, 2, 3, 4\}$
Sensor model	Mica 2 CC 1000

that the delivery rate may be greater than 1 due to multipath routing.

The application we used represents the usual situation where sensor nodes report information to the base station periodically, but not collectively at the same time. Each node has a parameter called *sampling rate*, all initialized with the same value and denoted by t_{sr} , which indicates the expected time between sensed values to be reported by a sensor node. In order to avoid collisions as much as possible, and for the better sharing of the network resources, each sensor node chooses a random value to transmit its i th value uniformly in the time interval $((i - 1)t_{sr}, it_{sr}]$. Other data collection models such as event-driven and query-based applications [Al-Karaki and Kamal, 2004] are not considered in this work. For these models, the results of the correlation between the centrality metrics and energy consumption are considered only for the subgraph formed by the nodes that are used to collect and relay data. Thus, one can expect similar results herein presented but restricted to the subset of nodes that are used.

2.4.3 Evaluation scenarios used in the SBet analysis

Table 2.1 presents the simulation scenarios we evaluated. A Monte Carlo simulation was performed, replicating independently 30 times each of the 192 scenarios

indexed by the parameters shown in Table 2.1. Observe that we disregarded the situations where $\sigma_1 = 0$ and $\sigma_2 = \{1, 2, 3, 4\}$ because UDG model induces bidirectional links always. This number of replications was considered sufficient for hypothesis testing sample mean differences at the 95% significance level.

The parameters σ_1 and σ_2 specify the wireless channel model as described in Section 2.4.2. When $\sigma_1 = 0$, σ_2 is set to zero, and we have the UDG model. When $\sigma_1 = 4, \sigma_2 = 1$, we have the realistic channel model as defined by Zuniga and Krishnamachari [2004] and Tselishchev et al. [2010].

The radio power is set to provide 15 m of communication radius under the UDG model. This radius induces graphs whose nodes typically have 11 and 49 neighbors, in average, for 100 and 400 nodes, respectively, when URP deployment is used. Thus, we assess from typical to high density WSNs. The communication radius concept is not applicable when the realistic channel is used.

We use the R package version 2.10.1 [R Development Core Team, 2009] for node deployment and statistical analysis, Omnet++ simulator version 3.3p1 for discrete event simulation, and Castalia version 2.3b [Boulis, 2009] for WSN models. Both wireless channel and MAC models were already available in Castalia; the routing and application models were implemented as specified. We also used the Mica 2 CC 1000 radio module available in Castalia.

2.4.4 Results

We studied centrality metrics borrowed from the theory of complex networks: (i) Betweenness [Freeman, 1979], (ii) SBet, (iii) eigenvector centrality [Bonacich, 1987], (iv) closeness [Freeman, 1979], (v) degree centrality [Freeman, 1979], (vi) Google page rank [Brin, 1998], (vii) constraints centrality [Burt, 2004], (viii) hubscore centrality [Kleinberg, 1999], and (iv) authority centrality [Kleinberg, 1999]. We investigated how those metrics are related to the energy spent by a node in all the scenarios described in Table 2.1, which describes a comprehensive set of situations.

Even though we investigate all of these metrics, only Betweenness and SBet presented good results. Those are the metrics with highest Spearman correlation with the energy. In the following, we present the results of our simulations omit-

ting the values of all other metrics, which did not appeared useful in our studies. Thus, we present the results of the correlation between the spent energy and two centrality metrics, namely Betweenness and SBet, when 4 situations are considered: (i) simple gossip routing on URP deployment, (ii) simple random tree routing on URP deployment, (iii) simple gossip routing on Q-Model deployment, and (iv) simple random tree routing on Q-Model deployment. These situations were assessed for eight different scenarios, varying the channel model (UDG and Realistic), the collision model (no-collision and additive collision model) and the sink position (centered and randomly placed). We also vary the probability of the links becoming non-bidirectional ($0 < \sigma_2 < 4$).

In all following situations the transmission is the task that employs most energy. Thus, we are looking forward to a metric able to indicate which node is more likely to transmit more packets.

2.4.4.1 Simple gossip routing and URP deployment

Figure 2.5a shows the correlation of SBet and spent energy when a simple gossip routing and a URP deployment are used. Each curve corresponds to a different value of σ_2 , and the cells correspond to the eight aforementioned scenarios. Notice that only the curve corresponding to $\sigma_2 = 0$ is shown under the UDG model, because it only produces bidirectional links. For UDG model, SBet is highly correlated to the spent energy, and it presents a very robust behavior when varying the number of nodes (i.e. the density), the sink position and collisions. Actually, UDG is an ideal model, which may be unable to capture real world characteristics and, thus, delivery rate is always very high. Even when collisions are considered, the MAC and the application models used in this work make efforts in order to alleviate the loss they inflict, and the correlation is kept almost constant. Note that this behavior is not verified in Figure 2.5b, which shows the correlation between spent energy and Betweenness. In this situation, the sink position is more relevant than when SBet is considered. This occurs because SBet is more robust with respect to the sink position, as betweenness performs better when the sink is at the center. Notice, also, that the number of nodes (i.e., the density, since the area is held constant) has influence on Betweenness even when UDG is considered,

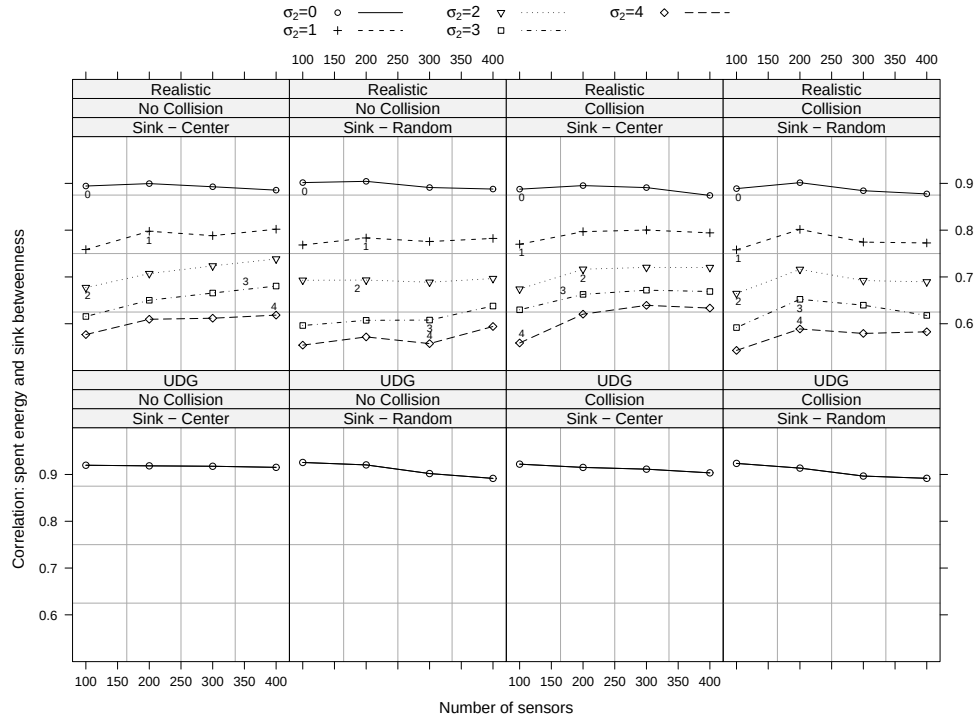
mainly when the sink is randomly placed.

When we consider more realistic channels, SBet exhibits high correlation (consistently greater than 0.75) when σ_2 is zero (bidirectional channels) and 1 (realistic channels). It is clear that an increasing σ_2 degrades the correlation, due to packet loss. Notice that gossip routing herein considered employs routing level information that may not capture the real configuration in the presence of mostly non-bidirectional links, and may lead to additional packet loss and degraded correlation. Betweenness fails to capture which nodes are more likely to waste energy in realistic scenarios. Note also that in all situations of realistic scenarios the correlation is reduced to negligible values, which were discarded in Figure 2.5b to enhance the visualization.

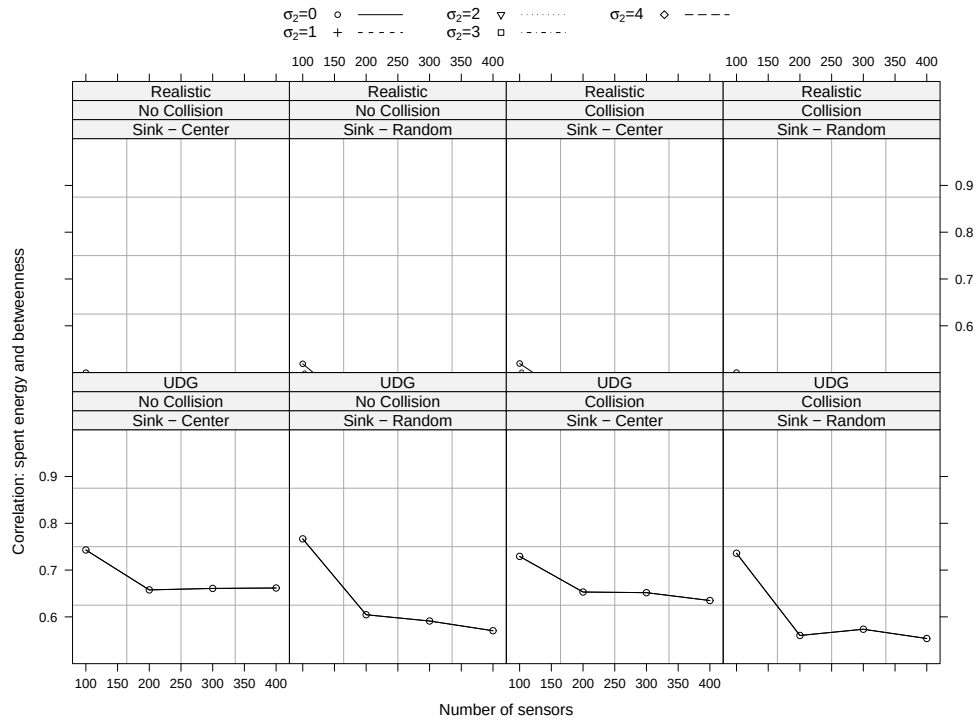
Figure 2.6 shows correlograms of selected scenarios, all with realistic wireless channel models and collisions. The pies depicted above the diagonal illustrate the correlation among the spent energy, Betweenness and Sink Betweenness, while the figures below the diagonal depict the scatterplots between these metrics. We observe that Sink Betweenness is almost independent of the number of nodes and of the sink position in more realistic models. When the number of nodes increases, the MAC loses the ability to manage collisions, and correlation degrades. In extreme situations, the delivery rate becomes quite low and the network metrics tends to degrade.

2.4.4.2 Simple random tree routing and URP deployment

Figures 2.7a and 2.7b show the correlation between the spent energy and SBet and Betweenness, respectively. In these situations we see that the behavior of the correlation does not change substantially. It means that SBet is able to characterize the energy hole problem, and that it is suitable for describing the distribution of the waste of energy related to the nodes' relay task. Betweenness, once again, presents high correlation only in specific situations such as with UDG model and 100 nodes. In all other situations, the correlation is low. For realistic channel models, Betweenness presents poor performance. In case of tree based routing, both SBet and Betweenness present slight higher correlation when compared with gossip based routing.



(a) SBet



(b) Betweenness

Figure 2.5: Correlation between spent energy and measures of centrality, simple gossip routing and URP deployment

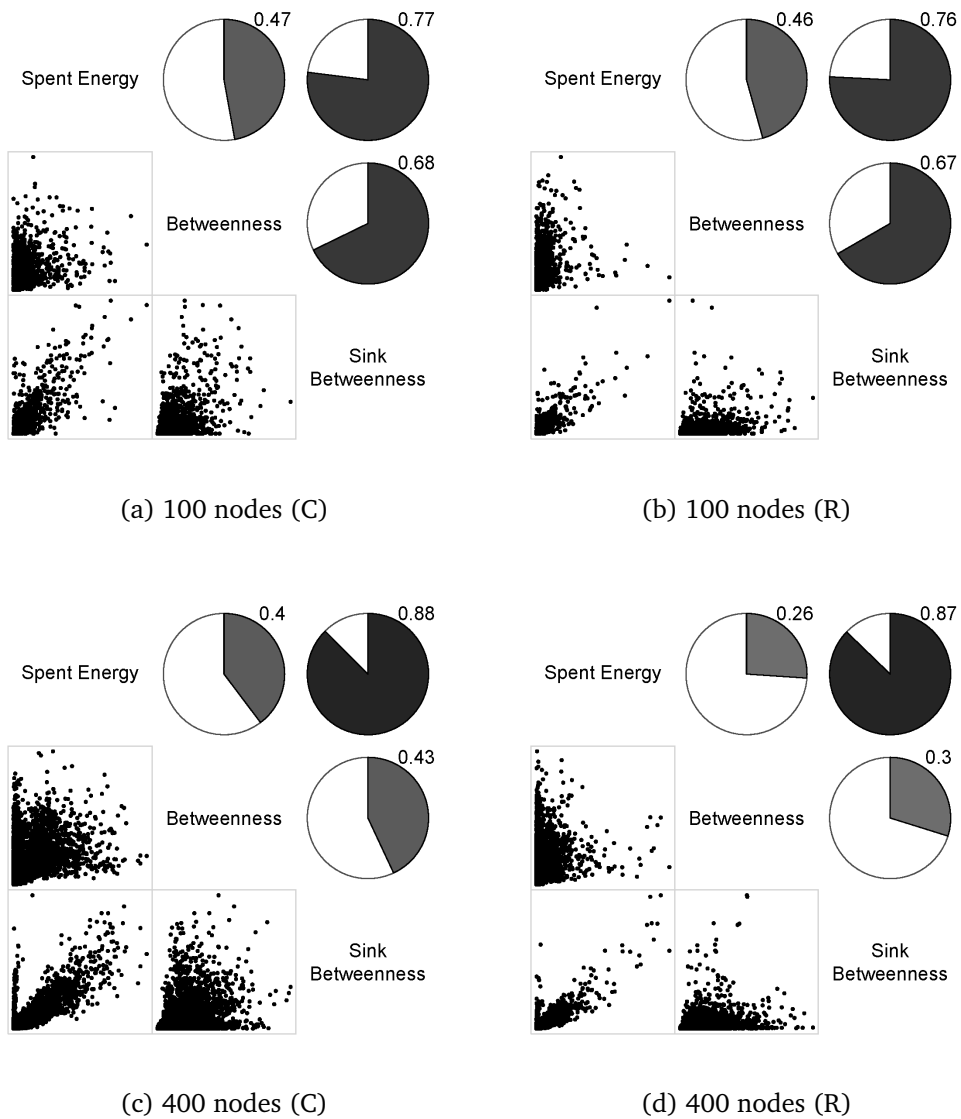
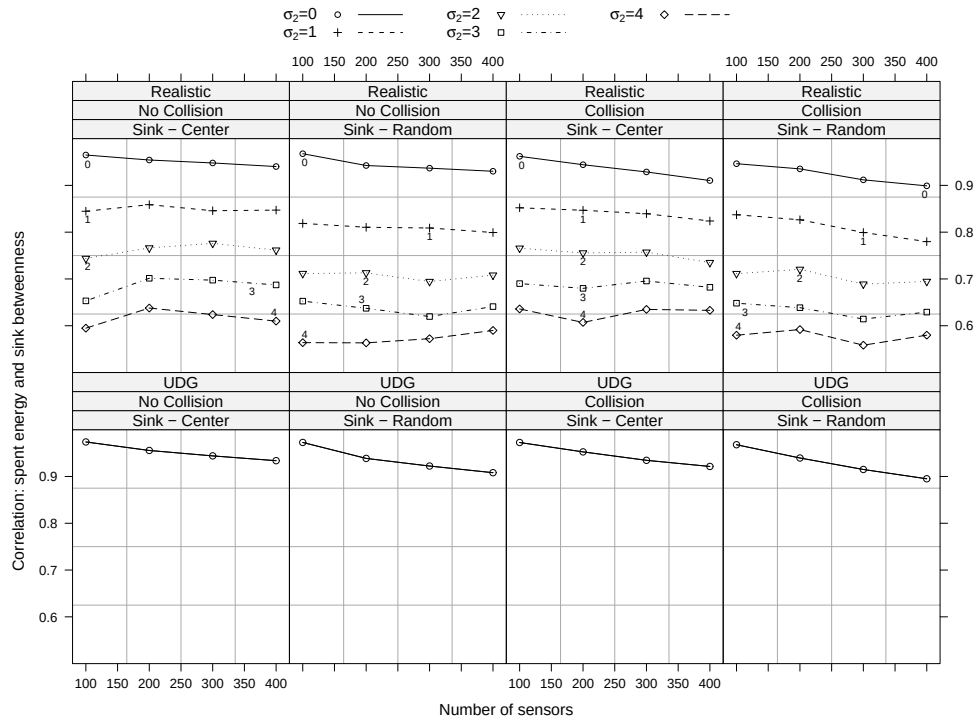
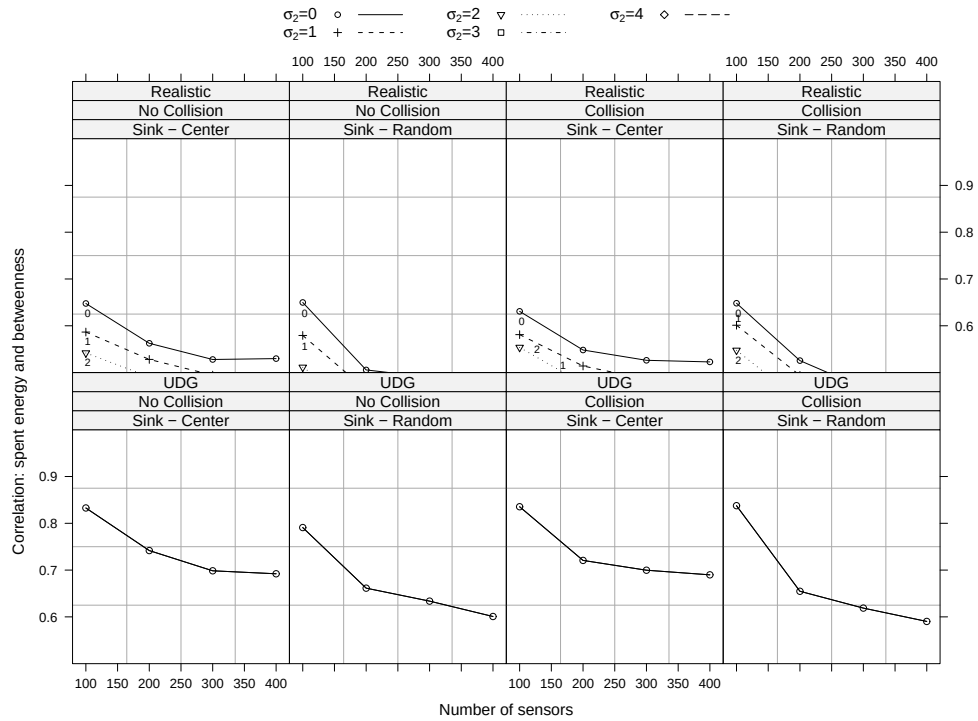


Figure 2.6: Correlograms and scatterplots for gossip routing and URP deployment, 100 and 400 nodes, centered (C) and randomly (R) placed sink



(a) SBet



(b) Betweenness

Figure 2.7: Correlation between spent energy and measures of centrality, random tree routing and URP deployment

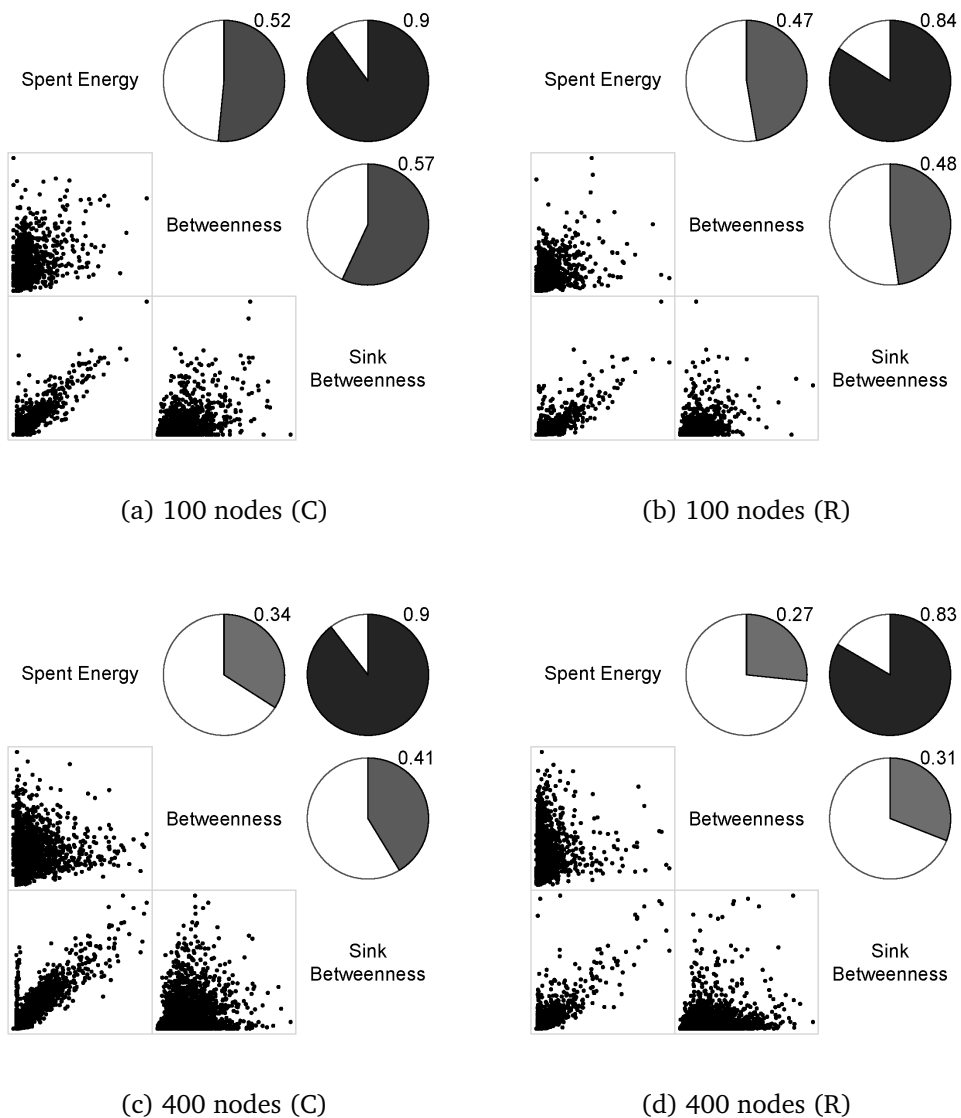


Figure 2.8: Correlograms and scatterplots for tree routing and URP deployment, 100 and 400 nodes, centered (C) and randomly (R) placed sink

Figure 2.8 shows the correlograms for 100 and 400 nodes, with the sink both centered and randomly placed. All correlograms correspond to a realistic channel with collision. Notice that Sink Betweenness presents higher correlation values than Betweenness.

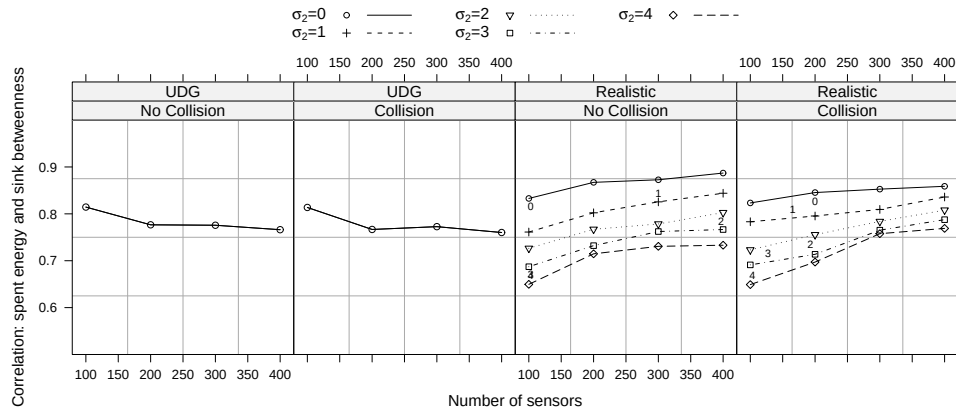
2.4.4.3 Simple gossip routing and Q-Model deployment

Figure 2.9a shows the correlation between spent energy and SBet, when gossip routing and the Q-Model (see Section 2.3.2) are used. We considered UDG and realistic wireless channel with and without collisions. Notice that, for this situation, we did not vary the sink position because the Q-Model assumes that the sink is at the center. SBet presents high values of correlation, in both UDG and Realistic channel models, being slightly higher in the latter without considering collisions. Considering the realistic channel ($\sigma_1 = 4$ and $\sigma_2 = 1$) SBet is highly correlated with the spent energy, and is able to characterize the energy hole problem. When σ_2 increases, the packet loss also increases and correlation degrades. Differently from the URP model, when the Q-Model is used correlation increases when the number of nodes increases. This occurs because the Q-Model tends to concentrate the nodes closer to sink, and the network's diameter is quite small. Thus, most nodes deliver their packets directly to the sink, and the relay task is alleviated. When the number of nodes increases, nodes are more spread on the sensor field increasing, thus, the network's diameter. Figure 2.9b shows that Betweenness presents poor performance in whenever the Q-Model is used, not being able, therefore, to characterize the energy hole problem.

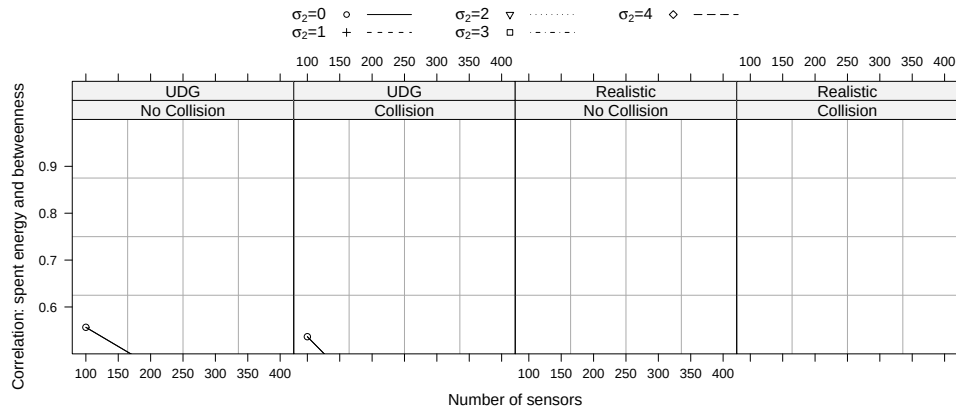
Figure 2.10 shows correlograms between spent energy and measures of centrality with gossip routing and the Q-Model, along with scatterplots. Betweenness and Sink Betweenness considered the realistic channel ($\sigma_1 = 4$ and $\sigma_2 = 1$) and collisions. Observe that Sink Betweenness is still highly correlated, regardless the number of nodes, while Betweenness presents poor performance for all scenarios.

2.4.4.4 Simple random tree routing and Q-Model deployment

Figure 2.11a shows the correlation between the spent energy and the SBet when tree routing and Q-Model are used. We considered UDG and realistic wireless channel with and without collisions. Observe that for Q-Model and tree routing, SBet is highly correlated (typically greater than 0.75) with the spent energy only when the node density is not very high (in our scenarios for 100 and 200 nodes). For the Q-Model, the density is not uniformly distributed over the sensor field;



(a) SBet



(b) Betweenness

Figure 2.9: Correlation between spent energy and measures of centrality when gossip routing and Q-Model deployment

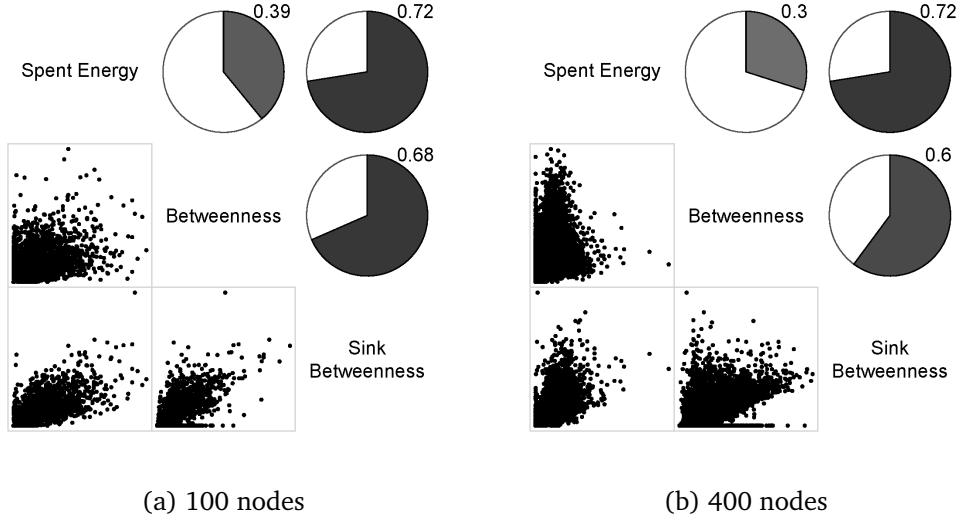


Figure 2.10: Correlograms and scatterplots for gossip routing, Q-Model, 100 and 400 nodes, centered placed sink

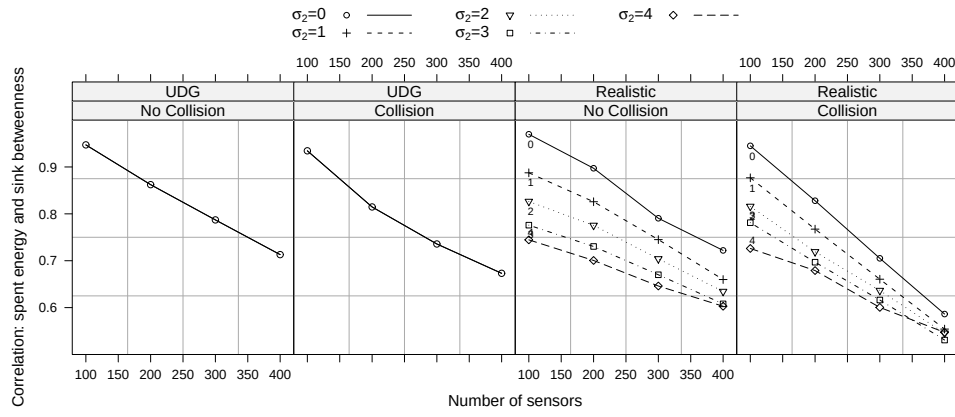
the region close to the sink presents higher density than other regions. In Figure 2.11b, we see that Betweenness presents high correlation (about 0.75) only in one specific situation, namely with 100 nodes and UDG without collisions.

Figure 2.12 shows correlograms between the spent energy and metrics of centrality with gossip routing and the Q-Model, along with scatterplots. Betweenness and Sink Betweenness considered the realistic channel ($\sigma_1 = 4$ and $\sigma_2 = 1$) and collisions. Observe, like all other scenarios, that Sink Betweenness is still highly correlated, while Betweenness presents poor performance for all scenarios.

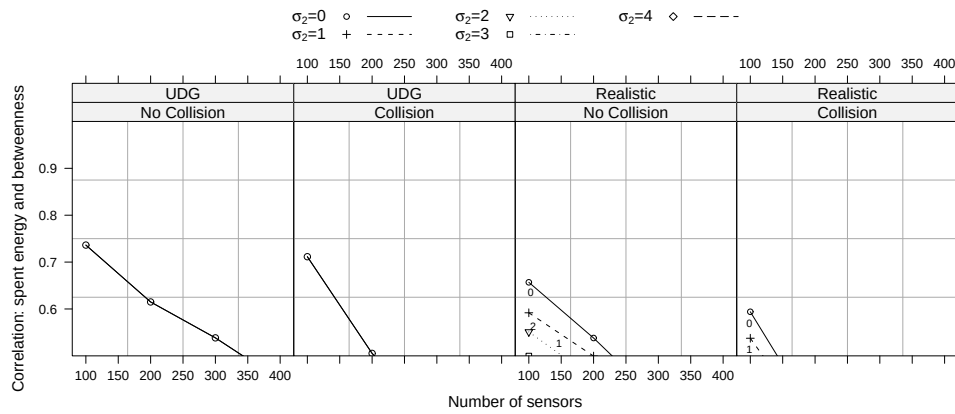
2.5 Evaluation of the M²P² model

The main goal of the M²P² evaluation is to assess properties of the proposed deployment model presented in Section 2.3, namely: (i) coverage and connectivity, (ii) small-world characterization, and (iii) energy hole behavior.

We consider a heterogeneous WSNs with sensors with the same sensing capability (r_s) and two levels of transmission (r_c and r_{ch}) ranges, which are, for



(a) SBT



(b) Betweenness

Figure 2.11: Correlation between spent energy and measures of centrality, tree routing and Q-Model deployment

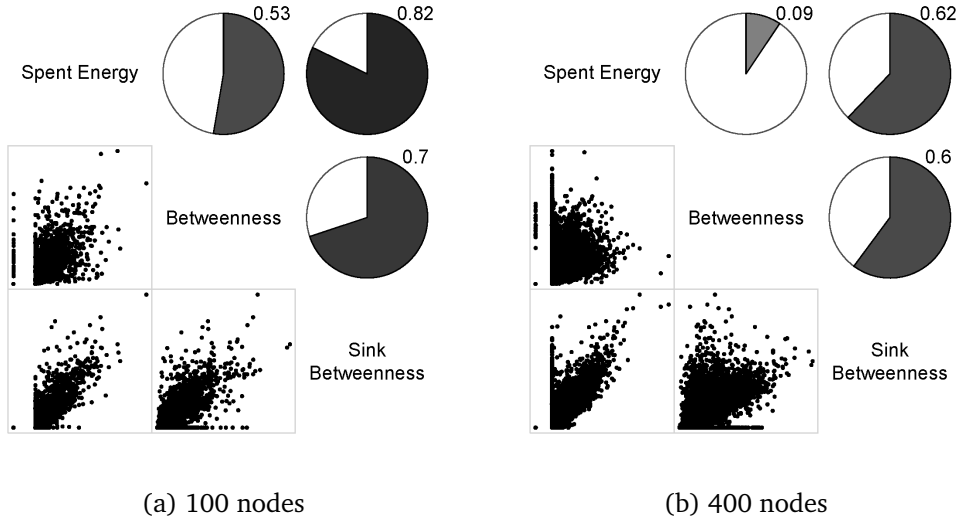


Figure 2.12: Correlograms and scatterplots for tree routing and Q-Model, 100 and 400 nodes, centered placed sink

the sake of simplicity, perfect circles. H-sensors have more capability to transmit data but the same sensing capability of the L-sensors. H-sensors have a radio that operates in two different channels, i.e., they are able to make long-distance communication only with other H-sensors and they use short-distance communication in order to send data to and receive data from L-sensors.

Coverage plays an important role in the WSN design. It indicates how the network covers the sensor field and has a strong influence on the quality of the information reported by the WSN. The coverage of a given topology can be calculated as the union of the areas of the circles centered at each sensor within the sensor field. We only considered nodes inside the connected component the sink belongs to, since only their data will be reported. No data fusion will be considered in the following. Connectivity is the percentage of sensors able to report their information to the sink by using a routing path. Two sensors are neighbors if and only if they are located within each other's transmission range.

Small-world networks share characteristics of both regular and random graphs, presenting high values of clustering coefficient (similar to regular networks) and small values of the average shortest path length (similar to random

networks) [Luciano et al., 2007; Helmy, 2003; Newman, 2003]. The high cluster coefficient makes the topology more fault-tolerant due to the presence of a high density of cycles of order three in the network. Thus, if a node fails, there exist more possible neighbors able to recover from the failure [Helmy, 2003; Guidoni et al., 2008]. Besides, the small values of the average shortest paths indicate that the shortcuts provided by the H-sensors tend to reduce the latency on data communication [Guidoni et al., 2008]. Thus, as stated by Helmy [2003], small-world networks present distinguished characteristics appropriated for WSNs.

The energy hole problem is a very important issue and it can be addressed by a planned deployment [Li and Mohapatra, 2007]. We evaluated the topologies generated by our deployment model by using two different metrics: (i) the SBet metric, as we shown it is high correlated to the energy consumption (see Section 2.4.4), and (ii) the amount of messages transmitted and received by the nodes that are close one hop to the sink node.

To estimate the total amount of transmitted packets we used a routing algorithm where each sensor reports its collected data by using a minimum cost path to the sink, being the cost the number of hops towards the sink. However, to distribute the energy depletion as equally as possible, we constructed a routing tree where the nodes store all neighbors that provide the minimum cost to the sink. Each time the node transmits data to the sink, it randomly chooses as its ancestral one of those neighbor nodes that presents the same cost. We considered an error- and a collision-free MAC protocol to isolate the influence of the aspects of this level on our assessment.

Table 2.2 presents the simulation scenarios we evaluated. Each scenario was replicated 30 times independently. This number of replications was considered sufficient for hypothesis testing sample mean differences at usual (95%) significance levels. Observe that the parameters are quite different from those in Table 2.1 as for heterogeneous networks like ones generated by M²P² model we are interested in large scale networks. Observe that each network represented by the communication subgraph formed by the L-sensors connected to each H-sensor is approximately similar to the homogenous networks characteristics presented in Table 2.1.

We assessed five different values for the parameter a of our proposed model

Table 2.2: Simulation scenarios used in the M^2P^2 model analysis

Parameter	Value
sink node	1 (center-most node)
network size	{1000, 1500, 2000} nodes
L-sensors' comm. radius	50 m
H-sensors' comm. radius	{100, 300, 500} m
number of H-sensors	{10, 30, 50} nodes
deployment model parameter (a)	{1, 5, 15, 30}
event duration	1000 s
data rate	1 packet/min
sensing radius	30 m
sensor field	$1000 \times 1000 \text{ m}^2$

as shown in Table 2.2. These parameters are:

1. **independent | independent**: represents the binomial deployment for both L-sensors and H-sensors, also called as totally independent deployment;
2. **independent | repulsive ($a = 1$)**: represents the binomial deployment for L-sensors and repulsive deployment for H-sensors;
3. **slightly attractive | repulsive ($a = 5$)**: represents the slightly attractive deployment for L-sensors and repulsive deployment for H-sensors;
4. **fairly attractive | repulsive ($a = 15$)**: represents the fairly attractive deployment for L-sensors and repulsive deployment for H-sensors;
5. **strongly attractive | repulsive ($a = 30$)**: represents the strongly attractive deployment for L-sensors and repulsive deployment for H-sensors.

Except for the totally independent deployment, H-sensors are placed by repulsive deployment with inhibition radius between r_c and $\ell/m^{1/2}$, where r_c is the L-sensor communication radius. More details about the inhibition radius will be provided in Section 2.6. The sink node is the center-most H-sensor of the sensor field.

The evaluation is based on a Monte Carlo experiment to quantify the influence of the proposed deployment model on coverage, connectivity, small-world characterization, and the energy hole behavior. We use the R package version 2.9 [R Development Core Team, 2009] for node deployment and topological quantities estimation, and Sinalgo simulator version v.0.75.3 [Sinalgo, 2008] for discrete event simulation.

The following sections discuss the simulation results. All figures of the next section are comprised of nine plots, presented in a 3×3 organization. The rows depict the variation of the communication radius of the H-sensors $r_{ch} \in \{100, 300, 500\}$ m) and the columns depict the variation of the total number of nodes ($n \in \{1000, 1500, 2000\}$ nodes). The abscissas represent the number of H-sensors ($m \in \{10, 30, 50\}$ nodes), and the ordinates are the variables under study. The curves represent the average values while the error bars represent the confidence intervals for 95% of significance.

2.5.1 Coverage and connectivity

Figures 2.13a and 2.13b show the results of the coverage and connectivity. We observe that for homogeneous networks, both coverage and connectivity are close to 100%, regardless of the total amount of nodes. It means that for a binomial process, 1000 sensors suffice to cover the field. But the attractiveness of the proposed model exerts influence on those metrics. For instance, we observe that regardless of the number of nodes, the coverage of the fairly and the strongly attractive deployment is less than the one observed in the independent distributions. On the other hand, slightly attractive topologies reach similar values when an appropriate number of nodes is deployed. This follows from the fact that the attractiveness increases the redundancy on areas close to the H-sensors and sink. Thus, other regions of the network are not covered.

We also observe that for 1500 nodes, being 30 of them H-sensors, the slightly attractive model presents results close to the independent deployment in terms of coverage and connectivity. Observe that, sometimes, for strongly and fairly attractive deployments, the increase in the number of H-sensors diminishes the coverage and connectivity until a certain quantity of H-sensors and

then, they start to increase. For instance, we observe this behavior with 1000 nodes and $r_{ch} = 100$. Between 10 and 30 H-sensors, coverage and connectivity tend to diminish and, after that, they start to increase. This follows from the unusual undesirable situation where the H-sensors subgraph is not connected. When the number of H-sensors increases, this subgraph tends to be connected and the behavior tends to regularize.

The general behavior is that we need more sensors to ensure the quality of service regarding the coverage and connectivity when the deployment is more attractive. These results are useful as a guide to how to choose the number of each type of sensors and the deployment method.

2.5.2 Small world characterization

Figure 2.14 shows the cluster coefficient and the average path length for our scenarios. We observe in Figure 2.14a that the clustering coefficient is noticeably higher for attractive deployments, regardless the total amount of H-sensors in network. It occurs due to the fact that attractiveness increases the probability of cycles of order three in the network. When the total amount of nodes increases, the cluster coefficient increases as well.

In Figure 2.14b the average path length behavior is presented, showing the influence of the presence of the shortcuts (links among H-sensors) on the shortest paths observed on the topologies herein assessed. In this plot we show the values when there is no H-sensors, i.e., a homogeneous network, to show that the average path length is high. In the presence of H-sensors, the average path length starts to decrease, specially when the r_{ch} increases. For some cases, as when $r_{ch} = 100$, specially when the attractive scenarios are used, we see a non-smooth behavior due to connectivity problems. When r_{ch} is set to 300 or higher, the average path length decreases monotonically for all deployment models when the number of H-sensors increases.

For the evaluated scenario, 30 H-sensors with $r_{ch} = 300$ suffice. We also observe that attractive topologies tend to diminish the average path length. Thus, attractive deployment with repulsive H-sensors lead to a network that tends to present the small-world behavior because it increases the cluster coefficient and

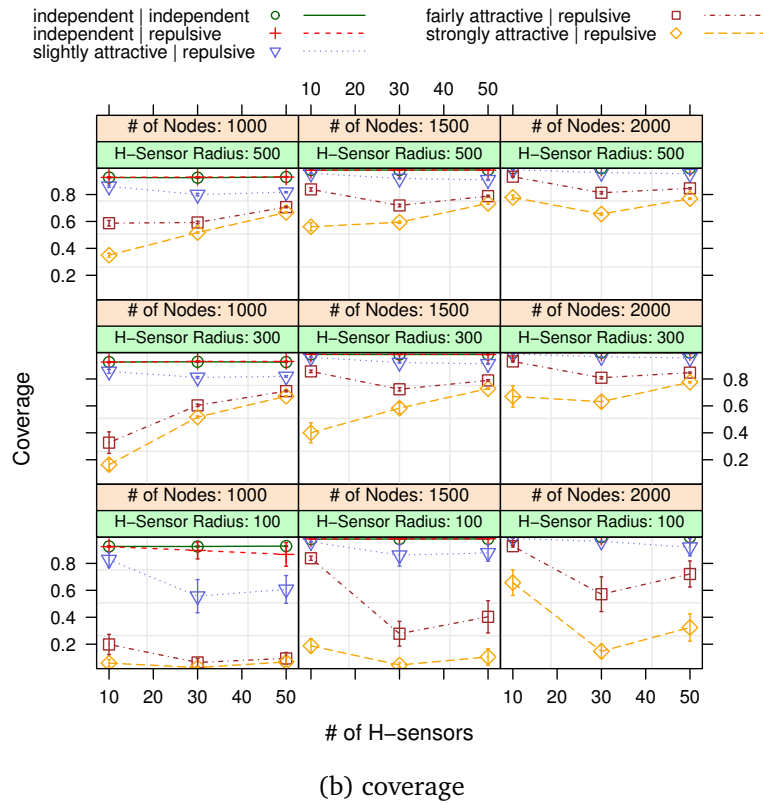
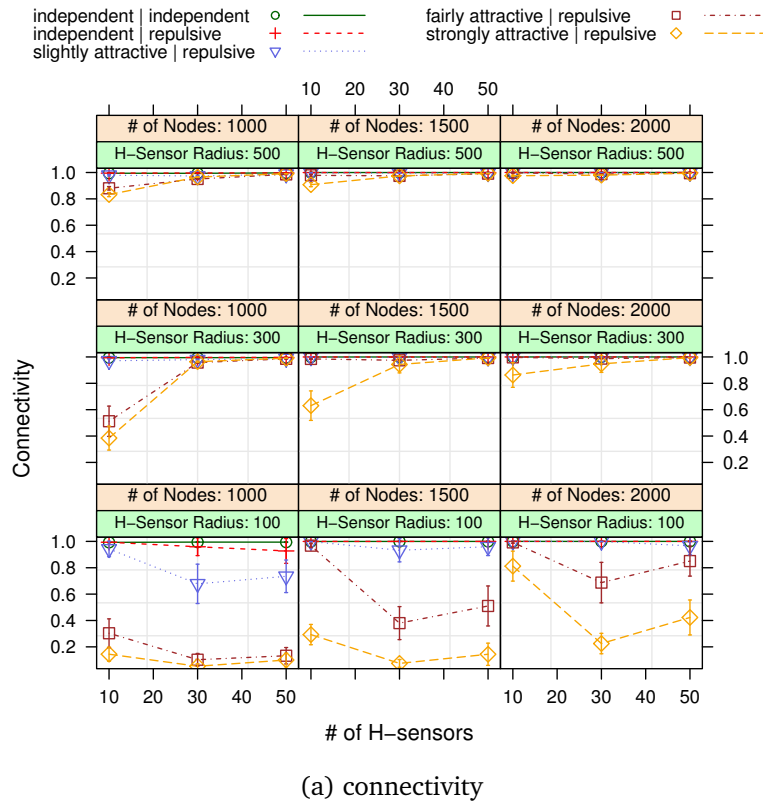


Figure 2.13: Coverage and connectivity as a function of the number of H-sensors

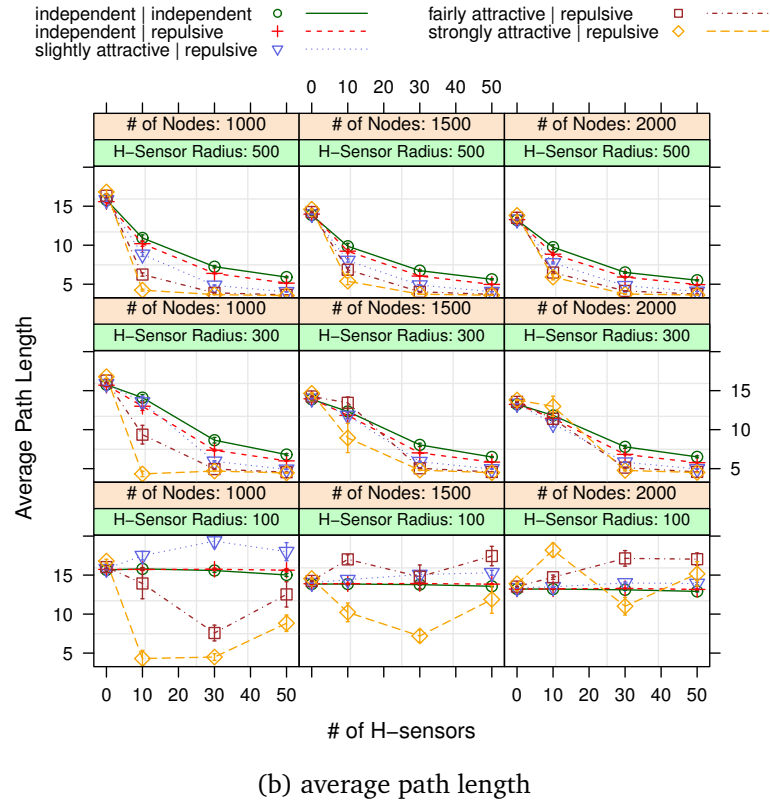
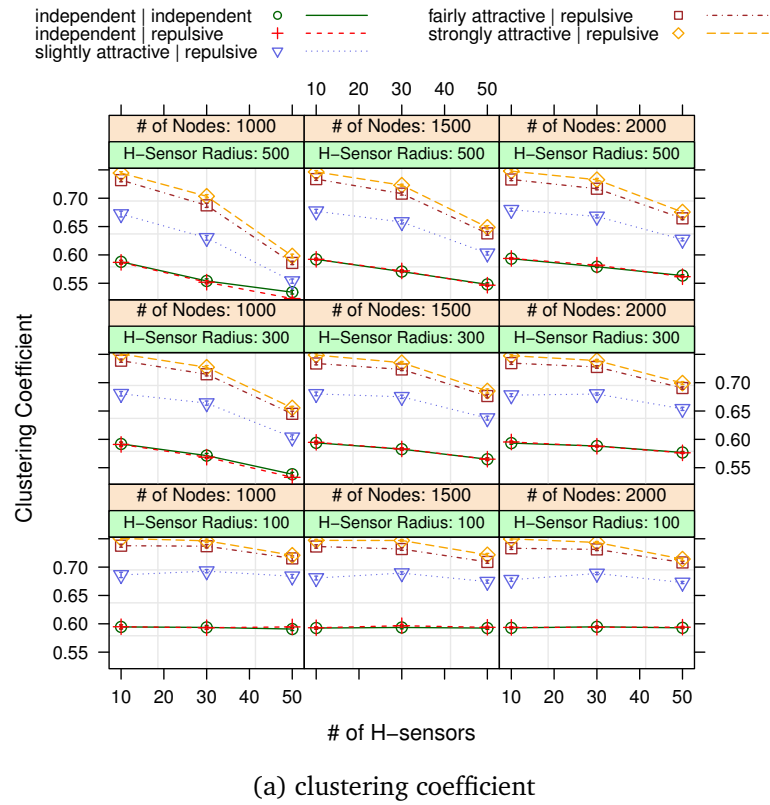


Figure 2.14: Clustering coefficient and the average path length

Table 2.3: Small world characterization of the M²P² model

Topology	$\overline{CC}$	$\widehat{\sigma}_{CC}^\dagger$	$\overline{L}$	$\widehat{\sigma}_L$
slightly attractive repulsive	0.658	9.546	6.313	0.553
independent independent	0.584	5.509	8.205	0.901
homogeneous network	0.595	6.671	13.878	0.194
Erdős-Rényi random graph	0.011	0.420	2.848	0.006

[†] for display purposes, these values are multiplied by 10^3 .

decreases the average path length.

For the sake of comparison, Table 2.3 shows mean estimates of the cluster coefficient CC and average path lengths L , and their respective sample standard deviation $\widehat{\sigma}_\bullet$ for four different topologies, namely, slightly attractive of L-sensors with repulsive H-sensors, totally independent placing of L- and H-sensors, homogeneous, and the Erdős-Rényi [Erdős and Rényi, 1959] random graph. These values were estimated with 1500 nodes and 30 replications. In the first two topologies, there are 30 H-sensors and $r_{ch} = 300$. For the generation of the Erdős-Rényi graphs, we used the same number of vertices, and we chose the probability for drawing an edge between two arbitrary vertices as 0.1, which generated outcomes with similar number of edges as models shown in Table 2.3.

We observe that the cluster coefficient of our proposal is the highest. Note also that the slightly attractive topology is more likely to create a small-world network than the others are. With the addition of 2% of H-sensors, we obtained an average path length about 2.2 times smaller than the solutions without H-sensors (homogeneous network). The solution with a non-planned deployment (independent | independent) leads to a 1.3 times greater average path length than the planned one. The Erdős-Rényi random graph is presented here as a lower bound to the average path length. It is clear that it is unfeasible to create WSNs with similar average path lengths due to geographical conditions.

2.5.3 Energy balancing

The lifetime of an individual sensor is the total time it can receive and transmit data until it runs out of energy. The entire network lifetime is a more sophisticated subject and it is out-of-scope of this work (we refer to Anastasi et al. [2009] for further details). The transmission and reception of data dominate the energy consumption of each sensor and, thus, we disregard any other energy spent in other tasks as sensing, processing and when the sensor is idle [Somasundara et al., 2006; Wang et al., 2008]. The lifetime (and the energy hole problem) is closely related with the total amount of messages transmitted and received by the nodes situated either one hop away from the sink or H-sensors. This account also indicates how fair-distributed is the relay task for each topology we assessed.

We estimate the SBet and the number of transmissions of the nodes one hop distant from the sink and the H-sensors. The results are presented in Figure 2.15a and 2.15b, respectively.

We observe in Figure 2.15a that attractive topologies tend to decrease the SBet and corroborates our hypothesis that planned deployments that use attractiveness around the H-sensors and the sink can address the energy hole problem. Slightly attractive topologies present SBet values close to one order of magnitude lower than the independent topologies while more attractive ones present SBet more than one order of magnitude lower. Observe that systematically the attractive topologies lead to a quite lower value of the SBet. Moreover, Figure 2.15b confirms the fact that the energy hole problem can be addressed with attractive topologies once it shows that the general behavior of the number of transmitted messages is quite similar to the behavior of the SBet metric. Note that an attractive deployment process leads to lower values of transmitted packets. The behavior of both metrics are similar and also corroborates with the fact that the SBet is high correlated with the energy consumption due to the relay task.

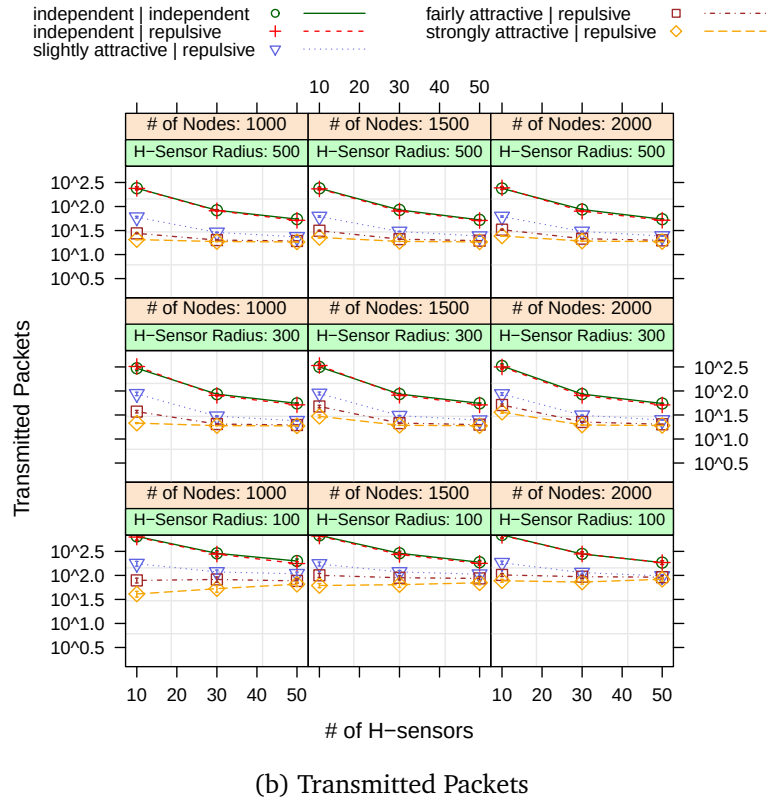
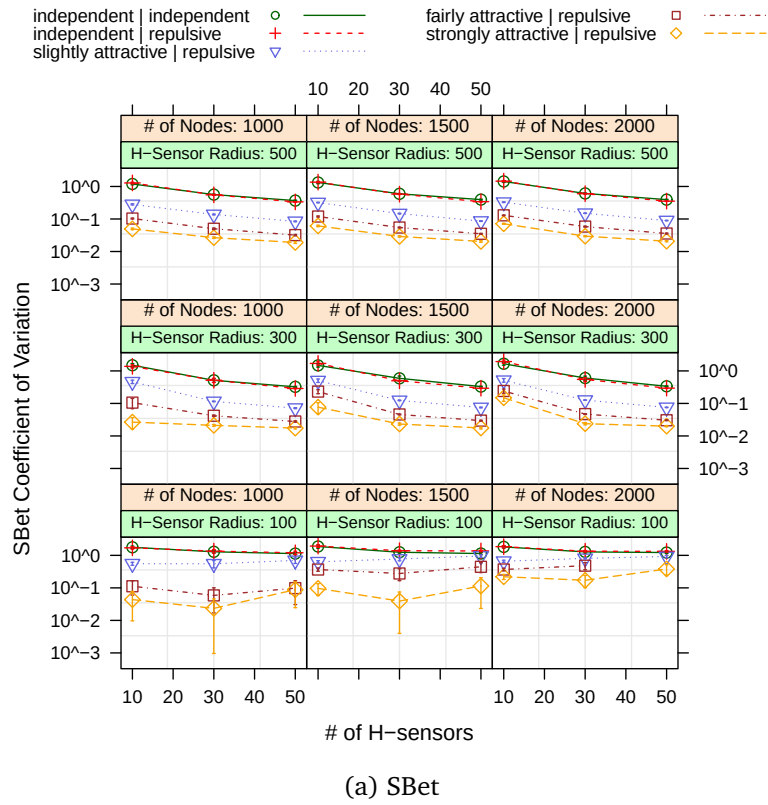


Figure 2.15: Energy consumption metrics as function of the number of H-sensors

2.6 A guide to a stochastic planned deployment

This section discusses the parameters for the M^2P^2 process that were used in our simulation, and how such parameters should be chosen for the description of real world situations.

The $M^2P^2(m, n, a, r_c, r_{ch}, r_i)$ process on W has the following parameters:

- The window W where the process takes place; the user should describe the actual geometry of interest.
- The distance measure among sensors, c.f. Equation (2.1). For the sake of simplicity, we used the UDG (Unit Disk Graph). The communication radii should be carefully specified as a function of the communication channel. This distance specifies r_c and r_{ch} .
- Number (n) and type of sensors required for precise, lasting and economic data acquisition and delivery; the m H-sensors, which are the most expensive type of sensor, aim at improving network performance, while $n - m$ L-sensors are primarily devoted to data collection and first-level data relay. Nordio et al. [2007] discuss the minimum number of homogeneous sensors required to grant a certain data fidelity.
- The inhibition parameter r_i that specifies the minimum distance at which H-sensors are allowed to lie. Overlapping H-sensors are redundant and waste resources, so the repulsive placement we proposed enhances network performance in an economic manner.
- The intensity parameter $a > 1$, which describes the expected proportion of L-sensors around an H-sensor; outside this area, the expected number of sensors is proportional to $1/a$.

As previously mentioned, the inhibition radius r_i grants that the areas of influence of H-sensors do not overlap and, at the same time, it allows the placement of all the m H-sensors on the window $W = [0, \ell]^2$. The first condition requires

$r_i \geq r_c$, while the last condition imposes $2r_i < \ell/m^{1/2}$. The most repulsive process, namely the one with $2r_i \approx \ell/m^{1/2}$, yields H-sensors deployed in an almost regular grid.

The intensity parameter $a > 1$ specifies the expected number of L-sensors around each H-sensor (assuming non-overlapping areas of influence); it can be chosen, among other criteria, using our simulation results.

Given the parameters that specify the process, i.e., m, n, a, r_c, r_i , the deployment can begin. Assume that a two-stage procedure will be done with the aid of a helicopter. The first flight will drop the m H-sensors, at distances not smaller than $2r_i$; these sensors are contained in m bags carrying Z (a random variable assuming integer values) L-sensors that will spread not further than r_c (communication radius of the L-sensors) around each H-sensor. $E(Z)$ is the expected number of L-sensors around each H-sensor. The second flight will spread the remaining sensors in a binomial fashion avoiding the areas already covered by the sensors delivered in the first flight.

We now derive an expression for $E(Z)$. For this, consider the stochastic process that uses the rejection method [Robert and Casella, 2000] to place $n - m$ L-sensors in W into two regions with different probabilities. The region W' is the union of all regions of influence of the m H-sensors. The probability of one point to be placed in the area W is

$$\frac{1}{a} \left(1 - \frac{\mu(W')}{\mu(W)} \right) + \frac{\mu(W')}{\mu(W)}. \quad (2.6)$$

The first term represents the probability of a point to be located in $W \setminus W'$ with intensity a^{-1} , while the second term is the probability of a point to be located in W' with intensity 1. Note that if one point lies in the region W' it is accepted with probability 1, and if it lies in the region $W \setminus W'$ it is accepted with probability a^{-1} . Thus, p points are expected to place $n - m$ points in the region W . Each point will appear with probability that follows the Equation (2.6). For this process,

$$p \left[a^{-1} \left(1 - \frac{\mu(W')}{\mu(W)} \right) + \frac{\mu(W')}{\mu(W)} \right] = n - m. \quad (2.7)$$

A fraction $p(\mu(W')/\mu(W))$ is expected to appear in W' . Therefore,

$$E(Z) = \frac{p}{m} \left(\frac{\mu(W')}{\mu(W)} \right) \quad (2.8)$$

L-sensors are expected around m H-sensors.

Thus, using Equations (2.7) and (2.8) we derive

$$E(Z) = \frac{n - m}{m \left(\frac{1}{a} \left(\frac{\mu(W)}{\mu(W')} - 1 \right) + 1 \right)}. \quad (2.9)$$

2.7 Chapter remarks

We proposed a novel modeling solution that is able to represent a wide variety of scenarios, from totally random to planned stochastic node deployment in heterogeneous sensor networks. This model can represent WSNs showing characteristics of small-world networks and can address the energy hole problem.

We showed that by using only about 3% of H-sensors (50 of 1500) and deploying nodes by using the slightly attractive L-sensors around the repulsive H-sensors model we observe important characteristics of the network topology such as low average path length, and high clustering coefficient. Moreover, we propose the SBet, a metric suitable to characterize the relay task of a node.

This measure's ability, in contrast to the relative insensitivity of the classical Betweenness, suggests other possibilities. SBet can be used in wide variety of applications, both in the design and operation of WSNs. For instance, the designer can assess the best deployment strategy in order to create graphs with more appropriate SBet distribution. Such assessment should improve the understanding and management of network lifetime, since the energy consumption becomes more evenly distributed among the nodes. Studies in that direction require only spatial point processes generators (in order to model the deployment models), and tools for graph analysis, i.e., there is no need of either complex discrete event simulators or network models. Both are provided by R, a free, multiplatform software environment for statistical computing and graphics which exhibits excellent numerical properties [c.f. Almiron et al., 2009].

This work suggests other possibilities. For instance, we envision the following research lines: the quantification of the relationship between the metrics herein used like the cluster coefficient, the average path length and the SBet with the fault-tolerance properties, the latency and the network lifetime, respectively; the introduction of fault-tolerance schemes based on the proposed model and metric; the use of topology control schemes, based on the SBet, to diminish the possibility of interference on nodes that were attractively deployed around the H-sensors and the sink, and the use of SBet to improve the routing performance in WSNs.

Topology-aware design of wireless sensor networks

“By three methods we may learn wisdom: First, by reflection, which is noblest; second, by imitation, which is easiest; and third by experience, which is the bitterest”

Confucius

The use of topological features, more specifically, the importance of an element related to its structural position is a subject widely studied in the literature. For instance, the theory of complex networks provides centrality measures that have been applied to a large variety of fields (e.g., social sciences and biology). In this chapter, we propose the use of a topological measure, the Sink Betweenness (SBet), introduced in Chapter 2, in the design of Wireless Sensor Networks (WSN). We use a distributed algorithm to calculate it, proposed in Oliveira et al. [2010], and show its applicability on energy balancing problems, more specifically in a problem called *energy hole*. In this problem, nodes closer to the sink are more likely to relay a larger number of packets than those that are further. This phenomenon is strongly related to the topology induced by the deployment

of nodes along the sensor field, and, thus, to deplete their energy budget. It can be effectively captured by the SBet metric. Thus, we devise a data-collection algorithm that is able to distribute the relay task more evenly. Simulation results show that the SBet metric can be satisfactorily used in this scenario. We compare the proposed approach with some data collection algorithms and show that the proposed algorithm allows to alleviate the energy hole effects by evenly balancing the relay load, and thus, increasing the network lifetime. This applications illustrate how the topology awareness can be used to improve a network function in a WSN.

3.1 Introduction

Wireless sensor networks suffer from constraints that are not common in other networks [Akyildiz et al., 2002] such as: (i) the number of nodes tends to be orders of magnitude greater than, for instance, in ad hoc wireless networks, (ii) the deployment is usually dense, (iii) the node failure is a common fact, and (iv) nodes present severe energy, processing and memory constraints. These characteristics and constraints make the design of a WSN a challenging task. Furthermore, several techniques employed in other networks are not suitable for WSNs, and new solutions must be devised [Perillo and Heinzelman, 2009].

Communication is an essential task of a WSN and is usually classified as: (i) data collection, and (ii) data dissemination [Al-Karaki and Kamal, 2004; Machado et al., 2005]. In the former case, the information flow occurs from nodes to the sink to report an event, for instance, whereas the latter is performed from the sink to nodes, for instance, to configure the network. In this work, we are only interested in data-collection communication.

Network topology plays an important role in the design of wireless sensor networks. Many important properties such as coverage, connectivity, data fidelity, and lifetime are directed influenced by the way nodes are placed in the sensor field and the way they communicate among them. Thus, some constraints presented by a WSN can be the result of the network topology.

In this work we propose the use of a centrality metric, called Sink Betweenness (SBet) (See Chapter 2), which stems from the theory of complex net-

works [Newman, 2003] but is adapted to WSNs and captures relevant information for this kind of network, to devise a data collection algorithm in scenarios where energy consumption is the most important parameter to address.

Data-collection schemes are usually devised to meet the application requirements. In practice, there is a large variety of WSN applications, and it is very important to identify models which are able to capture, from the application point of view, the essence of a group of communication patterns. In general, data-collection applications for WSNs can be classified as: (i) event-driven, (ii) continuous data, and (iii) query-based applications [Al-Karaki and Kamal, 2004].

In event-driven applications, sensor nodes may typically remain idle until the occurrence of an event of interest, when they react and build a communication infrastructure to deliver the data. This is known as a reactive data collection strategy, since the infrastructure is built only when needed. An alternative is to proactively build and maintain the infrastructure. In this work, we only consider the former scenario, i.e., reactive data collection schemes. Nakamura et al. [2006] show how a reactive scheme can save energy during idle periods in an event-driven WSN. In this kind of application, only nodes that detect an event report their data toward the sink. Other nodes may be part of the data collection infrastructure in order to relay data in a multi-hop fashion.

The main goal of data-collection algorithms is to provide a communication infrastructure to deliver the sensed data through paths that minimize the number of retransmissions and potentially decrease the energy consumption. Moreover, the network may use the processing capacity of its nodes to perform some additional in-network operations. A common task is data fusion [Nakamura et al., 2007] that aims at taking advantage of the data redundancy, increasing data accuracy, reducing the communication load, and saving energy. Krishnamachari et al. [2002] show that a Steiner tree comprised of nodes that detected the events, the sink, and additional nodes creates the optimal infrastructure data-collection infrastructure (Steiner nodes) for this application. Our proposed metric, SBet, was also applied to this kind of scenario, in order to build data collection infrastructures that favor the data fusion for event-driven WSNs [Oliveira et al., 2010].

In the continuous data scenario, it is expected that all nodes have the same

number or sensed packets to send to the sink at a given time interval, but due to the multi-hop communication, the closer to the sink, the larger the number of relay tasks the node will perform. As consequence, Wu et al. [2008] show that the lifetime of a uniformly deployed WSN is impaired by the sensors at the first hop from the sink, a problem known as *energy hole*. The two most usual approaches to mitigate the energy-hole effect are the increase in the node density nearby the sink, and the use of a mobile sink. In this Chapter, we propose a different approach that uses the SBet in order to balance the relay task and mitigate the energy hole problem.

In query-based scenarios, sensors capture data from the desired observed phenomenon and do not send them to the sink immediately. Instead, they store the data and possibly perform a local processing. The application uses the WSN as a distributed database, and triggers data collection throughout queries. Usually, the entity responsible for triggering the queries is the sink node, and the data-collection routing infrastructure is built on demand. This kind of application is not considered in the evaluation of our proposed algorithms, however, from the data-collection algorithms point of view the infrastructure presents similarities with the event-driven applications and algorithms that use the SBet metric, like those presented in Oliveira et al. [2010] and Ramos et al. [2012], can be easily adapted for such scenarios.

The rest of this Chapter is organized as follows: Section 3.2 presents a review of the literature related to the subjects studied in this work. Sections 3.5 present the evaluation of the SBet in a data-collection applications where the energy balance is the most important task. Finally, Section 3.6 discusses the final remarks and future work.

3.2 Related work

3.2.1 Topology-related algorithms

Celebi and Arslan [2007] propose a location-aware engine architecture for cognitive wireless radio and networks such that location information can be used to optimize the network performance. Based on that work, Hoydis et al. [2009]

suggest that topological neighborhood information can be used to improve performance. They also state that further work is still needed, i.e., there are open research venues involving the relationships between topology and network characteristics. Their studies only consider the relationship between throughput and topology.

Perillo and Heinzelman [2009] propose coverage-aware routing protocols to deliver messages that avoid sparsely deployed regions in such a way that coverage remains high for a long period. A careful study of that work reveals that routing metrics based on coverage properties are, actually, based on topology metrics (node density).

Xi and Liu [2009] propose a geographic routing algorithm that uses local (based only on neighbors) and non-local (based on a set of other nodes) topological information. They employ the concept of a small-world network to choose those non-neighbor nodes that will provide the topological information. They show that the proposed routing protocol enhances the packet delivery ratio, without the need of void-handling techniques; it yields nearly shortest forwarding paths, especially if the network is seriously obstructed by obstacles.

Zhang et al. [2007] propose an approach to alleviate network congestion by controlling the topology. They present a model to estimate the capacity of various topologies, and use this estimate to determine the nodes' duty-cycle. They show that this approach increases the data delivery rate, saves energy, and effectively meets real-time requirements.

Dolev et al. [2010] propose a generalization of the well-known Betweenness centrality metric [Freeman, 1977], called Routing Betweenness Centrality (RBC), which accommodates arbitrary loop-free routing schemes. Other variations of Betweenness have been proposed such as the Shortest Path Betweenness Centrality (SPBC) [Freeman, 1977] and the Load Centrality (LC) [Goh et al., 2001], but most of them rely on shortest paths. Dolev et al. state that this assumption is not always adequate, especially for data communication protocols, which use non-shortest paths to deliver some kind of traffic. In WSNs scenarios, shortest paths are more likely to be used since energy consumption aspects are vital. The authors also present distributed algorithms for calculating the RBC metric in a large variety of scenarios.

The aforementioned proposals suggest that more detailed and thorough studies involving the relationship between topology and network characteristics should advance the knowledge and practice of wireless communication, especially wireless sensor networks. In this direction, Hoydis et al. [2009] indicate that this subject is an emerging research area, Haenggi et al. [2009] present theoretical network limits imposed by the network topology, and Perillo and Heinzelman [2009] propose a distributed algorithm that uses topological information to improve network performance.

In the following, we present a review of the literature regarding techniques to address the energy hole problem and load balancing in WSNs.

3.2.2 Energy hole

Wu et al. [2008] show that the lifetime of a uniformly deployed WSN is limited by the sensors at the first hop from the sink, a problem known as energy hole. Mohapatra [2005]; Li and Mohapatra [2007] present the first mathematical model toward the characterization of the energy hole problem. They consider sensor nodes distributed according to the Uniform Random Placement (URP) law in a circular region divided in concentric coronas. They observe the impact of the following four factors on the energy hole problem: node density, hierarchical deployment, source bit rate and traffic compression. They show that simply adding more nodes to the network does not solve the problem.

Liu et al. [2007] propose a different approach to the energy hole problem; they consider a nonuniform node deployment. They derive a placement function based on the distance (in hops) to the sink. An extension of this idea is presented by Wu et al. [2008] who show that nearly balanced energy depletion is possible by increasing the density in geometric progression from the outer to the inner coronas. Thus, they propose a nonuniform node distribution strategy: the Q-Model.

Song et al. [2008] strive to mitigate the energy hole problem by performing power control that adjusts the transmission range to increase the node density surrounding the sink node. Liu et al. [2010] deal with a similar problem, but they consider hierarchical networks where cluster heads perform data aggregation. As

cluster heads also suffer from the energy hole problem, they use different cluster sizes to mitigate this problem.

Kim et al. [2010] propose a dynamic routing algorithm that relies on a mobile sink to balance the workload generated by the relay task, and thus alleviate the energy hole problem. Leu et al. [2009] also present another strategy that relies on a mobile sink.

3.2.3 Load balance in WSNs

Load balance is a key factor for WSNs because an uneven and unbalanced workload leads to an inhomogeneous energy consumption, and, thus, decreases the network lifetime. In this context, Zhao et al. [2012] present a scheduling technique called Virtual Backbone Scheduling (VBS), which forms multiple overlapped backbones that work alternatively to balance the workload. VBSs are created by duty-cycling the nodes in such a way that only nodes that belong to the same virtual backbone will be active at a time slice. By adopting VBS, the traffic can be routed throughout different backbones that are periodically rotated. The scheduling problem is shown to be $\mathcal{NP}$ -hard, so, the authors proposed an approximate algorithm that manages the nodes duty-cycle in order to balance the energy consumption.

Guan [Guan, 2008] uses a different approach to balance the load of the relay task. It is proposed a classification of sensor nodes into different levels according to their distance to the sink node (in hops). Using this classification, it is assumed that nodes establish multiple paths toward the sink, depending on the residual energy and the communication cost. Occasionally, the relay selection algorithm chooses a sub-optimal path besides the minimum energy path. Although the algorithm balances the relay workload, it needs to update the residual energy frequently to maintain this information up-to-date and available for the nodes that are communicating with the sink.

Another approach to minimize the uneven workload in WSNs is proposed by Ren et al. [2011]. They design an Energy-Balanced Routing Protocol (EBRP) by forcing packets to move toward the sink through the dense energy area, so they try to avoid nodes with relatively low residual energy. Similar to the work

presented in Guan [2008], nodes that are involved in any routing path needs to be aware of the residual energy information of their vicinity. This information can be costly to be maintained up-to-date.

Large-scale WSNs are shown to be inefficient when a plain topology is used [Heinzelman et al., 2002]. Thus, many works like Heinzelman et al. [2002]; Younis and Fahmy [2004] propose a hierarchical structure where some sensors are grouped into clusters and a cluster-header is elected to aggregate and deliver the group data to the sink node. Those cluster-based approaches usually provide a load balance scheme to avoid the excessive workload designated to the cluster-head. Mandala et al. [2006] proposes an algorithm that organizes cluster-heads into multiple chains such that the traffic load can be evenly distributed among different chains. Thus, they avoid the hot spot formation not only inside the clusters, but also outside them. By adopting a multi-chain approach that ensures different routes with different nodes, there is no guarantee that the communication will be performed through shortest paths. So, the overall energy consumption is higher than with a shortest-path based routing scheme.

In this Chapter, we are not interested in the use of different deployment strategies or mobile nodes. Instead, we estimate a centrality metric able to characterize the energy hole problem, and we use it to balance the relay task workload.

3.3 Sink betweenness and wireless sensor networks

SBet is similar to betweenness in the sense that it represents the centrality in terms of shortest paths, with added nifty properties for the WSN context. For example, SBet is much cheaper to calculate than betweenness. In Oliveira et al. [2010] we presented a distributed algorithm that uses only $2n$, messages, with n representing the number of nodes, to calculate SBet in non-weighted graphs. This algorithm was firstly introduced in the Master Thesis of Eduardo M. Oliveira [Oliveira, 2010], however, in order to make this document self-contained, it is reproduced in Section 3.4. Although it is necessary to perform two

floods, commonly, shortest path routing algorithms usually require one flooding to set up the routing structure [Nakamura et al., 2009]. Thus, the first flooding can piggyback the necessary data, and only another flooding is necessary to complete the algorithm. To calculate the betweenness by a similar approach, $2n^2$ messages are needed, which might be excessively costly for large WSNs.

Moreover, SBet provides a much better representation of the traffic pattern in WSNs than betweenness. For instance, Figure 3.1 shows a panorama of the distribution of betweenness and SBet. In this figure, the nodes are randomly distributed on the sensor field. The gray level of the nodes is proportional to its betweenness or SBet. The greater the betweenness or SBet are, the darker the point is. The sink node is represented by a triangle. When the sink is positioned at the center of the network, as shown in Figures 3.1a and 3.1b, both metrics are able to distinguish the nodes that concentrate more routes toward the sink. SBet is more selective and presents high values only in nodes that, in fact, participate in more paths to the sink. Betweenness presents more nodes with high values far from the sink, once it considers paths among all nodes. When the sink is located at the corner (Figures 3.1c and 3.1d), betweenness fails to represent nodes that participate in more paths to the sink, and lacks the ability of characterizing the traffic pattern of WSNs, while SBet maintains this desirable ability.

Figures 3.1e and 3.1f show smoothed histograms of these two metrics when the sink is at the center (green solid line) and when the sink is located at a corner (dashed red line). Notice the different values they span (Bet is more spread than SBet), and that the empirical distribution of Sbet is steeper than that of Bet. SBet is more stable than Bet and, furthermore, it is more selective since it is more concentrated in nodes that participate in more paths to the sink.

In order to quantify the relative stability of these measures we employ the Hellinger distance. It is defined for two discrete probability distributions $P = (p_1 \dots p_k)$ and $Q = (q_1 \dots q_k)$ as

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^k (\sqrt{p_i} - \sqrt{q_i})^2}.$$

The maximum Hellinger distance 1 is achieved when one distribution assigns

probability zero to every event to which the other one assigns a positive probability, and 0 only when they coincide. In our example, the Hellinger distances from the SBet to the Bet distributions are 0.76 and 0.70 when the sink is located at the center and at a corner, respectively. This fact corroborates with our hypothesis that the SBet and Bet distributions are quite different, regardless the sink position. We also calculated the Hellinger distance between the SBet when the sink is located at the center and when it is located at a corner. We obtained 0.088, indicating that the distribution of SBet does not change significantly with the sink location. In the case of the Bet metric, the Hellinger distance is 0.11. Thus, we can observe that the Bet metric is more sensitive to the sink location than the SBet.

The SBet ability of characterizing the traffic pattern of WSNs can be used for many purposes. For instance, on the one hand, the nodes with high SBet are more likely to be data fusion points, since they concentrate more paths toward the sink. On the other hand, those nodes are more likely to deplete their energy quickly due to the same reason, i.e., they concentrate more paths toward the sink. Thus, nodes with high SBet can be preferred or avoided in a data forward path toward the sink, depending on the scenario. In event-driven data fusion scenarios, the higher the SBet of a node, the better it is to participate in the data-collection infrastructure since it is a good candidate to be a data fusion point. For continuous data scenarios, SBet can be used to balance the relay task among sensors that belong to the same neighborhood.

Figure 3.2 depicts a hypothetical WSN where the sink is the triangle at the center of the sensor field, the dots are regular nodes, the lines represent all shortest paths toward the sink node, and the circumferences represent the hop distance. The innermost circumference represents hop distance zero, i.e., the sink node, and the outermost circumference represents hop distance two, i.e., nodes two hops far from the sink. It is worth to note that some nodes tend to present more shortest paths toward the sink than others. For instance, node A belongs to more shortest paths than node B, so the former is more likely to be a better data fusion point than the latter. On the other hand, node A is more likely to transmit more packets than node B, and, therefore, drain more energy when a shortest path based routing algorithm is used.

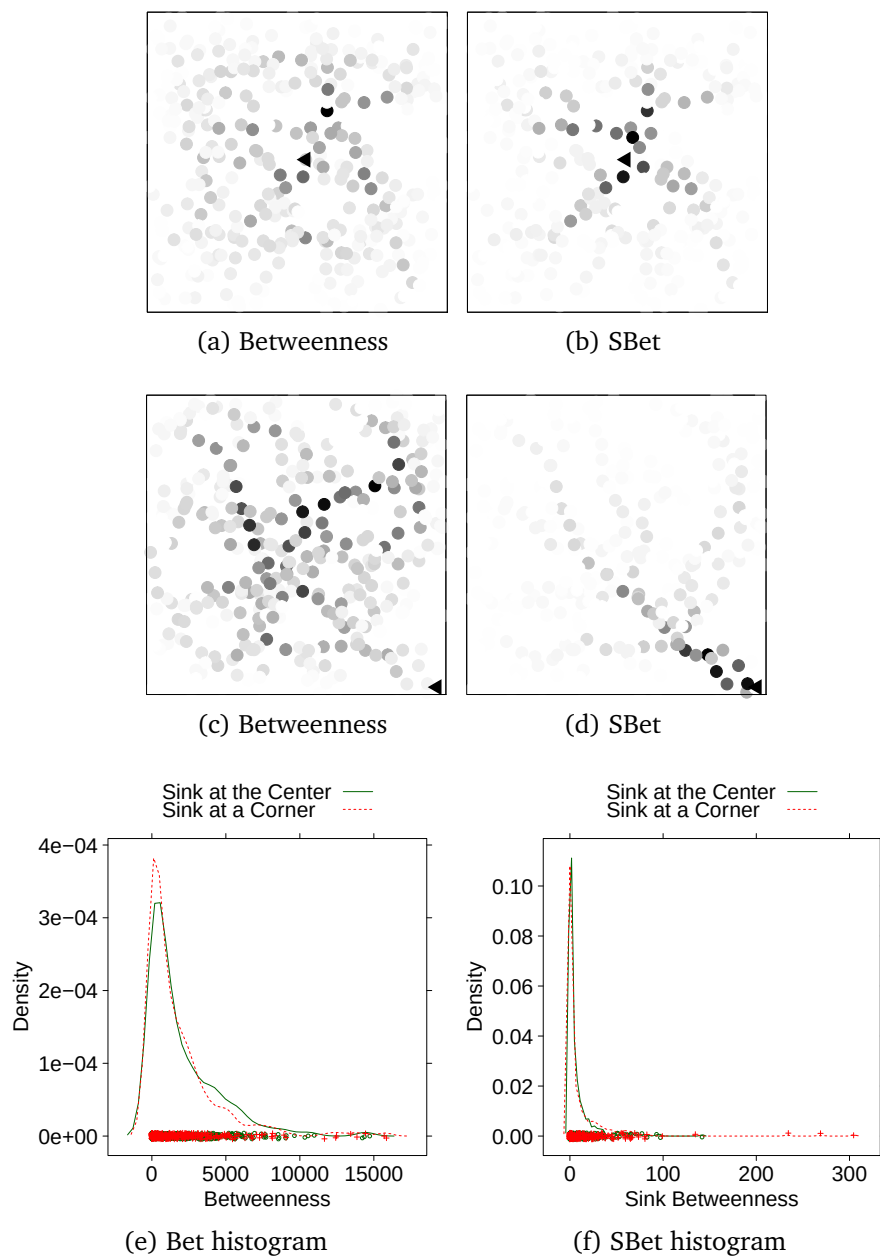


Figure 3.1: Examples of Betweenness and SBet values for two sink positions, center and corner and their respective histograms (the sink is represented by the triangle)

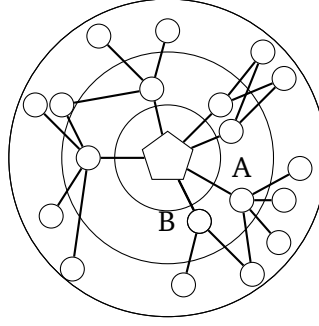


Figure 3.2: Node A is more central than node B in terms of number of shortest paths to the sink (the pentagon at the center)

Another important property of SBet in the context of the energy hole problem is its correlation with the energy spent by the nodes. As the transmission is usually the task that spends more energy, a metric that is able to indicate the nodes that are more likely to transmit more packets is useful to try to balance the workload. We showed in Section 2.4, that the SBet metric is able to characterize the energy consumption in a wide variety of WSNs scenarios. Thus, in the following Sections we show its application in the relay task workload balance, and thus, in the mitigation of the energy hole problem.

3.4 Distributed algorithm for sink betweenness

This section describes a way of calculating the SBet metric using only $2n$ messages, where n is the number of nodes. This algorithm firstly appears in the Master Thesis of Eduardo M. Oliveira [Oliveira, 2010]. Algorithms 1, 2 and 3 present a distributed procedure to calculate SBet for each node. Table 3.1 presents a brief description of the involved variables.

The algorithm has two phases: (i) a flooding initiated by the sink, and (ii) a flooding reflected by the border nodes (the ones that do not participate as an intermediate node of any shortest path to the sink). Those phases are described as follows.

Table 3.1: Description of variables used in Algorithms 1, 2 and 3

$role_t$	current role of t . For instance, border, sink, or regular node
$sonsPaths_t$	array with number, and frequency of $SP_{t \rightarrow sink}$ in ψ_t
$paths_t$	number of shortest-paths from sink to t
$hops_t$	distance in hops from sink to t
$sBet_t$	SBet of node t
$neighborMaxSBet_t$	the highest SBet received by t
$nextHop_t$	id of the selected node by t as next hop to the sink

Algorithm 1 Sink Betweenness distributed algorithm: node initialization

```

1: procedure INITIALIZESNODE T
2:    $role_t \leftarrow Border$ 
3:    $sonsPaths_t \leftarrow \emptyset$ 
4:    $paths_t \leftarrow 1$ 
5:    $hops_t \leftarrow 0$ 
6:    $sBet_t \leftarrow 0$ 
7:   if  $i_{id} = 1$  then
8:     Initializes P as Hello
9:      $role_t \leftarrow Sink$ 
10:     $Phops \leftarrow hops_t$ 
11:     $Ppaths \leftarrow paths_t$ 
12:    broadcast P
13:  end if
14: end procedure

```

Algorithm 2 Sink Betweenness distributed algorithm: dealing with hello packets

```

15: procedure RECEIVES PACKET P
16:   if  $Ptype = Hello$  then
17:     if  $Phops_t + 1 < hops_t$  then
18:        $hops_t \leftarrow Phops + 1$ 
19:        $Phops \leftarrow hops_t$ 
20:        $paths_t \leftarrow Ppaths$ 
21:     end if
22:     if  $Phops + 1 = hops_t$  then
23:        $paths_t \leftarrow paths_t + Ppaths$ 
24:     end if
25:     if  $Phops > hops_t$  then
26:        $role_t \leftarrow Relay$ 
27:     end if
28:     schedule broadcast P
29:     schedule Send Border
30:   else ▷  $Ptype = Border$ 
31:     if  $hops_t < Phops$  then ▷ From some descendant
32:       for  $j \leftarrow 1..length(PsonsPaths)$  do
33:          $sonsPaths_t[j] \leftarrow sonsPaths_t[j] + PsonsPaths[j]$ 
34:          $PsonsPaths[j] \leftarrow sonsPaths_t[j]$ 
35:       end for
36:       for  $k \leftarrow 1..length(sonsPaths)$  do
37:          $sBet \leftarrow sBet + sonsPaths_t[k] \times (paths_t/k)$ 
38:       end for
39:       broadcast P
40:     end if
41:   end if
42: end procedure

```

Algorithm 3 Sink Betweenness distributed algorithm: sending border packets

```

43: procedure SEND BORDER
44:   Initializes P as Border
45:   if  $role_t = Border$  then
46:      $Phops \leftarrow hops_t$ 
47:      $Ppaths \leftarrow paths_t$ 
48:      $PsonsPaths \leftarrow (paths_t, 1)$ 
49:     broadcast P
50:   end if
51: end procedure

```

3.4.1 Node initialization (Lines 1–14)

The sink initializes the process by sending a flooding message to the entire network with a *Hello* packet (Line 12). In our implementation, the sink is arbitrarily chosen.

3.4.2 Dealing with the Hello packet (Lines 16–29)

Calculates, for each node t , its distance to the sink, and the number of shortest paths from the sink to t (the numerator in the Eq. 2.4). Both distances are measured in number of hops. The *Hello* packet contains two integer variables representing these two distances.

3.4.3 Sending the Border packet (Lines 44–49)

For those nodes that have not had their $role_t$ changed to *sink* (Line 9) nor to *Relay* (Line 26), the initial definition, i.e., *border*, is maintained (Line 1). Border nodes are responsible for broadcasting the *Border* packet (Line 49) after the reception of the *Hello* packet.

3.4.4 Dealing with the Border packet and calculating SBet (Lines 31–39)

The *Border* packet structure contains the same *hops* field shown in the *Hello* packet, and two other fields: *sbet*, and *sonsPaths*. Line 31 checks whether the *Border* packet was sent by a node belonging to ψ_t ¹ (its *hops* field is greater than $hops_t$). Notice that all nodes in ψ_t are farther from the sink than t .

For a node t , the *sonsPaths* field is an array in which each key represents, separately, the number of paths from nodes presented in ψ_t , i.e., one position for each different value of the *path* variable from ψ_t nodes. The value assigned to each *sonsPaths* key is the number of occurrences of the key. For the sake of illustration, Table 3.2 shows the content of the *sonsPaths* array for each node contained in the illustrative network shown in Figure 3.3. For each message

¹see its definition in Eq.(2.4)

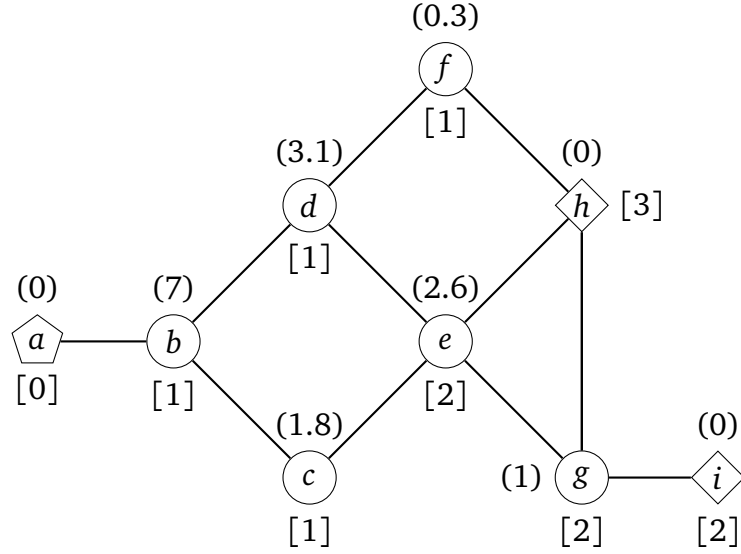


Figure 3.3: An illustrative network with sink (pentagon) and sensors (circles, and lozenges). For each sensor, we have the SBet value within parentheses, and number of paths from the sink within brackets. The circular-shaped sensors have the Relay role, while the lozenges have the Border role.

received from a node in ψ_t , t refreshes its *sonsPaths* (Line 33), and also updates the *Border* packet *sonsPaths* with its own table (Line 34), and calculates SBet (Line 37).

For example, the process of calculating SBet for node c is evaluated as follows. The first flooding defines to 1 the number of paths from the sink node to c ($paths_c$). The second flooding brings to c a *Border* packet with the *sonsPaths* array $[(2, 3), (3, 1)]$ (see Line 3 of Table 3.2), i.e, there are three nodes (e, g, i) with two paths to the sink, and one node (h) with three paths to the sink. Those four nodes comprise the ψ_c set. It is important to point out that the set ψ is merely a conceptual terminology, and the node does not need to maintain a structure to store it, or even know its own ψ set, since the distributed algorithm herein proposed does not need to consider this aspect to calculate SBet. Now c has the necessary data to calculate its SBet. If we apply the SBet definition we have the following expression: $SBet_c = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3}$.

The *sbet* field presented in the *Border* packet aims at informing neighbor-

Table 3.2: The content of *Border* packet field *sonsPaths*, and ψ set for each node of the network shown in Figure 3.3. Each position of the array is formatted as (key, value).

Node	<i>sonsPaths</i>	ψ_{node}
<i>a</i>	$[\emptyset]$	$[\emptyset]$
<i>b</i>	$[(1,3), (2,3), (3,1)]$	$[c, d, e, f, g, h, i]$
<i>c</i>	$[(2,3), (3,1)]$	$[e, g, h, i]$
<i>d</i>	$[(1,1), (2,3), (3,1)]$	$[e, f, g, h, i]$
<i>e</i>	$[(2,2), (3,1)]$	$[g, h, i]$
<i>f</i>	$[(3,1)]$	$[h]$
<i>g</i>	$[(2,1)]$	$[i]$
<i>h</i>	$[\emptyset]$	$[\emptyset]$
<i>i</i>	$[\emptyset]$	$[\emptyset]$

ing nodes about its own SBet value, which will be necessary to build the data collection algorithm.

The transmissions of *Hello* and *Border* packets, shown in Lines 28 and 29, respectively, are delayed to allow each node to receive packets from its entire neighborhood. This feature allows nodes to transmit only one of each packet type (*Hello* and *Border*) and ensure an overhead cost of $O(n)$ (two floodings). In our experiments, both delays are proportional to the distance to the sink, which is stored in the $hops_t$ variable. The best results were obtained by choosing a delay directly proportional when forwarding the *Hello* packet, and inversely proportional when forwarding the *Border* flooding.

3.4.5 Analysis

The distributed algorithm calculates the exact SBet value of all nodes. However, in real world scenarios, data may not be readily available due, for instance, to packet loss, which can be caused by collisions, radio error, interference, for instance. We assume that such problems should be tackled by underlying network layers and the SBet algorithm does not need to handle them. Another possible source of missing data regarding the distributed algorithm is the schedule time for the *Border* packet sending. This problem occurs when a packet arrives after

the timer expires and, thus, the node calculates and forwards its SBet without the information from its neighbors. As described in Section 3.4.4, a schedule delay inversely proportional to the distance to sink is, in our simulations, the best way to circumvent this problem.

For a node t , the delay used in the scheduling process, represented in Line 29 of the distributed SBet (Section 3.4), was $ke^{-\text{hops}_t}$ seconds, where k is a constant. The exponential delay herein used suffices to all *Border* packets from the nodes presented in ψ_t to arrive at node t before the timer expires. Constant k should be fine tuned in real world scenarios to compensate the channel latency. It is noteworthy that the largest delay is k/e sec for nodes one hop away to the sink.

Another aspect that should be assessed is the size of the array *sonsPaths* of the *Border* packet once, in WSN environments, the payload size is typically small; for instance, currently it is 114 bytes for the Telos B mote. Thus, we should consider this constraint in the use of the proposed algorithm.

A simple approach to deal with large messages is to use packet fragmentation. Table 3.3 shows the average number of packets necessary to send the *sonsPaths* variable when fragmentation is used. We vary the number of nodes from 128 to 1024, and the network density (average number of neighbors) from 10 to 20. The values were estimated considering that every entry on the *sonsPaths* array is an integer represented by 32 bits, and the packet payload is limited to 114 bytes. For this scenario, we present the average number of packets. As the nodes are placed on the sensor field by following a uniform random distribution, some topologies are likely to be not fully connected. We also present within parentheses the average number of nodes that belong to the same connected component of the sink node. Observe that the larger the network is, the more packets are needed to properly transmit the *sonsPaths* variable due to the increase of the message size.

Figure 3.4 shows the average number of packets necessary to transmit the *sonsPaths* variable for different network sizes and densities at each hop level. Observe that the closer the node is to the sink, the higher is the number of packets, especially for large and dense networks such as with 1024 nodes and density 20. Nodes close to the sink are likely to have large ψ set and, consequently, more

Table 3.3: Necessary overhead to calculate the SBet metric

# of Nodes		Density	Average # of Packets
128	(95.10)	10	101.27
128	(94.93)	15	104.00
128	(95.60)	20	105.77
256	(193.97)	10	232.03
256	(200.06)	15	256.70
256	(198.86)	20	265.27
512	(392.43)	10	557.23
512	(415.10)	15	673.40
512	(419.33)	20	733.03
1024	(806.36)	10	1465.63
1024	(846.63)	15	1756.30
1024	(868.43)	20	1980.00

elements in the *sonsPaths* array.

The *sonsPaths* variable stores a histogram of the number of paths toward the sink for the nodes in ψ_t . If the overhead is excessively high and unsuitable for the application requirements, a natural way of decreasing the size of this variable is increasing the bin size of the histogram whenever the *sonsPaths* size exceeds a threshold. This technique might be useful mainly for the nodes nearby the sink (see Figure 3.4). This approach introduces an additional error on the calculation of the SBet metric and its adoption should be carefully evaluated according to the application. Observe that SBet should be calculated only during the initial set up phase of a WSN, and should be periodically updated to deal with eventual topology changes. In static WSNs, topologies changes seldom occur, and the cost of calculating SBet is worth for several applications, as the one present in the following Section. We also apply this distributed procedure in order to devise a data collection algorithm that takes advantage of the knowledge of the SBet of the node in order to choose the relay that is more likely to concentrate more routes to the sink, and thus, favors the data fusion procedure [Oliveira et al., 2010].

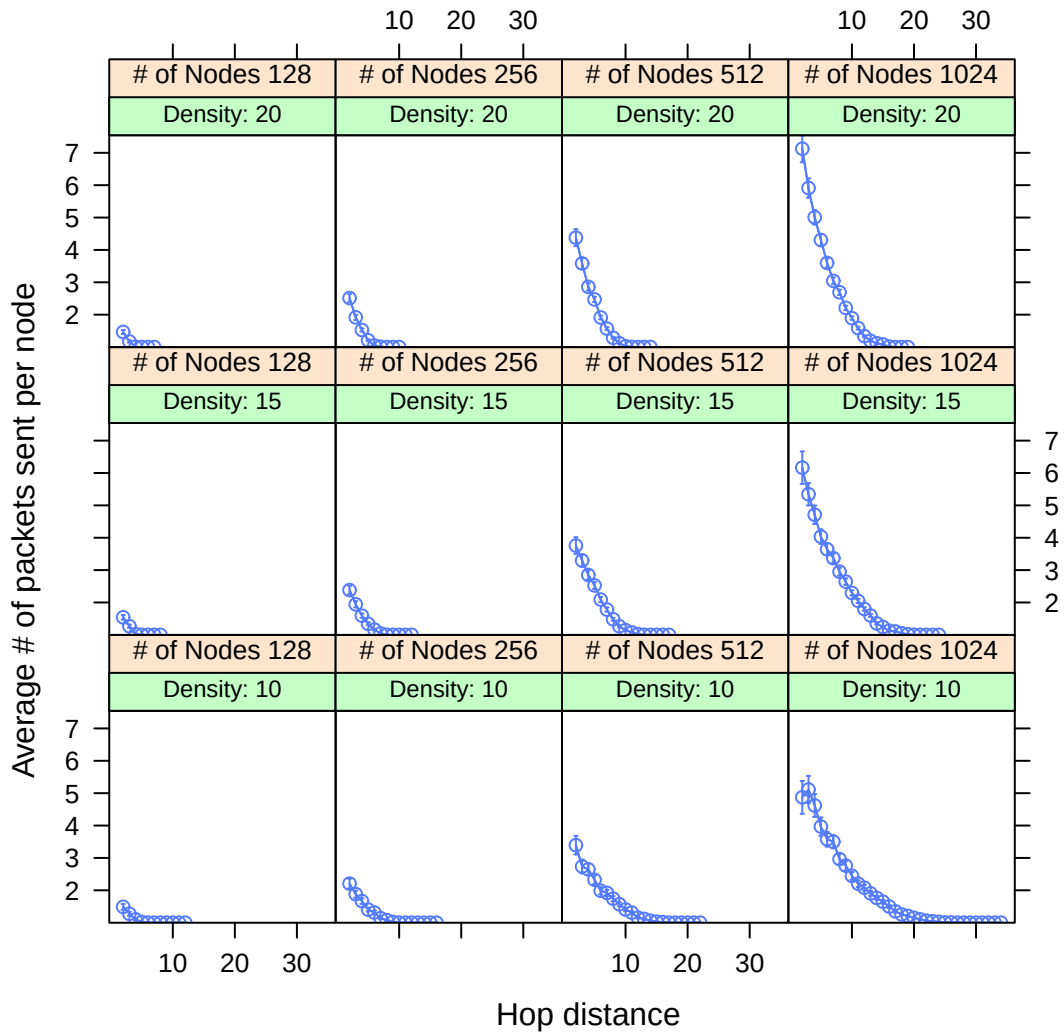


Figure 3.4: Average number packets sent per node upon the hop level

3.5 Using the Sink Betweenness metric to address the energy hole problem

3.5.1 Methodology

The most common solutions found in the literature to alleviate the energy hole problem rely on new deployment models that tend to place a higher node density surrounding the sink node. Although those strategies present some drawbacks such as lower coverage and high probability of data collisions within the high density areas, the energy hole problem can be alleviated once there are more nodes to share the relay task nearby the sink. The Q-model introduced in Liu et al. [2007] is a good example of this strategy.

Consider a WSN that performs a continuous data collection where sensor nodes report information to the base station periodically through a tree-based routing infrastructure. For this initial analysis, we consider a simple routing strategy based on shortest paths, where nodes choose their parents after a flooding initiated by the sink node [Al-Karaki and Kamal, 2004; Nakamura et al., 2009]. We used a network consisting of 300 Mica2 CC1000 nodes, with their transmission power level set to -10 dbm, deployed on a square field of side 150 m.

Figure 3.5 presents the percentage of transmitted messages as a function of the hop distance for both deployment strategies, URP (Uniform Random Placement) and Q-model. Observe that, as stated by Wu et al. [2008], the Q-model is able to distribute the relay task more evenly than the URP model. In the case of URP, more than 60% of the transmissions were performed by nodes situated one hop from the sink. When the Q-model was used, the nodes situated in the first hop transmitted less than 30% of the total number of packets.

Despite the more even distribution of the transmitted packets among nodes located in different hop distances, the deployment strategy cannot do much about the distribution among nodes situated at the same hop. For instance, in Figure 3.6 we observe that both URP and Q-model lead to an uneven number of transmissions among nodes at the same hop. In this example, we depicted only the first level but this behavior is observed in all levels with different intensities. It means that the workload is more intensely uneven in the first level and progressively

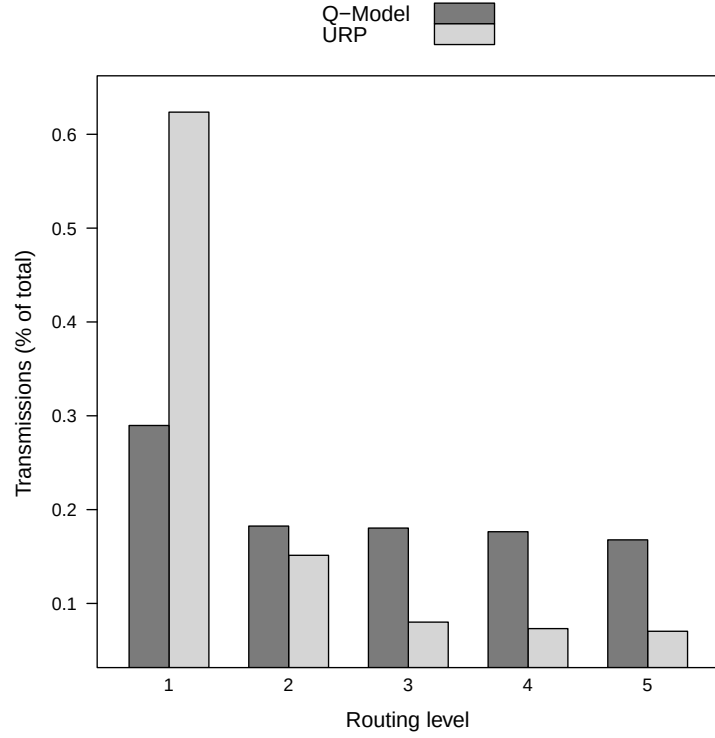


Figure 3.5: Percentage of transmitted messages as a function of the hop distance

decreases until the last level, once these nodes only transmit their own sensed packets.

Observe in Figure 3.6 that when we use the URP model a few nodes transmit much more packets than other nodes and the difference can reach about five times more packets. For the Q-model, the total number of transmissions decreases because there are more nodes only one hop distant to the sink but the general behavior is similar, and the difference is about five times more packets as well. This situation illustrates the fact that the deployment strategy can alleviate the energy hole among the hop distance, but it is still necessary a data-collection scheme that strives to balance the workload of the relays nodes.

We are therefore interested in using the SBet metric to devise a topology-aware data collection scheme able to alleviate the energy hole problem that happens among nodes sharing the same hop distance to the sink. In that direction, we propose a new data-collection algorithm where every node $v_i \in V$ knows the

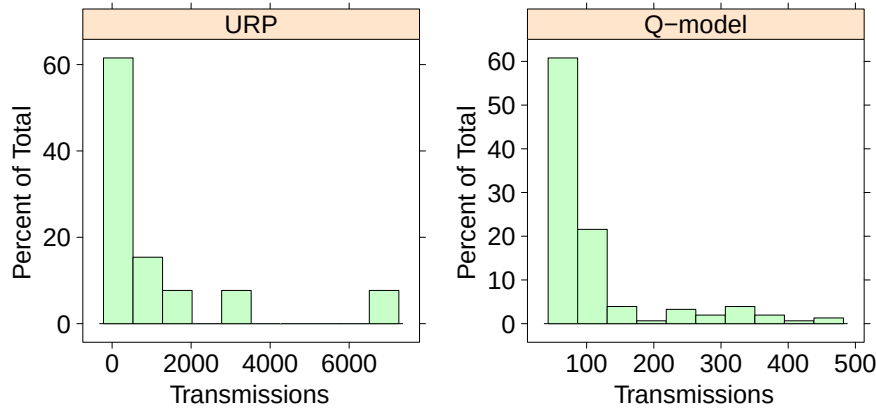


Figure 3.6: Uneven distribution of transmissions for nodes located one hop from the sink

SBet of the nodes that belong to its neighborhood. To do so, the same algorithm shown in Section 3.4.5 is used to calculate the SBet of all nodes, and during the second flooding the nodes piggyback their own SBet. Thus, with the same cost, all nodes can be aware of their neighbors' SBet. Therefore, every time a node has a packet to transmit (it can be its own packet or a relay packet), the node can use the SBet of its neighbors to choose which neighbor will be used to relay the packet. The hypothesis we will verify is that making this choice in a per packet basis and balancing the load among its neighbors guided by the SBet, the workload among neighbors can be distributed as even as possible. As a node with high SBet is more likely to be used as a relay by its neighbors, our algorithm uses the SBet to decrease the probability of those nodes to be chosen, and thus, balance the load among the neighbors.

The decision rule used to choose the next hop of node $v_i \in V$ is performed as follows:

1. **SBet calculation and announcement:** we apply the distributed algorithm described in Section 3.4.5 to calculate the SBet, and piggyback its value (announcement) into the packet of the second flooding. The node must calculate its own SBet before sending this packet.
2. **Neighbor filtering:** to ensure the creation of the shortest paths, only neigh-

bors that are closer to the sink than node v_i are eligible as relays, where Ω_i is the set of eligible nodes. The algorithm used in the last phase provides this information.

3. **SBet normalization:** SBet is normalized based on the set Ω_i . This metric is called nSBet (normalized SBet). We observe that each neighborhood presents different values of SBet, and, thus, this process makes the values comparable.
4. **Neighborhood probability assignment:** a probability inversely proportional to nSBet is assigned to every eligible node $v_j \in \Omega_i$. This is the probability of node v_j be chosen as relay of node v_i .
5. **Relay selection:** the relay node is randomly selected according to the probability of item 4.

To define the probabilities of item 4, we use a parameter called “temperature” that controls how intensely the node v_i will try to avoid the choice of a neighbor with high SBet. The probability of node $v_i \in V$ to choose node $v_j \in \Omega_i$ as its relay at temperature T is:

$$\Pr_T^i(j) \propto e^{-\text{nSBet}(j)/T}, \quad (3.1)$$

thus, we calculate

$$k_T^i = \sum_{j \in \Omega_i} e^{-\text{nSBet}(j)/T}, \quad (3.2)$$

and use (3.2) to transform the values of (3.1) in probabilities, as follows:

$$\Pr_T^i(j) = \frac{1}{k_T^i} e^{-\text{nSBet}(j)/T}. \quad (3.3)$$

The Equation (3.3) represents the Boltzmann distribution (a.k.a the Gibbs Distribution) [Kirkpatrick et al., 1983] with the Boltzmann’s constant equal to one. Thus, when $T \rightarrow \infty$ all relay candidates have the same probability to be chosen, i.e., the process approximates to an uniform random distribution. When $T \rightarrow 0$, the node with the smallest SBet value will be chosen with high probability.

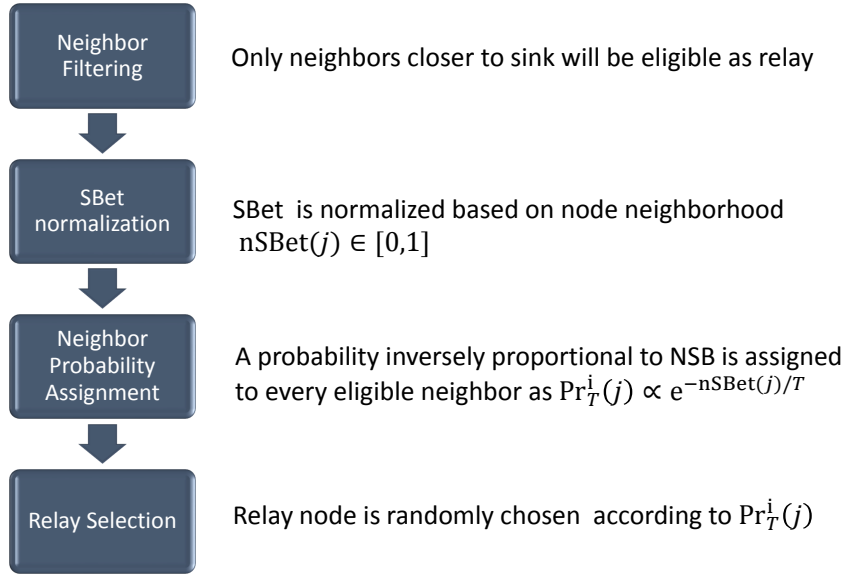


Figure 3.7: Relay selection decision rule

Both situations are not desirable for our problem. Thus, in Section 3.5.2, we evaluate a temperature that alleviates the bias of choosing the node that presents the highest SBet (the energy hole problem), while does not penalize the node that presents the smaller SBet value.

As this process is executed in a per packet basis, it is expected a load balancing of the relay task weighted by the SBet of the node.

Figure 3.7 shows a schematic view of the proposed relay selection for the proposed data collection algorithm.

3.5.2 Evaluation

The main goal of our simulations is to assess the performance of the data-collection schema described in Section 3.5. We performed a set of simulations to evaluate for each node located one hop to the sink the following metrics:

- **Max number of transmissions:** This metric gives an indication of how the proposed solution decreases the number of transmissions and, therefore, increases the lifetime of nodes. Observe that our main goal is to decrease

the dispersion of the number of transmitted packets. The ideal situation is when all nodes at same level transmit the same number of packets. Although the maximum is very sensitive to outliers, it gives a good idea of how our solution performs. An analysis of the dispersion metrics is less sensitive to outliers and provides another perspective of the performance evaluation.

- **Dispersion analysis:** We use the IQR (inter-quartile range), i.e., the difference between the 3th and 1st quartiles, to measure the dispersion of the number of transmissions of nodes located one hop to the sink. In this case, any other measure of dispersion could be used. We adopted IQR because it can be directly observed on the box plots and, thus, is intuitive. It is also less sensitive to outliers.
- **Entropy analysis:** This metric provides an upper bound for our analysis. The Shannon entropy of a discrete random variable is defined as $H = -\sum_i p(x_i) \log(p(x_i))$, where $p(x_i) > 0$ is the bin probabilities [Ebrahimi et al., 2010]. Note that the ideal solution would perfectly balance the load among the nodes, which means that nodes at the same hop distance will transmit the same number of packets. This ideal situation will exhibit the maximum entropy. We compare the entropy of our solution to the entropy of an ideal solution. It is known from information theory that the maximum entropy is obtained when all outcomes are equally likely, as uncertainty presents the highest value when all possible events are equiprobable. Thus, the desirable solution should present the highest entropy.

We conducted our assessment by comparing the performance of the proposed algorithm, namely `randomSbetTree`, with two other data-collection algorithms namely `simpleTree` and `randomTree`. All three algorithms are based on SPT (shortest path tree), and the main difference among them is how they choose the best shortest path to deliver the data. The `simpleTree` algorithm builds a structure where every node chooses a fixed parent forming a tree. The tree formation is started by the sink node by flooding the network. When a node receives the first flooding packet, it chooses its parent and forwards the flooding message. This

Table 3.4: Simulation scenarios used in the randomSbetTree algorithm analysis

Parameter	Value
Sink node	1, the center-most node
Network size	{100, 200, 300, 400} nodes
Sensor field	Square, {750, 1000, 1250} m of side
Deployment model	URP
Simulation time	3600 s
Data rate	1 packet/min
Collision model	Additive Model
Routing Algorithms	randomSbetTree, simpleTree, randomTree
Temperature (T)	[0.001, 100]
Sensor model	Mica2 CC1000
Transmission Power	10 dbm
Application	all nodes report packets periodically

algorithm is also known as delay tree [Parsa and Garcia-Luna-Aceves, 1998]. The randomTree algorithm is a modification of the simpleTree idea that tries to randomly balance the relay workload of nodes. Thus, instead of choosing the parent by the first flooding packet, the node waits to receive the flooding packet from every neighbor and creates a vector of eligible neighbors Ω_i . For every packet, it uniformly randomly chooses one node v_j from Ω_i as the relay for that specific packet. This approach is the same as $T = \infty$ in the randomSbetTree. We used the common application that all nodes periodically send the same amount of data toward the sink node.

Table 3.4 presents the simulation scenarios we evaluated. A Monte Carlo simulation was performed, replicating independently 30 times each situation indexed by the parameters shown in Table 3.4. This number of replications was considered sufficient for hypothesis testing sample mean differences at 95% significance level.

We use the R package version 2.12 [R Development Core Team, 2009] for node deployment and statistical analysis, Omnet++ simulator version 3.3p1 for discrete event simulation, and Castalia version 2.3b² for WSN models. Both wireless channel and MAC models are already available in Castalia; the data-

²Castalia: A simulator for Wireless Sensor Networks, <http://castalia.npc.nicta.com.au>

collection algorithms and application models were implemented as specified above. We also used the Mica2 CC1000 radio module available in Castalia.³

3.5.3 Results

In the following, we present the evaluation results of the algorithm proposed in this work. All the following results show the behavior of the number of transmissions of the nodes located one hop from the sink. This behavior is similar for other hop distances but it is smoother, once the workload tends to be more even when the distance to the sink increases.

Figure 3.8 shows the behavior of the randomSbetTree algorithm when the parameter T is varied for all scenarios of evaluation. In the plot, the lower and the upper box limits represent the 1st and 3th quartiles, respectively, the whiskers represent a range 1.5 times the IQR (interquartile range), and the points that eventually appear below or above the whiskers range are considered outliers. The black dots represent the mean value. We have one box plot for each scenario, being 12 variations of field size and number of nodes, for 16 different values of T . The y-axis, i.e., the number of transmissions, is shown in log scale.

We can easily observe that, typically, low and high temperatures lead to higher IQRs (box height) than mild ones. This is an expected result as low temperatures will make the nodes with smaller SBet very likely to be chosen as a relay, while high temperatures tends to make a near uniform choice. In some scenarios, such as for 400 nodes on a $750 \times 750 \text{ m}^2$ field, we can observe that the IQR is very small for T between 0.3 and 1, and start increasing when $T > 1$. This represents the sweet spot where under those temperatures the load balance strategy works better. This behavior is replicated, with different intensities, for all other scenarios. Some scenarios such as with 100 nodes on a $1250 \times 1250 \text{ m}^2$ field, the temperature does not influence significantly the results. This happens because the scenario is so sparse that there is no opportunity to balance the relay workload. Thus, we observe that the denser the scenario, the better the load balancing due to the fact that there are more shortest paths to choose. In a general

³Seeds, sources and scripts can be obtained upon request from the first author.

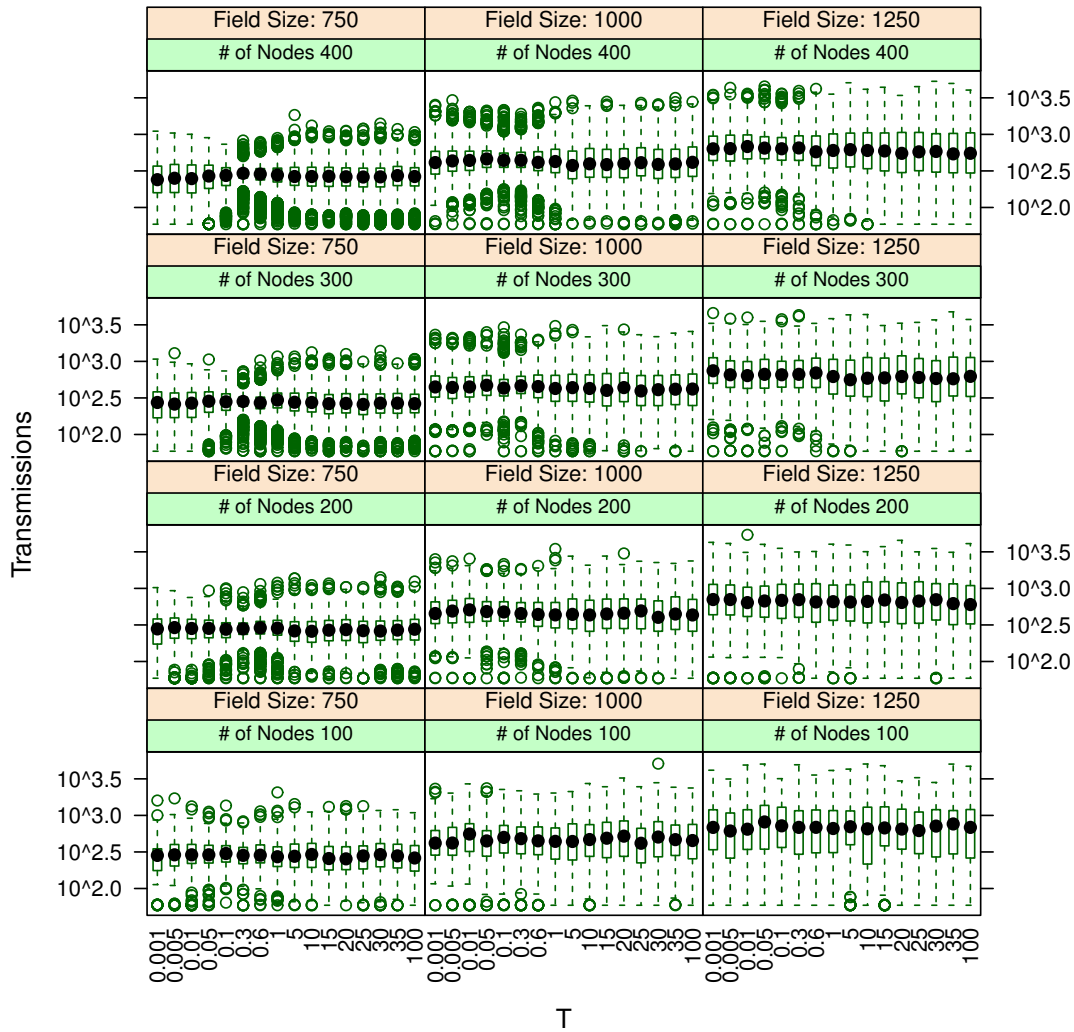


Figure 3.8: Behavior of the randomSbetTree algorithm upon varying the parameter T

way, in our scenarios, $T = 0.3$ represents a good choice. Thus, for the rest of our analysis, we consider the randomSbetTree under this temperature.

Figure 3.9 shows a boxplot with similar features as Figure 3.8, comparing the behavior of the three data-collection algorithms shown in Table 3.4 for all evaluated scenarios. Observe that when the density is low, for example when the field size is 1250 and the number of nodes is 100 the performance of the

three routing algorithms is similar. This fact can be explained because for this situation the density is too low (about 4 neighbors in average) that after the Neighbor Filtering step detailed in Section 3.5, there are not many eligible nodes to be chosen as relay nodes. Thus, the algorithms perform somewhat similar, with a small advantage to the randomSbetTree. When the density increases, we can observe that the randomSbetTree performs better than the others. For instance, when the field size is 1250 and the number of nodes is 400, the average density is about 13 nodes and we can see that the IQR is much lower for randomSbetTree than for the other algorithms. When the field size is 750 the average density is about 9, 20 and 26 and we can see that the randomSbetTree performs better than the other algorithms. This general behavior holds for all evaluated scenarios, i.e., only if the density is too low that the randomSbetTree is comparable to the other algorithms. For all other situations the randomSbetTree performs better. As we mentioned before, these boxplots are very useful for a qualitative analysis, and the following plots will show this general behavior quantitatively. An important behavior is that the average number of transmissions is almost the same for all routing strategies. This occurs because the expected number of packets that come from the overlying levels is the same, and the nodes that belong to the first level are also the same, in average.

Figure 3.10 shows the maximum number of transmitted packets from nodes located one hop away from the sink for all evaluated scenarios. Observe that the randomSbetTree presents a desirable property of not increasing significantly the maximum number of packets when the number of nodes increases, while the other algorithms lack this ability. It means that even when we increase the number of nodes and all those nodes are transmitting packets to the sink, the network lifetime will be maintained when the randomSbetTree is used. This fact can be explained once the randomSbetTree performs a more even distribution of the relay task among the nodes at the same hop distance. We expect that the number of nodes present in a given hop distance will increase proportionally to the increase of the total number of nodes. Thus, an even data-collection strategy shows this property. The difference of the maximum number of transmitted packets of randomSbetTree increases when the network density increases as well. For instance, for 400 nodes the max number of transmitted packets of randomSbetTree

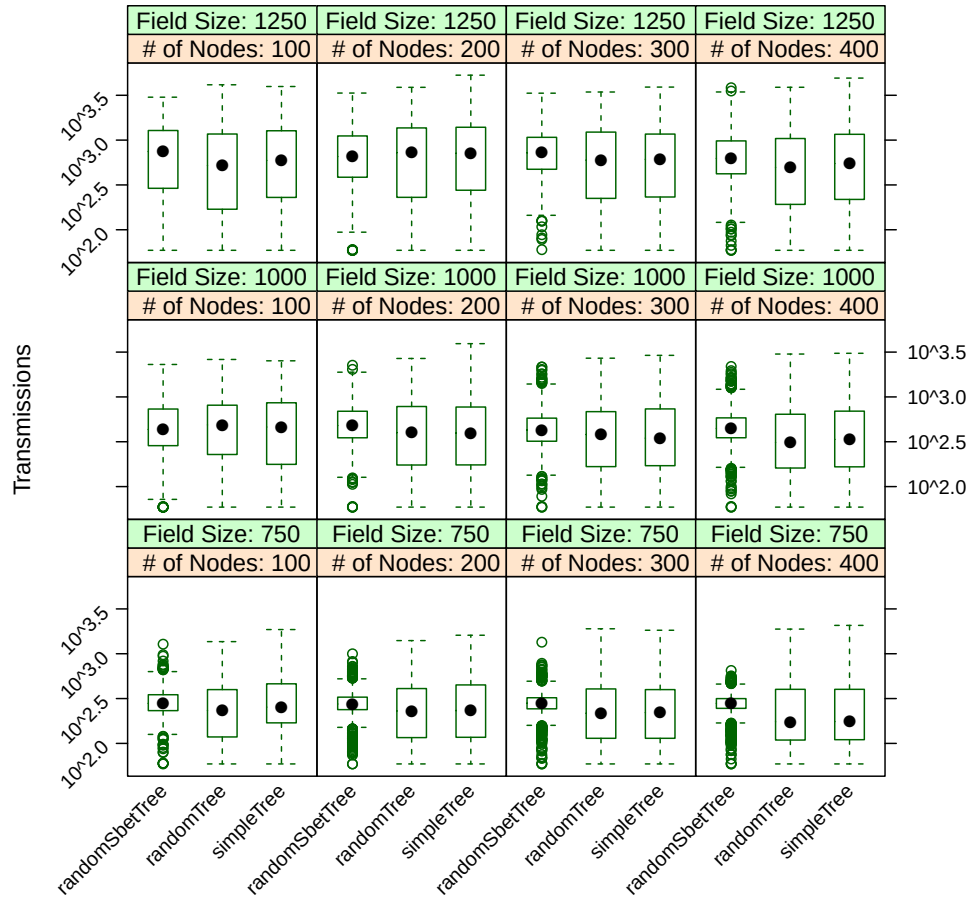


Figure 3.9: Analysis of the number of transmissions

is 70%, 40% and 25% lower than the second best algorithm for field sizes of 750, 1000, and 1250, respectively. With 100 nodes, we cannot see a substantial difference among the three routing algorithms once the density is too low (about 4 to 6 neighbors in average), so there is not enough possibility to balance the workload. Note that both simpleTree and randomTree do not show considerable differences in terms of the maximum number of transmitted packets.

Figure 3.11 shows the behavior of the IQR as a function of the adopted data-collection algorithm for all evaluated scenarios. This metric is more robust to outliers than the maximum number of packets. We observe that randomSbetTree

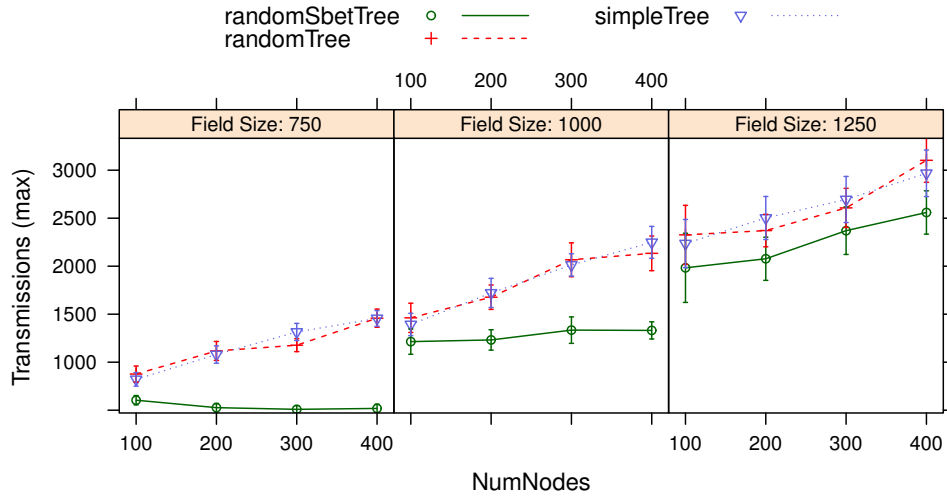


Figure 3.10: Max number of transmissions upon varying the number of nodes

consistently leads to a lower IQR. It reinforces our hypothesis that the distribution of the relay task is more even than the other algorithms. With this metric we show that not only the maximum value is decreased but the minimum value is increased accordingly. Thus, all nodes tend to relay a number of packets closer to the average value than when other algorithms are used. For instance, when the number of nodes is 400, the randomSbetTree presents IQR 80%, 55% and 35% lower than the randomTree for field sizes of 750, 1000, and 1250, respectively. Observe that for IQR, the randomTree performs slightly better than simpleTree.

Finally, Figure 3.12 shows the behavior of the entropy of the number of transmissions observed on the nodes one hop away from the sink, with respect to the ideal case. The results are shown as $H_{(.)}/H_{ideal}$, where $(.)$ represents the technique we are comparing to the ideal solution. In all situations, we generated a set of uniform random vectors with the same parameters (size and mean value) as the outcome of the technique in question, and considered the average entropy as the ideal. As mentioned in Section 3.5.2, this metric is a good indication of how far our solution is from the ideal situation compared to the other solutions: the closer to one, the better the routing. We observe that when the sensor field

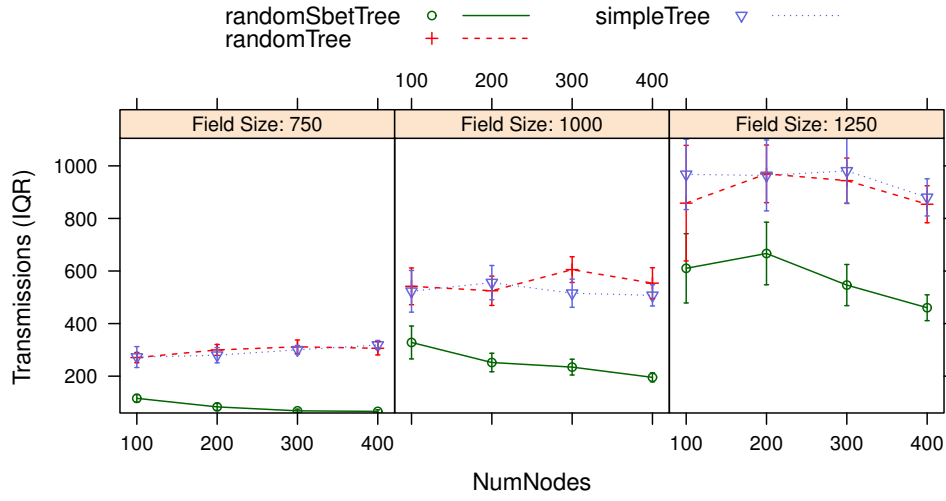


Figure 3.11: IQR of transmissions upon varying the number of nodes

is 750, our solution is very close to the ideal situation. This happens due to the higher density observed in those scenarios (between 8 to 26 neighbors, in average). When the density is higher, the algorithm has more options in Ω_i and it can perform a better load balancing. Thus, we observe that for other values of the sensor field, the randomSbetTree is slightly further from the ideal solution. This is an indication that our solution performs better when the density increases. With 100 nodes and field size 1250, our solution is similar to the others because it cannot find options in Ω_i . The randomTree and simpleTree approaches perform consistently worse than the randomSbetTree and they do not vary much with the network size.

3.5.4 Summary of the results

The randomSbetTree algorithm was tailored for continuous data applications in WSNs scenarios. The main goal is to balance the workload of the relay task as much as possible, and, thus, prolong the network lifetime. The most suitable algorithm to tackle this problem is the one that leads to all nodes to transmit the

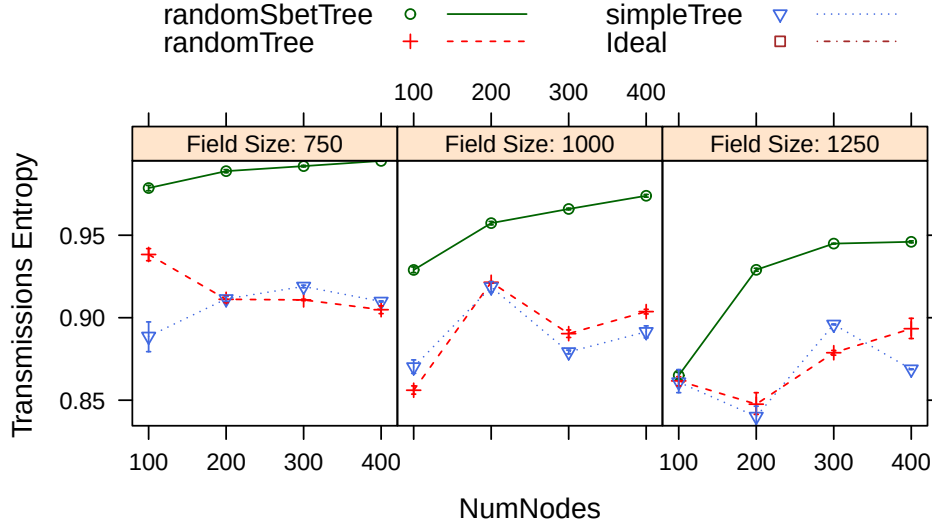


Figure 3.12: Relative entropy of transmissions upon varying the number of nodes

same number of packets. Due to the geographic constraints, it is not possible that with a single sink approach, all nodes transmit exactly the same number of packets, an effect known as energy hole problem. Thus, we proposed the randomSbetTree algorithm to alleviate the energy hole problem without the need of changing the deployment model, or to use multiple or mobile sink approaches. For this, we used the SBet metric in order to try to avoid the node with high SBet to be overloaded. We compared the results of the randomSbetTree algorithm to two tree-based algorithms, namely simpleTree, and randomTree. All algorithms herein evaluated are based on a shortest path routing strategy. We started our evaluation by fine-tuning the temperature parameter T of the randomSbetTree algorithm. By varying this parameter in a wide range, we observed a sweet spot for our scenarios when T is between 0.3 and 1. Thus, we also observe that 0.3 represents a good choice for all evaluated scenarios, because it decreases the IQR of the transmission packets, as shown in Figure 3.8. Furthermore, we adopted $T = 0.3$ to compare our algorithm to the simpleTree and randomTree algorithms. We presented qualitative analysis by using the boxplot (Figure 3.9) to show that our algorithm outperforms the others for all situations, and presents close results only when the density is low and all algorithms perform similarly. Moreover,

Figure 3.10 shows that the randomSBetTree algorithm is more scalable than the others, in the sense that it decreases the maximum number of transmissions, and decreases the growth rate of the maximum number of transmissions when the number of nodes increases. We also showed that the IQR of our solution is the lowest among the evaluated algorithms and presented these results in Figure 3.11. Finally, we showed that the randomSBetTree algorithm performs close to the ideal solution by using the entropy analysis presented in Figure 3.12. Thus, by using the SBet metric, we were able to devise a routing algorithm that alleviates the energy hole effect and is close to the theoretical ideal situation.

3.6 Chapter remarks

In this Chapter, we proposed the use of a novel centrality measure called Sink Betweenness to devise data-collection algorithms for WSNs. As shown in our assessment, this metric is suitable for dealing with two different scenarios in WSNs: event-driven data-fusion applications, and energy balancing in continuous-data application.

This measure's ability to characterize some features present in WSNs suggests other possibilities. Sink Betweenness can be used in a wide variety of applications, both in the design and operation of a WSN.

The proposed data-collection algorithms can be easily integrated with any routing protocol that relies on shortest paths. Thus, questions regarding routing algorithms such as clustering, multiple sinks, and other energy-related issues can be handled by the routing algorithm on top of the data-collection algorithm herein presented.

Regarding the WSN operation, we envision other topology-aware algorithms that use each node's value of SBet to increase the network performance. For instance, MAC algorithms can take advantage of using this measure to optimize the duty-cycle of the nodes to alleviate the energy consumption.

Low energy GPS-based localization in mobile ad hoc networks

“Belief is the death of intelligence”

Robert Anton Wilson

Location is one of the most basic topological features, however, it is also one of the most useful for ad hoc networks. With the advent of mobile devices, location based applications has emerged as a new trend. A wide variety of applications have used the location information in order to improve the user experience. Thus, location is a dominant user context information for many mobile applications, such as traffic, navigation, search, advertising, social networking, and personal reminders. While some applications are satisfied with a single location, many other services require or can take advantage of continuous location traces. In this context, GPS is the most common outdoor location sensor on mobile devices. However, the high energy consumption of GPS sensing prohibits it to be used continuously in many applications. In this Chapter, we propose a low energy assisted positioning solution that carefully partitions the GPS signal processing pipeline and shifts delay tolerant position calculations to the cloud. The GPS receiver only needs to be on for some milliseconds to collect the sub-millisecond level propagation delay for each satellites signal. Our solution works

in two different modes, considering two different scenarios. If a reference to a nearby object, such as a cell tower, is available, the server can infer the rest of the information necessary to perform GPS position calculation, if not, the server performs multiple hypothesis test and uses other geographical constraints to filter out the wrong solutions and calculates the receiver's position. Thus, the server application can derive good quality GPS locations from a few milliseconds of raw data. Compared to more than 30 seconds of heavy signal processing on standalone GPS receivers, we can achieve three orders of magnitude lower energy consumption per location tagging. We analyze the accuracy and energy benefit of our solution and use real user traces to show that it can save up to 80% GPS energy consumption in typical trajectory-based service scenarios.

4.1 Introduction

Location based applications have gained much attention in recent years mainly with the advent of mobile devices such as smart-phones, PC-tablets, and net-books. These devices facilitate the position data collection and allows some applications to make use of this data. For instance, mobile search is taking advantage of the local information to improve the user experience.

Among the most common location sensors, GPS is the most accurate providing small location error and allowing, for instance, high accurate applications such as turn-by-turn navigation. Therefore, the sensors commonly used to collect location data are energy hungry and tend to deplete the device's battery very quickly. Although GPS is the most accurate current location sensor, it is also the most energy hungry, thus, nowadays the use of GPS sensor is very restricted in mobile application scenarios.

Bearing in mind this constrained scenario, this work aim at studying the GPS theory and proposing some alternatives of using the GPS sensor in low energy consumption mode. The idea behind this proposal is to use a very low duty-cycle GPS mode, when it is possible, while the application requirements are met.

This work is organized as follows: Section 4.2 discusses the GPS theory necessary to understand the rest of the text; Section 4.3 details our proposal and shows the evaluation results, Section 4.4 presents some related work and

highlights the main differences of our approach; while Section 4.5 concludes and provides the final remarks of this Chapter.

4.2 GPS basics

The Global Positioning System (GPS) is a satellite based radio navigation service created by the U.S Department of Defense (DoD) and offer the U.S military accurate estimates of position, velocity and time. Even though there are many comprehensive books on GPS subject such as Misra and Enge [2006]; Borre et al. [2006]; van Diggelen [2009] and Kaplan and Hegarty [2005], the rest of this section sums up the basic GPS concepts spread on these books, which are necessary to the comprehension of the rest of this work.

The GPS system was developed to provide precise estimates, roughly, with position error of 10 m, velocity error of 0.1 m/s, and time error of 100 ns, all in terms of root-mean-square [Misra and Enge, 2006]. The system was planned to be available to an unlimited number of users and to provide two types of service: (i) Standard Positioning Service (SPS), and (ii) Precise Positioning Service (PPS). The former is offered to peaceful civil use and the latter to DoD-authorized users, protected by cryptography. The SPS signal used to be intentionally added with noisy in order to decrease the precision, but nowadays, this restriction was revoked and there are no significant difference, in terms of precision, between both signals.

The GPS system consists of three basic parts: space, control and user segments. The space segment is composed of a constellation of 24 satellites distributed in 6 orbital plans, being 4 satellite in each plane, in medium Earth orbit. Currently, there are 31 operational satellites to assure uninterrupted service. The constellation was planned in a way that all users with a clear view of sky have a minimum of four satellites in view, but is more likely that a user would see six to eight satellites [Misra and Enge, 2006]. The control segment is maintained and developed by the U.S. Air Force that monitors satellite orbits, maintains satellite healthy and GPS time, predicts satellite orbit and clock parameters, and updates satellite navigation messages. The user segment is composed of hundred of thou-

sands of U.S. and allied military users of the PPS service and tens of millions of civil, commercial and scientific users of SPS service.

Each satellite vehicle (SV) broadcasts an unique signal in direction of the Earth surface and the GPS receiver estimates its position based on the time of arrival (TOA) of satellite signals. Each satellite transmits continuously using two radio frequencies in the L-band, namely Link 1 ($L1$) and Link 2 ($L2$), being $f_{L1} = 1575.42$ MHz and $f_{L2} = 1227.60$ MHz. Two signals are transmitted on $L1$, one for civil and other for DoD-authorized users, while the $L2$ channel is used only by DoD-authorized users. The signal transmitted on $L2$ band is encrypted and is not considered in the rest of this work.

4.2.1 GPS signal

The GPS signal consists of three components: (i) carrier, (ii) ranging code, and (iii) navigational data. Basically, each satellite broadcasts the carrier frequency $L1$ modulated with a unique pseudo-random (PRN) code, namely C/A code. It is a CDMA code used to identify the satellites on the receiver side, and helps on mitigating the undesirable effects of multi-path propagation and interfering signals perceived on GPS receiver antenna. It's modulated with a binary-coded message consisting of the satellite health status, ephemeris (satellite position and velocity information), clock bias parameter, and almanac (reduced-precision ephemeris data).

The C/A code is a sequence of 1023 bits, called chips. The duration of C/A chip is $1 \mu s$, thus, the duration of each C/A code is 1 ms and the chipping rate is 1.023 MHz. The navigational data is transmitted in packets of 1500 bits sent at 50 bits/sec, with a bit duration of 20 ms. It takes 30 sec for one entire packet to be received. This is the main reason that sometimes we experience a long time until a standalone GPS outputs the position.

The C/A signal at satellite can be modeled as

$$s_{L1}^{(k)}(t) = \sqrt{2P_{tmt}} x^{(k)}(t) D^{(k)}(t) \cos(2\pi f_{L1} t + \theta_{tmt}),$$

where P_{tmt} is the transmitted signal power for signal carrying on $L1$, $x^{(k)}$ is the C/A-code sequence of satellite k , $D^{(k)}$ is the navigational data of satellite k , f_{L1}

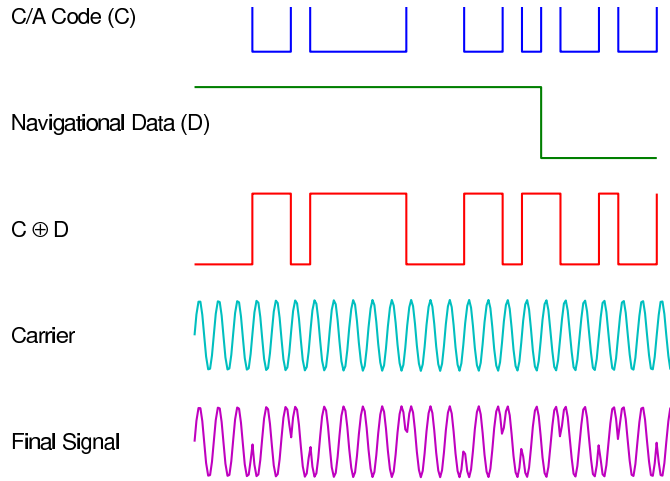


Figure 4.1: The structure of GPS signal. Source: Borre et al. [2006]

is the carrier frequency of $L1$ band, and θ_{tmt} is the phase off-set. The C/A signal on $L1$ perceived on the receiver can be modeled as

$$r_{L1}^{(k)}(t) = \sqrt{2P_{rcv}} x^{(k)}(t - \tau^{(k)}) D^{(k)}(t - \tau^{(k)}) \cos(2\pi(f_{L1} + f_D)t + \theta_{rcv}) + n^{(k)}(t),$$

where $P_{tmt} \gg P_{rcv}$, the subscripts tmt and rcv stand for ‘transmitted’ and ‘received’, respectively; $\tau^{(k)}$ is the signal propagation time for the satellite k , f_D is the frequency shift deviation by Doppler effect, and $n^{(k)}(t)$ is the noise. Figure 4.1 illustrates the structure of the GPS signal, modulated by binary phase-shift key technique (BPSK).

The C/A code repeats every millisecond, resulting in 20 repetitions per each data bit sent. The purpose of the C/A code is to allow a receiver to identify the sending satellite and to estimate the time elapsed for the signal propagation from that satellite until it reaches the receiver. The GPS signals take from 64 to 89 milliseconds to travel from a satellite to the Earth’s surface. Since light travels at 300 km/ms, in order to obtain an accurate distance measurement the receiver must estimate the signal propagation delay to the microsecond level. The millisecond (NMS) and sub-millisecond (subMS) parts of the propagation time are detected very differently: the NMS is decoded from the packet frames, while the subMS propagation time is detected at the C/A code level using correlations.

4.2.2 GPS receiver

A GPS receiver estimates its range to each visible satellite to solve its position equations. The range estimated by the receiver is known as pseudorange due to the uncertainty of those measures during this process. Although the receiver's position can be calculated using three range measurements from known points, in case of GPS, four known reference points (satellite positions) are necessary in order to accomplish the common clock bias range equivalent calculation. This follows from the fact that the GPS receiver is not tightly time synchronized with the satellites, so, all measures will be biased by the same value.

In order to calculate its position, a receiver needs three pieces of information:

- A precise time T ;
- A set of visible SVs and their locations at time T ;
- The distances from the receiver to each SV at time T , the *pseudoranges*.

Typically, these are derived from processing the signals and data packets sent from the satellites.

While the precise time and satellite locations are decoded from the packets, the distance measurement from each SV to the receiver is obtained using much lower-level signal processing techniques.

The pseudorange equations can be written as following:

$$\rho^{(k)} = \|X_s^{(k)}(t - \tau^{(k)}) - X\| + b + \epsilon^{(k)}, \quad (4.1)$$

where $\rho^{(k)}$ is the pseudorange from the receiver to satellite k , $X_s^{(k)}(t - \tau^{(k)})$ is the position of satellite k at the time of transmission, $\tau^{(k)}$ is the propagation delay, X is the receiver position, b is the common clock bias range equivalent, and $\epsilon^{(k)}$ represents the unknown error in measurement. Thus, we need at least four linearly independent equations to solve for the four unknown variables, three for position and one for the common clock bias. When more than four satellites can be visible, we have an overdetermined system of equations, generally solved by least square method applied to minimize the difference between the computed

pseudorange and its measured value, this difference is called the pseudorange residual.

Before calculate its own position, the receiver needs to conditioning the received signal in order to make it more manageable. Actually, the signal reaches the receiver with a very low power after travel about 20,000 km, about -10^{-14} W/m^2 (-128 dbm) and it's very sensible to many natural and man-made radio-frequency interference. Those undesirable interferences should be removed in the receiver's front-end. Additionally, the receiver carrier frequency must be down converted from about 1.6 GHz to something more manageable, by a factor between 100 to 1000. After conditioned, the signal must be amplified to ignite the A/D (analog-to-digital) convert. After A/D conversion, the receiver performs at least three tasks: (i) acquisition, (ii) tracking, (iii) position calculation.

The main purpose of the acquisition phase is to find visible satellites and initial values of the carrier frequency and code phase. Basically during the acquisition the receiver perceive the signal's frequency different from its nominal value due the Doppler effect. Typically the frequency vary about 10 kHz from its nominal value. As reported in Borre et al. [2006], a regular receiver typically performs the search sweeping bins of 500 Hz. To find out the code phase, the receiver needs to search a space of 1023 chips for every frequency bin. Thus, a search space of $1023(2^{\frac{10,000}{500}} + 1) = 41,943$ combinations is swept. For each bin of this search space the receiver correlates the local generated copy of satellite's C/A code with the observed signal. Acquisition is a time and CPU demanding task that is typically implemented in massive parallel correlation algorithms in order to decrease the processing time.

Code phases change over time as the satellite and the device on the ground move. Thus, GPS receivers use a tracking mode to adjust previous Doppler frequency shifts and code phases to the new one. Using the results of the acquisition phase as starting point, the receiver is able to reproduce a local copy of the signal transmitted by the satellites that will be used to track the signal. During the tracking step, the code phase and frequency estimates are refined and the receiver keep track of these measures during the time. Basically the tracking is comprised of two parts: (i) code tracking, generally implemented as a delay

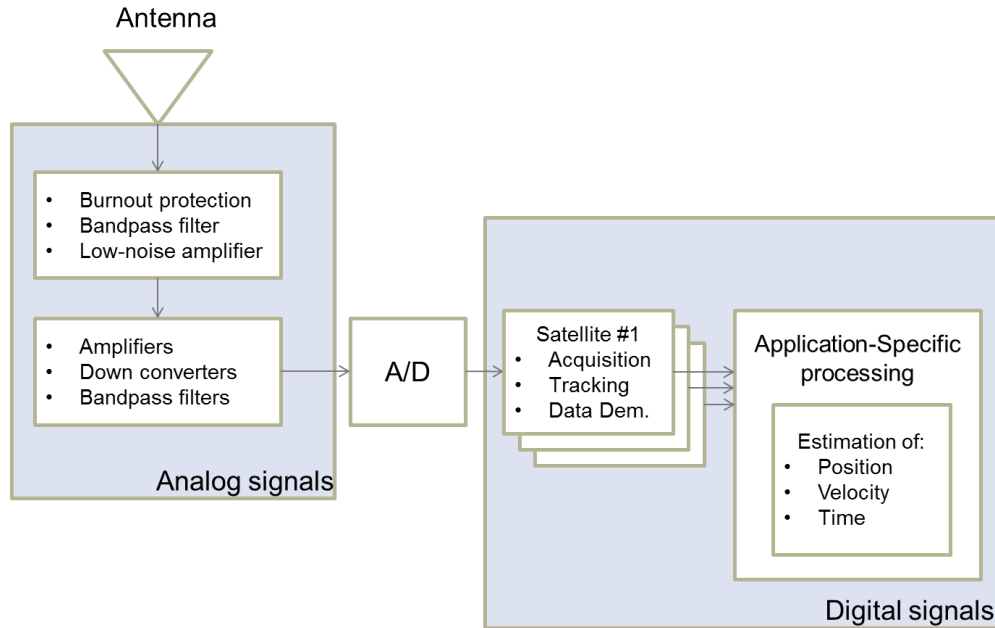


Figure 4.2: A schematic view of a GPS receiver: analog and digital signal processing

lock loop (DLL) where local copies are correlated with the observed signal, and (ii) carrier frequency/phase tracking. Typically, the receiver keeps a continuous tracking loop to follow the changes, and whenever the receiver loses a satellite, another acquisition is necessary. Tracking is a relatively inexpensive process, using feedback loops. So, once a GPS produces its first location fix, subsequent location estimates become fast. However, once the GPS stops tracking, the utility of previously known Doppler shifts and code phases diminish quickly. This is the reason that in vehicle navigation applications, the GPS receiver has to run continuously without duty cycling.

With correct tracking, the receiver can *decode* the packets sent by the SVs. In general, the receiver needs to decode SV ephemeris every 30 minutes (its valid time span) and time stamps every 6 seconds. Figure 4.2 shows a schematic view of the aforementioned modules of the GPS receiver. On the left side of the Figure, we can see the analog digital processing procedures, while, on the right side, we can observe the digital signal processing ones.

One important aspect of the GPS system is the time-of-first-fix (TTFF), i.e.,

the time elapsed until the GPS receiver unit can compute its position with a reasonable precision. Basically standalone GPS receivers can calculate its first position by a three different ways:

Cold Start. When the receiver has no prior knowledge of the satellites, it has to search the entire space. Usually, GPS receivers do not buffer the raw data to perform the search, rather they perform one code phase search per millisecond as the signal comes in. Although typically there are hardware correlators that perform acquisition in parallel, it still takes a few seconds to acquire one satellite. This is one of the main reasons for the slow initial position fix and high energy consumption for standalone GPS devices.

Warm Start. When the receiver has a previous lock to the satellites, it can start from the previous Doppler shift and code phases and search around them. In general, if the previous lock is less than 30 second old, a warm start can quickly find the new lock. Otherwise, the receiver has to revert to the cold start process.

Hot Start. When the previous satellite locks are within a second, the receiver can skip the acquisition process and start directly from tracking to refine the Doppler and code phases. In this mode, all information that the receiver needs is already in place.

Typically, GPS receivers take about 1 min, 30 s, and 6 s, for cold, warm, and hot start, respectively [van Diggelen, 2009].

4.2.3 Navigation equations

In order to estimate the pseudoranges by using the Eq. (4.1), the receiver needs a time reference to know in what time some portion of the signal was transmitted by the satellites. To do so, the time-of-week included in the field (HOW) is decoded from the navigation data. The satellite time-of-week represents the number of seconds passed since the last GPS week rollover (every Saturday midnight). Figure 4.3 shows the GPS navigational packet structure and its five sub-frames. Basically the navigation data is 1500 bits long, composed of five 300 bits long

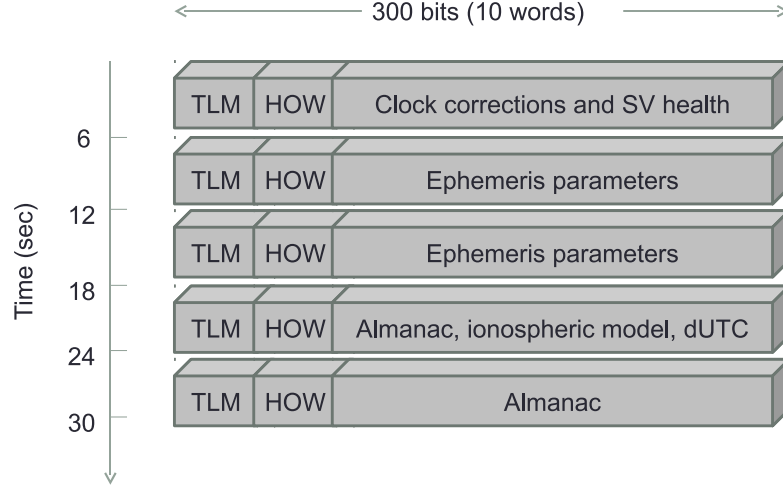


Figure 4.3: Navigational data: one frame composed by six sub-frames. Source: Borre et al. [2006]

sub-frames. It means that each sub-frame takes 6 s to be transmitted. The first field (TLM) includes the packet preamble and some parity information, the second field (HOW) includes more parity information and the time-of-week (TOW), while the payload includes satellite's information and the navigational data itself.

In other words, the receiver needs to decode the packet not only to get the navigational information but also to identify the packet borders that represent the time reference for the pseudorange calculation. Let's considered that the binary data collected by the receiver is composed by chunks of 1 ms (one C/A code length), thus, when the receiver detects a sub-frame border, it realizes the millisecond component of the transmission delay, while the sub-millisecond is determined by the code range alignment during the tracking phase. Thus, the transmission delay can be seen as [Sirola, 2001]:

$$\tau^{(k)} = \frac{1}{1000}(N^{(k)} + \varphi^{(k)}), \quad (4.2)$$

where $N^{(k)}$ is the millisecond component estimated by the receiver as the time of the detection of the sub-frame sent by the satellite k , and $\varphi^{(k)}$ is the sub-millisecond component that is estimated by the observed code delay into the

1 ms chunk. Thus, applying Eq. (4.2) in Eq. (4.1) we get:

$$\frac{c}{1000}(N^{(k)} + \varphi^{(k)}) = \|X_s^{(k)}(t - \tau^{(k)}) - X\| + b + \epsilon^{(k)}, \quad (4.3)$$

where c is the speed of the light. The left side of the Eq. (4.3) represents the observed pseudorange while the right side represents the calculated one.

As the Eq.(4.3) presents four unknowns (three for the receiver's position and one for the common bias), we need at least 4 satellites to have a well conditioned system. More often than not, receivers can see more than 4 satellites and we have an over-determined system of non-linear equations. The most usual way to solve this problem is by linearizing the equations and using the least-square method. To do so, we need to solve

$$\delta z = H\delta x + \epsilon, \quad (4.4)$$

where δx is the vector of updates to the a priori state (x, y, z , and b), $\delta z = z - \hat{z}$ is the vector of a priori measurements residual, being z the vector of measured pseudoranges and $\hat{z}$ the vector of predicted pseudoranges, ϵ is the unknown error (tropospheric model uncertain, and other stochastic errors), and

$$H_{k \times 4} = \begin{bmatrix} -e^1 & 1 \\ \vdots & \vdots \\ -e^k & 1 \end{bmatrix},$$

being e^k the three-dimensional unit vector that points from the predicted receiver's position to the satellite k . The matrix H is obtained after the linearization process of the Eq. (4.1). This process is out of the scope of this document and further discussion can be found in van Diggelen [2009]. The initial position is usually assumed to be the center of the earth, i.e., $X = 0, Y = 0$, and $Z = 0$, when a better approximation is missing.

Notice that once the satellites are acquired, distance measurements, and thus the location of the receiver, can be estimated every millisecond. A typical GPS receiver will average over multiple LS solutions to further reduce noise and improve location accuracy.

4.2.4 A-GPS

Assisted GPS (A-GPS) is an augmentation technique used to improve some functionalities of the GPS system [Sirola, 2001]. It relies on further existing network infrastructure to provide additional information to a GPS receiver. For instance, mobile phones can use data communication such as 3G, EDGE, WIFI or Bluetooth in order to get all necessary extra information. Notice that the A-GPS receiver makes measurements of the GPS satellites while grabs additional data from a network, thus, an A-GPS receiver still needs to process the satellite signals.

Basically, the A-GPS system provides the information that allows the A-GPS receiver to be aware of the set of expected visible satellite, the expected frequencies that should be swept, additional ephemerides and/or almanac data used to calculate the satellite's orbit, time synchronization, and initial position estimate. Provided by this assistance, A-GPS receivers can be able to calculate its TTFF quickly, typically in the order of some few seconds in contrast of 1 min from the standalone GPS receiver.

An A-GPS system can assist the receiver device by many different ways [van Diggelen, 2009]. For a cold start, it can provide all possible frequencies and code delays to narrow the acquisition search. In this case, it's still necessary to decode the TOW and ephemeris from the navigational data. For a warm start, the already stored almanac can be used in conjunction to a priori position and time in order to decrease the search space. It's also necessary to decode the TOW and ephemeris from the navigational data. For a hot start, a priori position and time can be used in the same way as in warm start. In this case, if a precise time is known, it's not necessary to decode the TOW and ephemeris.

A-GPS systems can be classified in two general approaches: (i) MS-Assisted, where the position is calculated at a server, and (ii) MS-based, where the position is calculated at the receiver, helped by network assistance.

In MS-Assisted approach, as the position is calculated at a server, the GPS receiver's job is simplified. It needs to acquire satellites' signal and send the measurements to the server. Hence, the receiver doesn't need any satellite ephemeris and/or almanac. The receiver can send its pseudoranges or raw data. If the receiver sends the pseudoranges, it needs to execute the acquisition algorithms

Table 4.1: Summary of A-GPS assistance for different type of starts. Source: van Diggelen [2009]

Type of Start	Acquisition	Ephemeris	Time of Week	TTFF
Standalone Cold	≈ 30 s	From satellite data ≈ 30 s	From ephemeris	≈ 1 min
Standalone Warm	≈ 1 s	From satellite data ≈ 30 s	From ephemeris	≈ 30 s
Standalone Hot	≈ 1 s	In memory	Accurate real-time clock, or decoded from HOW ≈ 6 s	seconds
Assisted Cold, Coarse-Time	≈ 1 s	From server	decoded from HOW ≈ 6 s	seconds
Assisted Cold, Fine-Time	≈ 1 s	From server	Assistance time	≈ 1 s

that can also be assisted by network data in order to narrow the search space. If fine-time assistance is available, the server can directly compute the expected code delays and send to receiver.

The MS-based approach is mostly used by hardware manufacture and the position is calculated at the receiver itself. In this approach, the receiver needs to have all the functions of a standalone GPS device and the assistance is performed to provide the expected Doppler frequencies and code delay, navigational data (almanac and/or ephemeris), coarse or fine time (for code delay assistance, fine-time is necessary), and an initial estimate of the receiver's position. As fine-time assistance, it's considered an accuracy better than 1 ms, otherwise, it's called coarse-time assistance. Table 4.1 is adapted from van Diggelen [2009] and shows the benefits of assistance by means of TTFF.

4.2.5 Coarse time navigation

Some authors such as Syrjarinne [2000]; Sirola [2001]; Akopian and Syrjarinne [2009] and van Diggelen [2009] describe different methods to estimate the millisecond part of the Eq. (4.3) without decoding the data packet. Adopting those

solutions, GPS devices are able to reduce its TTFF because the data sub-frame takes about 6 s to be transmitted. Here we will describe the last solution that we adopt in the rest of this work.

Coarse Time Navigation (CTN) is a particular A-GPS assistance that helps the receiver to decrease the TTFF. The basic idea is to provide a nearby landmark to avoid decoding the time stamps from the SV packets. Since light travels at 300 km/ms, two locations within 150 km of each other will have the same millisecond part of the propagation delay, rounded to the nearest integer. For mobile phones, it is natural and convenient to use cell tower locations as the landmarks. The landmarks can further help with acquisition by providing initial values for Doppler shifts and code phases.

If the GPS receiver has a coarse time synchronization, an already decoded ephemeris/almanac, and, a good initial position guess, it can estimate the initial pseudoranges by using the initial position and the ephemeris/almanac to calculate the satellite position. This process adds another source of error, namely the coarse time error, because the receiver does not have a precise time reference. An error on the receiver's time reference will lead to wrong satellite position estimation. This error, differently as the common bias effect, leads to a different amount of errors on the satellites' position. The coarse time error is described in van Diggelen [2009] and can be model as:

$$\hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)}(t_{tx}) = \hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)} - \hat{z}^{(k)}(\hat{t}_{tx} + \delta_{tc}) \quad (4.5)$$

$$= -\mathbf{v}^{(k)} \cdot \delta_{tc}, \quad (4.6)$$

where δ_{tc} is the update to the a priori coarse-time state, t_{tx} is the actual time of transmission, $\hat{t}_{tx}$ is the coarse-time estimate of t_{tx} , $\mathbf{v}^{(k)} = (e^k \cdot \mathbf{v}^{(k)} - \dot{\delta}_t^{(k)})$ is the pseudorange rate, $e^{(k)}$ is the unit vector as described before, $\mathbf{v}^{(k)}$ is satellite velocity vector, and $\dot{\delta}_t^{(k)}$ is the satellite-clock error rate. The two last can be obtained from the ephemeris. The Eq. (4.6) can be included in the navigation equations and the least-square equations aforementioned will present one more state, the coarse-time error. Thus, the state update vector is $\delta x = [\delta_x \ \delta_y \ \delta_z \ \delta_b \ \delta_{tc}]'$,

and the matrix H is

$$H_{k \times 5} = \begin{bmatrix} -e^1 & 1 & \nu^{(1)} \\ \vdots & \vdots & \vdots \\ -e^k & 1 & \nu^{(k)} \end{bmatrix},$$

all terms were described before.

Even though the millisecond part of the pseudorange can be estimated by the initial position and the coarse-time, any small error on this estimate can lead to a very inaccurate position calculation. For instance, an error of 1 ms produces results 300 km away from the actual location. Thus, to avoid problems with integer rollover, that typically incurs this amount of error, van Diggelen [2009] propose the following method to reconstruct the full pseudoranges. Let the transmission delay be modeled by:

$$N^{(k)} + \varphi^{(k)} = r^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)} \quad (4.7)$$

$$= \hat{r}^{(k)} - d^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)} \quad (4.8)$$

where $r^{(k)}$ is the actual geometric range from the satellite k , $\delta_t^{(k)}$ is the satellite clock errors obtained from the ephemeris at the a priori coarse time for the satellite k . As $r^{(k)}$ is unknown, we can substitute $r^{(k)} = \hat{r}^{(k)} - d^{(k)}$ where $\hat{r}^{(k)}$ is the estimated pseudorange from the a priori position at the coarse time of transmission and $d^{(k)}$ is the error in $\hat{r}^{(k)}$. The method involves the choosing of a reference satellite, $k = 0$, where $N^{(0)} = \text{round}(\hat{r}^{(0)} - \varphi^{(0)})$ is the millisecond part of the pseudorange of the reference satellite¹. This value is used in order to reconstruct the millisecond pseudoranges for all other satellites relatively to the reference satellite. Thus, if we subtract the Eq. (4.8) from the reference satellite full pseudorange we get:

$$\begin{aligned} N^{(k)} &= N^{(0)} + \varphi^{(0)} - \varphi^{(k)} + (\hat{r}^{(k)} - d^{(k)} - \delta_t^{(k)} + b + \epsilon^{(k)}) \\ &\quad - (\hat{r}^{(0)} - d^{(0)} - \delta_t^{(0)} + b + \epsilon^{(0)}). \end{aligned}$$

¹van Diggelen [2009] recommends the use of the highest satellite in view as reference, and provides a good reason for that

We still don't know the values of $d^{(0)}$ and $d^{(k)}$, but considering that we have a initial position and coarse time close to the correct values (100 km and 1 min), the order of magnitude of $(-d^{(k)} + \epsilon^{(k)} + d^{(0)} - \epsilon^{(0)})$ is less than about 150 km, van Diggelen [2009] shows that we can correctly estimate $N^{(k)}$ by:

$$N^{(k)} = \text{round} \left(N^{(0)} + \varphi^{(0)} - \varphi^{(k)} + (\hat{r}^{(k)} - \delta_t^{(k)}) - (\hat{r}^{(0)} - \delta_t^{(0)}) \right). \quad (4.9)$$

This method avoids the undesirable integer roll over problem that leads up 300 km of error for each wrong estimate.

By adopting this solution, the full pseudorange is estimated by the CTN approach as the millisecond part, and the sub-millisecond is obtained by the code phase estimate from the raw signal. CTN can be computed either on the device or on a back-end server. In order to compute it on a device, the device must update its ephemeris data every 30 minutes, and must have a database of the precise locations of a set of landmarks. Alternatively, since the only device-dependent data are the code phases for each visible SV, the location computation can be moved completely to the cloud. We explored the CTN technique in Ramos et al. [2011b] in order to devise a energy-efficient MS-Assisted GPS solution for Internet capable devices where the receiver sends only raw GPS data to the cloud, and the position is computed in the back-end server (see details in Section 4.3).

4.2.6 GPS energy

In this Section, we describe how GPS receivers drains the battery when they are active. We use a HTC desire smart-phone running Android 2.1 to trace a power/energy profile.

Observing the receiver's building blocks shown in Figure 4.2, we can empirically derive the following energy model:

$$E_{GPS}(t) = \epsilon_{on} + \epsilon_{analog}(t) + \epsilon_{digital}(t) + \epsilon_{off},$$

where ϵ_{on} and ϵ_{off} represent the energy required to turn the GPS circuitry on and off, respectively; ϵ_{analog} represents the energy required to perform all the analog signal processing on the receiver, such as bandpass filter, amplifiers, frequency

down conversion, and A/D conversion; $\varepsilon_{digital}$ represents the energy drained by the digital signal processing such as correlators, filters, demodulators, least-square calculation, i.e., all tasks required to perform the acquisition, tracking, data demodulation, and position calculation. Observe that ε_{analog} and $\varepsilon_{digital}$ depend on the time that the system dwells on each phase, for example the acquisition phase can be performed quickly with network assistance, and thus, drain less energy. The terms ε_{on} and ε_{off} depend only on a fixed amount of energy drained during those phases.

The digital signal processing can be detailed as:

$$E_{Digital}(t) = \varepsilon_{acquisition}(t) + \varepsilon_{tracking}(t) + \varepsilon_{position}(t).$$

We've performed two experiments in order to trace a GPS power profile. In the first experiment we measure the power consumption when the GPS was trying to acquire satellites in an indoor environment. Thus, the tracking and position calculation were avoided and only the component $\varepsilon_{acquisition}(t)$ of the digital signal processing was measured. Obviously other components such as ε_{on} , ε_{analog} and ε_{off} are present but, in this work, we are only interested on the digital side. So, for our solution we are interesting to know the differences on the power consumption when the GPS device performs the acquisition, tracking and position calculation. The second experiment aims at empirically estimating the power consumption when the GPS is used in outdoor environments. In this last case, all phases are performed.

Figure 4.4 shows a power profile of the GPS device from the smart-phone HTC desire when trying to get a position in indoor environments. Note that we can identify some peaks when the GPS is trying to acquire satellites. Those peaks represent about 400 mw of power consumption. Figure 4.5 shows a power profile of the GPS device from the same smart-phone when it tries to get a position in outdoor environments. Note that now we are able to identify all the phases, being the acquisition phase according with the first experiment, i.e., about 400 mw, the tracking phase about 700 mw and the position calculation about 1200 mw. For this experiment we query the position many times to force the separation of the phases, thus, we can see many short peaks when the position is being calculated.

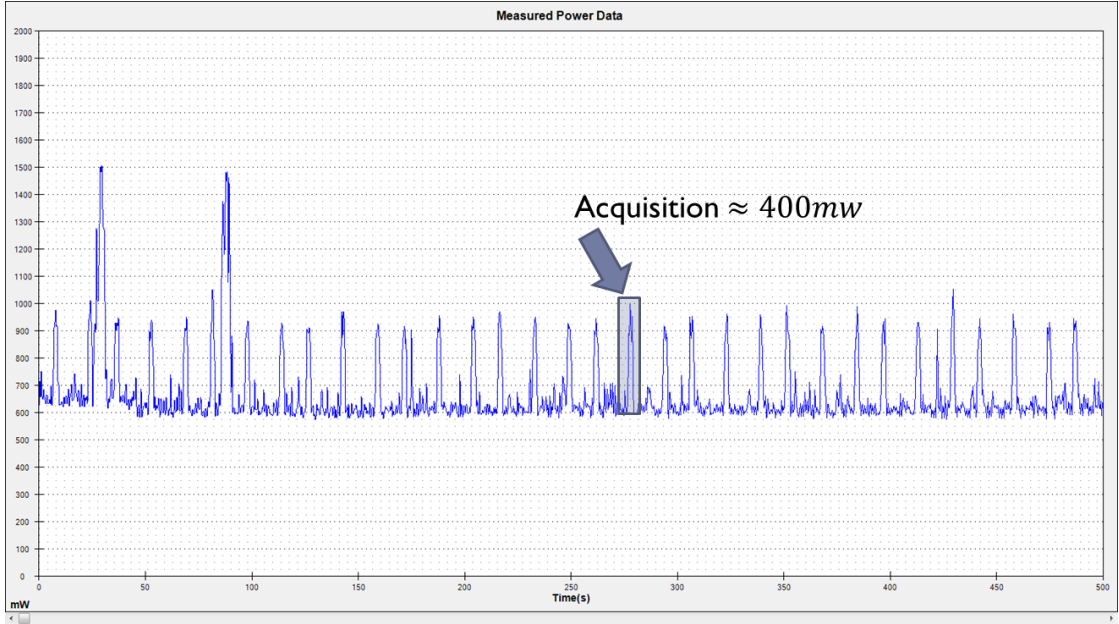


Figure 4.4: Instantaneous power consumption for acquisition phase

We assume that those high peaks represent the least-square calculation on the smart-phone's main processor.

The power profile presented in this Section is used in order to illustrate a baseline experiment that evaluates the energy savings of using our proposed solution, detailed in the next Section, when compared to regular A-GPS devices. This analysis is shown in Section 4.3.6.

4.3 Our proposal

The new trend of mobile applications brings new challenges for localization techniques. As mobile devices are typically equipped with tiny devices and sensors, the localization service should be flexible to properly work in devices with severe energy, processing and storage constraints. Thus, as discussed in above sections, standalone GPS, and conventional A-GPS devices can barely be used in those scenarios. Cloud-offloaded GPS (CO-GPS) comes as an alternative to meet those strict requirements.

CTN is a known GPS technique, which have received little attention [Ramos

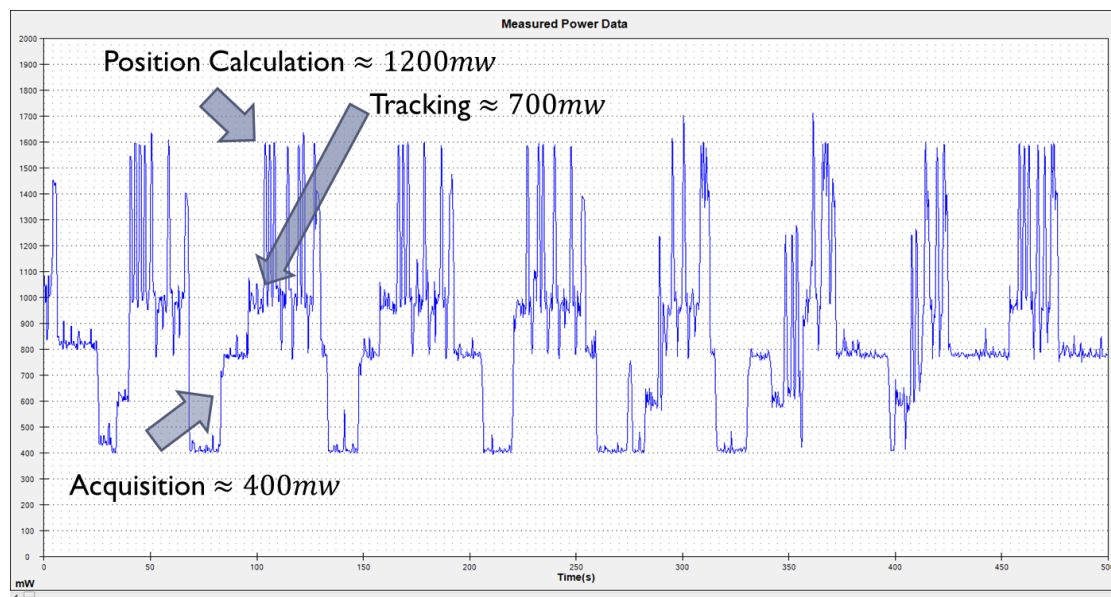


Figure 4.5: Instantaneous power consumption for acquisition, tracking and position calculation phases

et al., 2011b] mainly because it does not necessarily improve TTFF, which has so far been the primary optimization goal for mobile GPS design, however, it provides the best foundation for partitioning GPS receiving stages between the device and the cloud. Moreover, standalone GPS does not have a communication link to the cloud, thus, CO-GPS design takes advantage of the CTN technique in order to properly partitioning the GPS receiving signal process pipeline, by offloading most of the demand of processing to the cloud. The idea is to take advantage of the Internet connectivity that those mobile devices are usually capable. Even devices that do not have continuous connectivity, the data can be stored for batch upload, if the application does not require immediate use of the position information (delay-tolerant position information).

A CO-GPS device should implement only the analog signal processing part shown in Figure 4.2, while all the CPU demanding digital processing can be moved to the cloud. Additionally, the cloud is also responsible for estimating the full pseudorange from the raw signal, the time stamp, and the CTN technique. Thus, a CO-GPS service allows a low cost and low energy localization service for devices that do not require real-time localization.

The rationale behind the CO-GPS technique is to collect few milliseconds of the raw GPS signal and store it along with a time stamp. As we know from the previous discussion, the C/A code repeats every millisecond. So, in theory, one millisecond of raw signal contains all the information needed to acquire the satellite and compute the code phases. The device, opportunistically sends this data, and additional data required by the application, to the cloud. Next, two different approaches can be used: (i) if there is a known nearby landmark, the CTN technique can be used as it, on the server-side, to calculate the receiver's position. We refer to this approach as Low Energy Assisted Positioning (LEAP) [Ramos et al., 2011b]. And (ii) if there is no landmark information, the cloud generates a number of candidate landmarks, and then, performs multiple hypothesis test and uses other geographical constraints to filter out wrong solutions. This is a generalization of the former proposal, and we refer to it as CO-GPS [Liu et al., 2012].

For the cases that the device does not have a reference to a nearby landmark, the millisecond part of the propagation delay is not specified, which causes ambiguity in the solution. To illustrate this effect, we take a real GPS trace and apply CTN with an array of landmarks across the globe. There are 7 satellites in view. The landmarks are generated by dividing the latitude and longitude with a 1° resolution. Thus, we picked $180 \times 360 = 64800$ landmarks. Figure 4.6a shows the total of 84 converged points. We call the false solutions as *shadow locations*. The key challenge of the CO-GPS with no a priori location is to rule out shadow locations and find the actual location.

The first step in eliminating shadow locations is to reduce the number of possible landmark guesses. If past locations of the sensor are available and the receiver has not moved more than 150 km between the samples, then we can use them as the landmark. However, in the bootstrap process, or when the time difference between readings is large enough to allow movement greater than 150 km, we have to assume no prior knowledge of the location of the sensor.

Because a light-ms is 300 km, the elevation of a shadow location is likely to be far away from the Earth's surface. For example, Figure 4.6b shows the number of possible solutions when we limit the elevation to be within the $[-500, 8000]$ meters range.

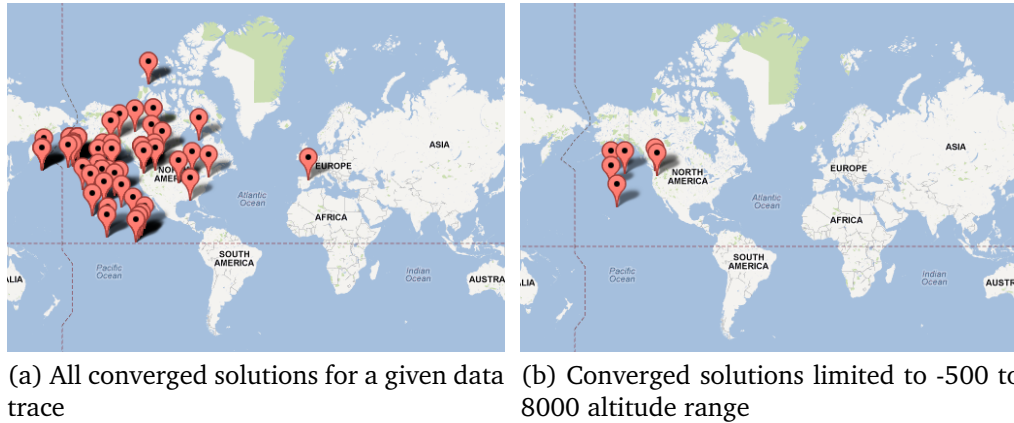


Figure 4.6: Solution ambiguity

As we could observe, absolute elevation by itself does not yield a unique solution. However, the true elevation of the Earth’s surface is known on the web. For example, the United States Geological Survey (USGS) maintains a service that returns the actual elevation at any given latitude/longitude coordinate. It is almost impossible to have two nearby locations (a few hundred km apart) such that both elevations are correct. We validate empirically that using an elevation map can always yield the correct solution.

4.3.1 Web services

Putting everything together, the CO-GPS backend web service must perform the following steps, as shown in Figure 4.7.

In the CO-GPS web service, an agent runs periodically to fetch the ephemeris data and record them into the service’s storage so they can be used by other service components. It uses the National Geospatial-Intelligence Agency’s precise ephemeris service², which contains both position and velocity information for each satellite.

Our core GPS processing engine is based on SoftGNSS [Borre et al., 2006], which is implemented in Matlab. When a raw signal chunk is sent to the service together with a time stamp, it is first replicated and patched to construct a signal

²<http://earth-info.nga.mil/GandG/sathtml/ephemeris.html>

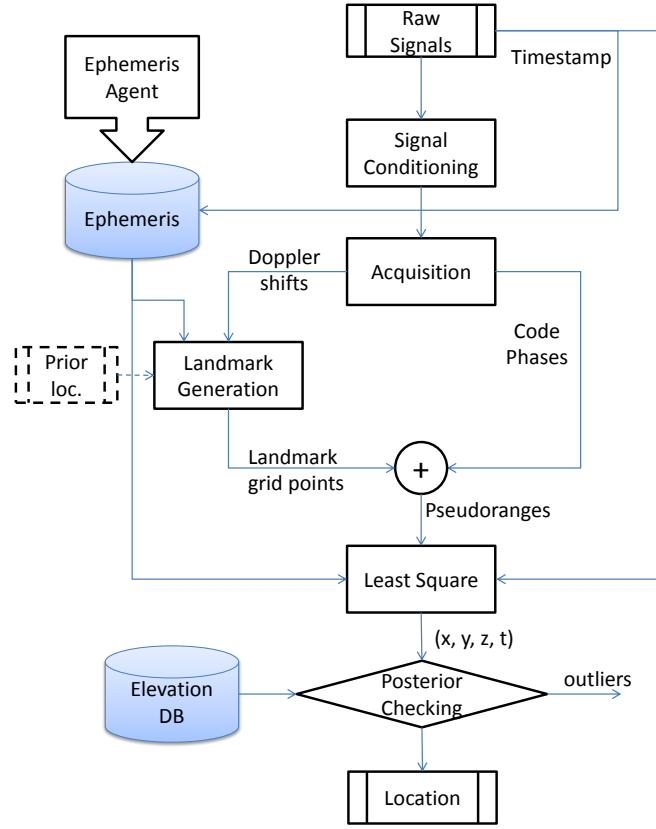


Figure 4.7: The flow of CO-GPS backend web service.

of proper length. For example, a 10 ms signal is required for a typical acquisition process. If we receive a 2 ms signal chunk, then the signal is replicated 5 times to fit in the processing pipeline.

After acquisition, the Doppler shifts and code phases are extracted for each visible satellite. The Doppler shifts are used with ephemeris to infer the initial landmarks, as discussed in Section 4.3. If the location of the device is known in the near past or is provided by the system (LEAP mode), then that information is used as a landmark. Once the set of landmarks is obtained, we can use Equation (4.9) to calculate the NMS and apply the CTN method in parallel to process each landmark and solve for its location. Finally, for each converged result, the USGS elevation database is checked, and any result that is outside the

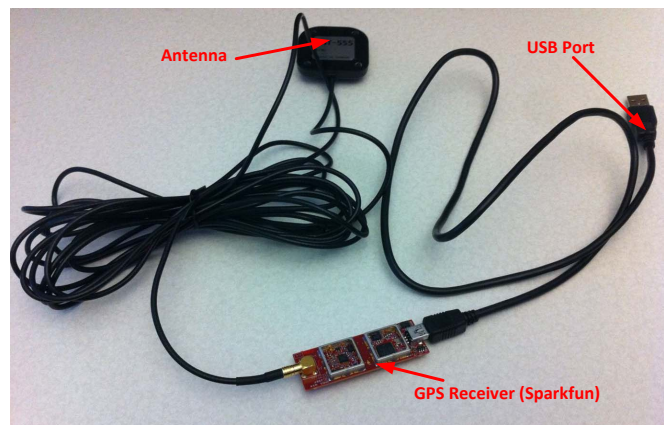


Figure 4.8: The GSP data collector used for experiments

error bound is discarded.

4.3.2 Evaluation

We evaluate the quality and limitations of the CO-GPS approach using raw GPS samples and a version of Soft-GNSS [Borre et al., 2006] that we have modified to simulate receiver duty-cycling and computation offloading.

This evaluation uses about 100 sets of raw GPS data taken from six different locations in both the northern and southern hemispheres of Earth. We used a SiGe GN3S v3 sampler dongle (available from Sparkfun Electronics³), as shown in Figure 4.8, which gives us the flexibility of varying the sample length for evaluation. Our data set contains the baseband GPS signal sampled between 10 to 60 seconds. For ephemeris data, we use the Ultra-Rapid Orbits available on the International GNSS Service (IGS) website⁴.

We perform the cloud-side computation on a server with a quad-core Intel Xeon 3520 @ 2.66 GHz and 6 GB memory. We observe that the acquisition process takes about 3 seconds to finish. Variations in signal length have minor effects on the execution time. Once the code phases are obtained, CTN takes less than 300 ms to calculate the actual location.

³<http://www.sparkfun.com/products/10981>

⁴<http://igscb.jpl.nasa.gov/>

To acquire location ground truth for these samples, we apply unmodified Soft-GNSS to calculate the receiver's position from the traces. This method achieves Dilution of Precision (DOP) values below 6, which is considered at least good. The produced results have expected errors less than 20 m.

When considering the values presented here it is important to note just how different the CO-GPS approach is from a standalone GPS implementation when the same signal trace is used. In addition to regular GPS error sources, CO-GPS adds the following possible sources of error: (i) the position is calculated by using code phase samples that are closer to each other than in regular GPS; therefore they are more likely to suffer from transient noise that spans multiple samples, (ii) CO-GPS does not use the lock loops (PLL and DLL) that are implemented in the tracking steps in regular GPS; only the code phase and Doppler frequency estimated in the acquisition step are used, which may contribute to less accurate results, and (iii) the use of CTN technique adds additional potential error, especially when the number of satellites is low.

4.3.3 Acquisition quality

Since the goal of CO-GPS is to achieve the best possible energy efficiency in GPS sensing, we first evaluated how much data to use and the best duty-cycle strategy to employ to determine an appropriate trade-off between accuracy and low energy use. One of the key parameters that improves single location calculation is the number of satellites that can be acquired. So, we first vary the parameters to check the acquisition quality.

Table 4.2 shows the three scenarios we evaluated to determine the appropriate amount of data to use to estimate the receiver's position. Each row is a different combination of the number of chunks, the chunk duration in milliseconds, and the time gap (or sleep period) between each chunk. As illustrated in Figure 4.9, the chunk and gap parameters define the duty cycling of the receiver when sensing one location. In order to improve acquisition quality, at every location, one may collect multiple chunks of data with some gaps in between.

In theory, one millisecond of signal is enough to estimate a pseudorange for each SV, and therefore infer the receiver's position. However, due to local clock

Table 4.2: Scenarios of evaluation for CO-GPS

	# of chunks	chunk length (ms)	gap length (ms)
1	1	{2, 4, 6, 8, 10}	0
2	{1, 2, 3, 4, 5}	2	0
3	5	2	{0, 10, 50, 100}

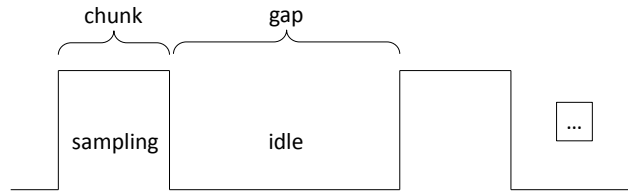


Figure 4.9: Duty cycling in experimental evaluation. After an idle period (called a gap), the receiver collects a chunk of raw data.

drift, 12 ms of data can be challenging for boundary alignment, so we extract a set of chunks of at least 2 ms duration and average the position outcomes derived from them.

Figures 4.10 and 4.11 show the influence of these parameters on the number of visible satellites detected and in the results in terms of absolute error, respectively. Since our goal is to evaluate acquisition results, when multiple chunks are used, we select the chunk of best quality (in terms of having the maximum number of satellites) for location calculation. We evaluate averaging multiple locations in the next set of experiments.

We can observe in Figure 4.10a that increasing the chunk length from 2 ms to 10 ms does not yield an increase in the number of satellites in view. In Figure 4.11a we also can observe that the chunk length does not influence the error. Therefore, increasing the GPS duty-time slot contiguously up to 10 ms does not improve the location results. This is because 10 ms is a short period of time and the quality of the GPS signals does not change significantly during it. Since increasing the chunk length does not improve our location results, we can instead duty-cycle the GPS circuitry to save energy, by intercalating sleeping intervals

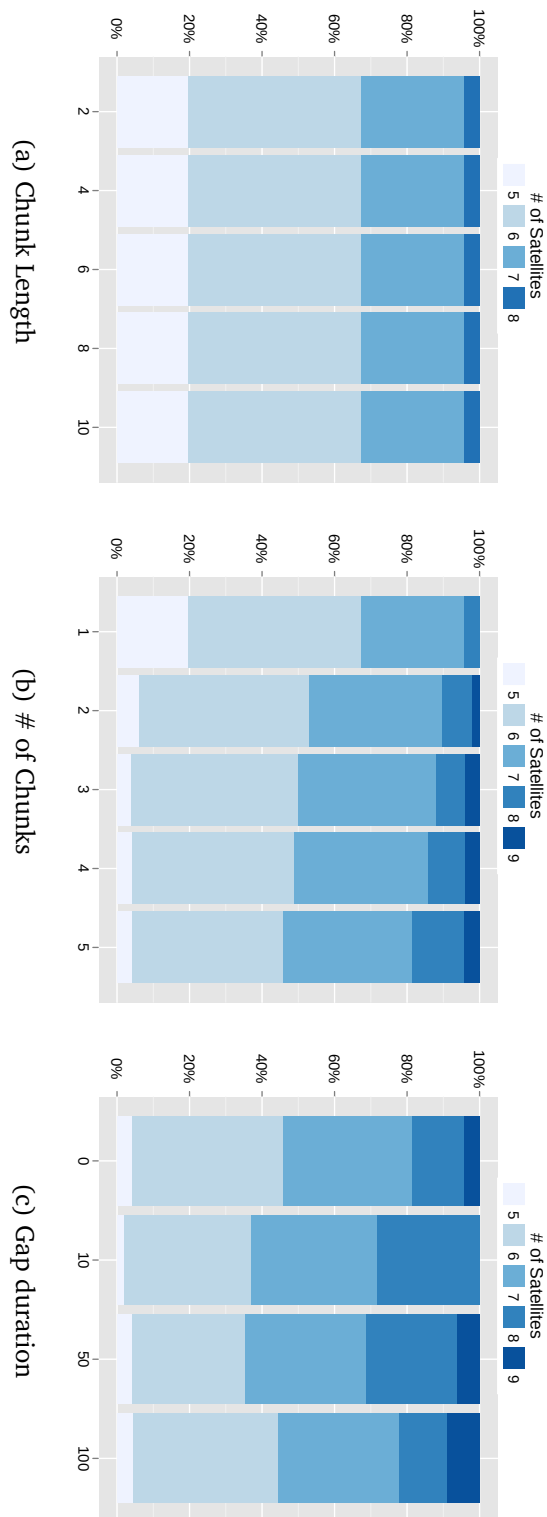


Figure 4.10: The number of acquired satellites in various experiment settings.

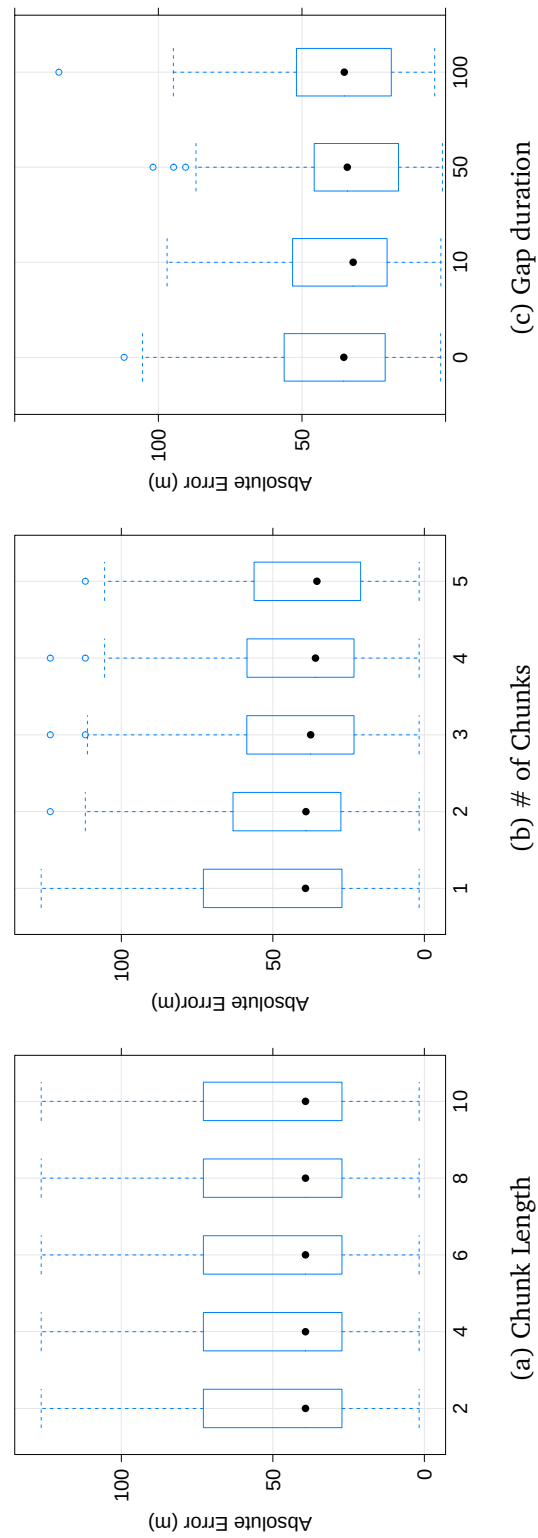


Figure 4.11: Location error distribution in various experiment settings when single chunk is used for location calculation. In (b), we select the “best” chunk according to the number of satellites in view.

between each data collection interval.

Figure 4.10b shows that the number of visible satellites detected can be improved if we process multiple chunks of data. Observe that the acquisition results using a contiguous 10 ms chunk of data are different from the results when we split this data into 5 chunks of 2 ms. The acquisition algorithm considers a satellite to be visible when it is stable along the entire signal, so it is more likely to recognize transient satellites in separate chunks of 2 ms than in a single 10 ms chunk. These transient satellites may help improve the position calculation. Thus, we can select only the chunk that yields the highest number of satellites in view and use this chunk to estimate the receiver's position. In Figure 4.11b we can observe that when only the best chunk is used among multiple chunks, the location accuracy only improved slightly. In fact, the average error barely improves, while the variance is lower with more chunks collected.

Figure 4.10c shows that we are also able to increase the number of satellites in view when we separate the sampling intervals by intercalating some sleep time (the gap duration). We can observe that for this parameter, the number of satellites in view generally increases as gap length increases, but not monotonically. This is because the signal can change over time in ways that are not always directly related to the gap duration. For example, a moving receiver may be blocked briefly by trees, buildings, bridges, or tunnels. Atmospheric conditions may also change slightly, and for large gap intervals, the satellite and/or receiver movement can result in a different satellite arrangement relative to the receiver. Thus, we observe that 50 ms seems to be a reasonable gap value as the satellite's and receiver's movement can be considered negligible, and the obstacles and shadowing are not likely to change significantly. We also can observe in Figure 4.11c that the error variance slightly decreases accordingly.

4.3.4 Location accuracy

Figure 4.12 presents graphs of satellite visibility and error metrics. Figure 4.12a shows the number of visible satellites recognized when each location request contained 5 chunks of 2 ms with a 50 ms gap between them. To improve the location accuracy, we use each chunk independently and then average the result

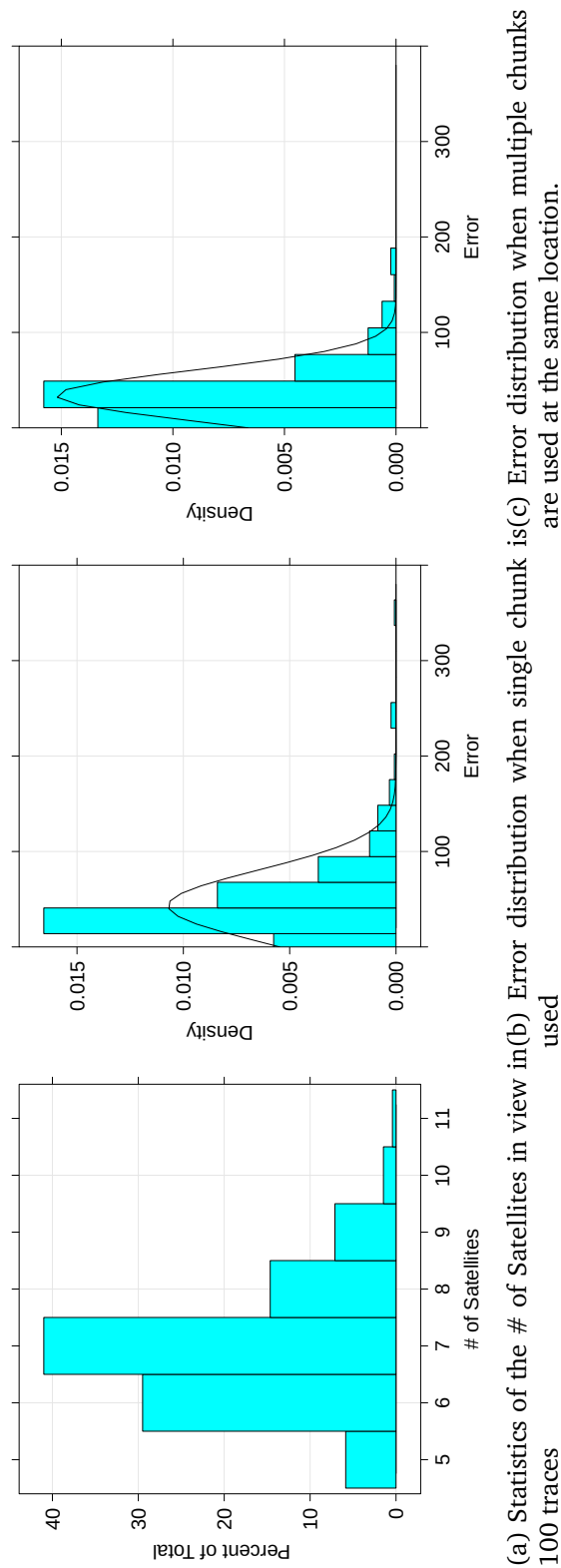


Figure 4.12: Overall location accuracy distribution.

Table 4.3: Error statistics

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Single Chunk	0.96	19.00	32.45	43.16	56.79	349.70
Multiple Chunk	0.45	16.56	26.67	33.77	42.17	181.10

location to obtain the final location.

For our data, the receiver was more likely to have between 6 and 8 satellites in view, with 7 being the most frequent count obtained. As CO-GPS uses CTN, at least five satellites are required, and the accuracy is greatly improved when more satellites are available. Figure 4.12b shows the error p.d.f. estimate when we use the single chunk approach, i.e., when we process only the chunk that yields the largest number of visible satellites and discard the other 4 chunks. In contrast, Figure 4.12c shows the error p.d.f. estimate when we use the multiple chunk approach (averaging the results of all 5 chunks). We can observe that, in general, the error is smaller when we adopt the multiple chunk approach. This happens because we are using more information to calculate the position while the samples are independent. We can observe that when we use multiple chunks, the results have a smaller variance and thus are more accurate. Table 4.3 presents some statistics of the absolute error corresponding to the single and multiple chunk approaches. Observe that the mean error is about 20% smaller when the multiple chunk approach is adopted.

Figure 4.13 illustrates some outcomes of CO-GPS's location estimation for the 6 locations we evaluated. We plot a circle of radius 100 m around the ground truth, represented by a push pin, to give the sense of a block level accuracy (accurate to within a city block). As we can see from Table 4.3, CO-GPS can achieve about 40 m location accuracy, in average, with 10 ms of data.

4.3.5 Time accuracy

Two pieces of information are required in order to obtain good results when CTN navigation is adopted: the initial position, and the time stamp corresponding to the moment that the GPS signal was collected. In order to fine-tune CO-GPS's

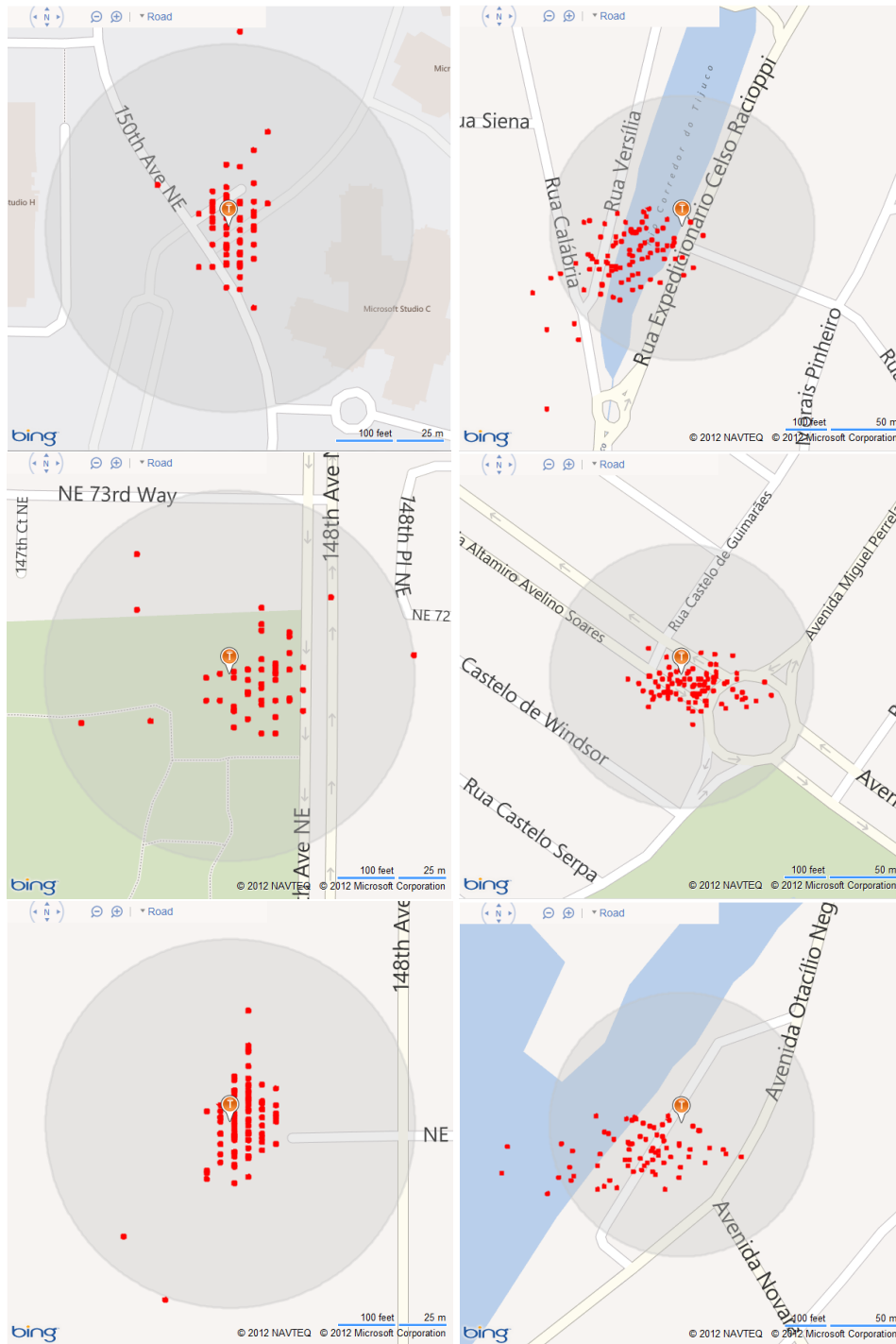


Figure 4.13: Overall results from 6 locations. The shadow is 100m in diameter. We see that there are bias errors in some cases.

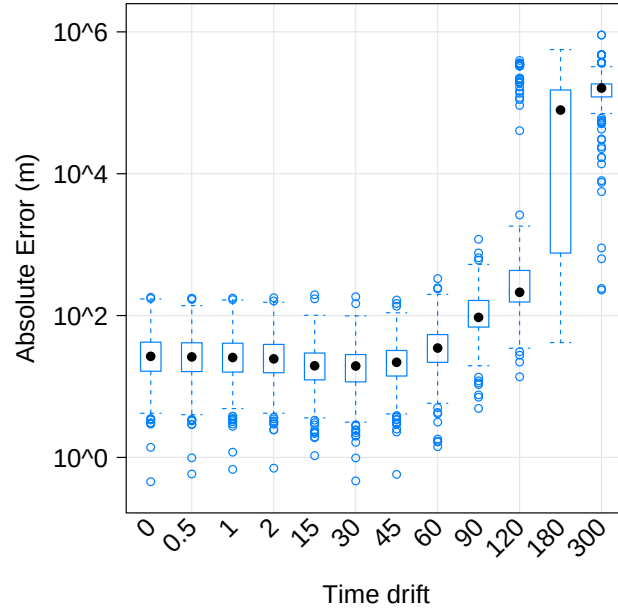


Figure 4.14: Error due to time drift

time synchronization mechanism, we evaluated how the error changes as the time drift increases. This is a key parameter that influences how tight the time synchronization must be on a CO-GPS implementation.

Figure 4.14 shows the error boxplots when the time drift increases from 0 up to 300 s. Observe that the error does not change significantly when the time drift varies from 0 to 60 s. After that, the error increases sharply, eventually reaching 10^6 m. This is due to the fact that under bad initial conditions, CTN navigation is not able to estimate the pseudorange millisecond part properly. Thus, as light travels about $3 \cdot 10^5$ m/ms, errors of the same order of magnitude are expected due to integer rollover. This synchronization requirement can be easily achieved by many different common time synchronization approaches such as the National Institute of Standards and Technology (NIST) longwave standard time signal WWVB [National Institute of Standards and Technology, 2011], Tiny-sync [Yoon et al., 2007] and Network Time Protocol (NTP) [David L. Mills, 2006].

4.3.6 Energy consumption

The energy savings offered by using the CO-GPS mode in practical usage scenarios are evaluated next using real user location traces. The location traces are collected by asking volunteers to run a data collection application on their mobile devices. This application continuously samples location, stores them locally, and uploads the trace once a day to our server. The trace data set consists of 18 unique users from US, Europe, and Africa, with 1 to 20 days of data per user. Each user's trace has different periods of satellite visibility depending on where they live, work, and travel. Consider two types of scenarios:

Trajectory Tracking: For this scenario, the user location is tracked continuously throughout the day at a few minutes granularity (once a minute in our runs) at moderate accuracy (40 m). The data is uploaded when the user has wired power, so communication energy is ignored. Such usage is representative to applications that mine user behavior over time, or collect movement patterns over large populations (e.g. skyhook⁵). The location data is not required instantaneously.

Local Social: In this scenarios, location is tracked continuously every few seconds (6 s in our runs) at high accuracy and uploaded to a cloud service every 5 minutes. Such usage represents social networking applications to find friends meeting at a public place, advertisement applications to track user movement and suggest currently relevant local services and/or products, and services that require recent motion trajectory. In this type of scenarios, the communication energy of uploading location is important.

It is tempting to compare CO-GPS with WiFi-based location services. If we consider proper duty cycling, the amortized energy expense is about the same for the two approaches without hardware optimization for CO-GPS. And both require communication to the server for location resolution. However, not all places have WiFi coverage. In our dataset, on average 25% locations do not have WiFi in sight, especially when driving on highways. In addition, WiFi services typically provide 30 ~ 100m location accuracy, while CO-GPS can provide better

⁵<http://www.skyhookwireless.com/>

accuracy $10 \sim 40\text{m}$ with same energy expense. So we focusing on comparing CO-GPS with standard GPS solutions in this section.

We perform the evaluation based on trace-driven simulation. We assume that location tracking is turned off when the device does not have GPS visibility, such as indoors, and an automated mechanism using a low energy sensor, accelerometer [Zhuang et al., 2010], is available to resume GPS use when user movement is detected. This implies that savings from CO-GPS increase only when GPS satellite visibility is available, making our comparison disadvantageous for CO-GPS since savings for other parts of the day are suppressed.

There is no GPS receiver on the market that is optimized for CO-GPS. For example, in the ideal case, when the GPS receiver is not involved in decoding the satellite packets, corresponding parts should be turned off. To evaluate energy benefits, we take a loose assumption that CO-GPS uses the same power as a regular GPS receiver when it is active. We assume the same GPS hardware for both CO-GPS and without CO-GPS, using an average draw of 400 mW (see Section 4.2.6) when powered on (brief impulses of higher power are observed for non-CO-GPS platform for certain calculations such as least squares phase, and are ignored, making the comparison disadvantageous for CO-GPS). So the main energy differences come from duty cycling, as we show in Section 4.3.2.

To make the simulation evaluation close to reality, we also measured the energy use for accelerometer sampling (259 mJ for 1s of sampling) for motion detection, and location trace upload (10.1 J for 5 minutes of location trace) relevant to the second scenario.

Figure 4.15 shows the energy savings provided by CO-GPS. Since different users have different number of days worth of data and the absolute energy usage varies widely, we normalize the data to the energy usage on non-CO-GPS mode and plot only the percentage savings. Savings are significant for many users. The users for which savings are small are mostly ones for whom GPS visibility was available only for small time windows during the entire day and the energy use is dominated primarily by the accelerometer sampling. The actual energy use is very different at 60 s sampling and 6 s sampling with upload but the percentage savings are similar. This happens because the communication energy of an upload every 5 minutes is negligible compared to GPS sampling every 6 s.

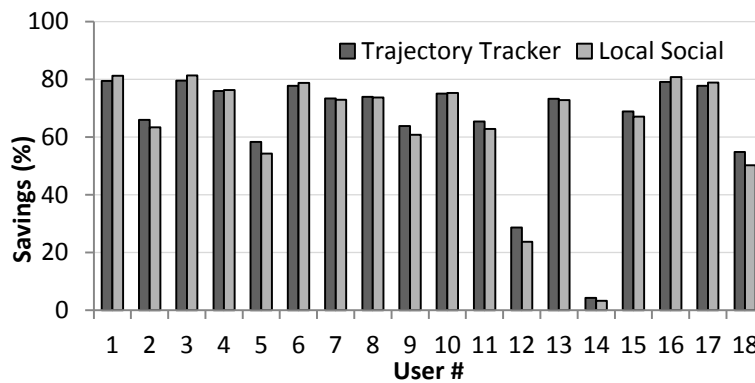


Figure 4.15: Energy savings from CO-GPS mode in two representative scenarios. Users #12 and #14 experience less energy saving than others mainly because they have little or almost no GPS visibility during the experimental periods.

4.4 Related work

Bearing in mind the problem we are interested in, current solutions presented in the literature tend to explore two different approaches, as following.

Some authors have focused on the devising of novel signal processing and positioning techniques mainly to increase the GPS receiver's accuracy and TTFF. For instance, Sun et al. [2005] present a comprehensive survey on signal processing techniques used in AGPS systems. Akopian and Syrjarinne [2009]; Sirola [2001] and van Diggelen [2009] show different approaches on how to model the GPS positioning equations and how to solve them without decode any portion of data when some assistance is provided. Those solutions are useful when the receiver can use assisted data from network, thus, they can decrease the TTTF and allow the use of GPS devices even when data decoding is not possible such as in presence of noise, shadow and incomplete data.

By different approaches, some authors try to establish a smart use of GPS by changing its duty-cycle and using different types of sensors available in current smart-phones. For example, Lin et al. [2010] propose a smart mechanism, namely A-Loc, which automatically determines the dynamic accuracy requirement for mobile search applications. A-Loc chooses the sensor that saves energy while attend the application's accuracy requirements. This approach relies on the idea that location applications do not always need the highest available accuracy.

To do so, they use a Bayesian estimation framework to model the location and sensor errors, and thus, choose the most adequate sensor.

Paek et al. [2010] propose a rate adaptive GPS-based positioning for smartphones. They argue that, due the so-called urban canyons problem, even GPS sensor is not able to provide a high accuracy position, thus, the device should be turned off. In those situations, alternatively, they use other location mechanism such as accelerometer, space-time history, cell tower blacklisting and bluetooth to decrease the active time of GPS devices.

Zhuang et al. [2010] present an adaptive location-sensing framework to reduce the use of GPS device in various scenarios. Their framework is based on four design principles: (i) substitution, where alternative location-sensing mechanisms are used instead of GPS device; (ii) suppression, where the information provided by other sensor can suppress the use of the GPS device, for instance, if the accelerometer can identify that the user is not moving, the GPS can be turned off; (iii) piggybacking, that synchronizes location request from different applications, and (iv) adaptation that adjust sensing parameters when battery level is low.

Differently from the current proposals as those herein discussed, we are not trying to avoid the use of GPS sensor once it's the more accurate known sensor to provide location service. In this work, we strive to provide a low-power GPS-based location service as described before.

4.5 Chapter remarks

This Chapter presented a study on GPS theory, proposed a new low-energy GPS system, and showed some experiment results to test the proposed solution. We used a software-based GPS lab kit in order to assess our proposal and we got promising results. We show that GPS devices, with adequate assistance, can reduce the time to get the position from 6 – 10 s, typically, to some milliseconds (about 10 ms). This approach leads to a large amount of energy saving once we only need to collect some milliseconds of the signal, and thus, the radio needs to be on only during this short period, leading to a very low duty-cycle.

Even though our solution leads to additional position error, we argue that it is affordable for a wide variety of applications. For instance, our solution can be applied to continuously sense the user's position and provide valuable location information to mobile location-based applications such as mobile search, geo photo tagging, and weather forecast, among others. Those applications require city block level precision only, and it is the precision is easily achieved by CO-GPS solution. In case of applications that require more accurate results, our solution can be easily adapted to collect more data (changing the duty-cycle) and maintain similar precision as standard GPS devices that use 30 s, or current AGPS devices that use about 6 – 10 s.

Cooperative target tracking in vehicular ad hoc wireless networks

“Anyone who lives within their means suffers from a lack of imagination”

Oscar Wilde

Target tracking plays a key role for vehicular ad hoc networks (VANETs) due to the fact that a wide variety of envisioned applications rely on the ability of this technique of detecting, localizing and tracking objects surrounding a vehicle. Tracking the location of a set of vehicles of interest, we may be able to manage the highly dynamic topologies present in this kind of networks. This subject has been studied in fields such as airborne traffic, computer vision, and wireless sensor networks. A VANET brings out new challenges that should be addressed. For instance, the cluttered and dense scenarios, communication issues such as short term links, and the variety of objects considered to be targets are some of the new ingredients to be taken into account. Applications such as collision warning/avoidance systems require strict time constraints, while others impose only mild restrictions. This complex and heterogeneous environment is discussed in this work, where we didactically divide the main problems into four components: the targets’ motion model, measurement models, data association problem, and

filtering. We also discuss the communication issues and how they affect these systems.

5.1 Introduction

Localization and tracking systems have been identified as key issues for Intelligent Transportation Systems (ITS) and Vehicle Ad Hoc Networks (VANETs) [Boukerche et al., 2008; Yousefi et al., 2006]. A wide variety of interesting location-aware applications have been proposed for ITS and with the advent of the vehicle-to-vehicle (V2V) and road-side infrastructure (V2I) communication [Lee and Gerla, 2010; Fernandes and Nunes, 2007] we can expect to have an increasing interest for a new trend of applications such as cooperative collision avoidance/warning systems, fleet tracking and control, smart adaptive cruise control, autonomous driverless vehicles, video streaming, road obstacle/condition warning, blind spot detection, cooperative lane changing assistance, among many others. These applications are often classified as comfort and/or safety systems and some of them have also been studied for autonomous vehicles, i.e., without taking the benefits of the underlying VANET communication capabilities.

Cooperative target tracking [Liu et al., 2003; Rockl et al., 2008] in VANET environments is a relatively novel challenge that differs from the traditional mobile ad hoc networks in the sense that the mobility pattern of a vehicle is considerably different, restricted by roads, lanes and junctions. Moreover, the network density is very inhomogeneous and time- and location-dependent. It is also noteworthy that vehicles are typically not affected by strict energy constraints and can be equipped with a wide variety of sensors and processor units.

Almost all of the aforementioned applications require knowledge of the vehicle's position to work properly. Even when applications do not directly take benefit of the vehicles' position, the underlying data dissemination protocols can take much advantage of such information. For instance geocasting and geographic routing rely on the vehicle's location and are desirable for many VANET scenarios [Yousefi et al., 2006].

Beyond the location knowledge, target tracking can be used to detect and predict future trajectories of single or multiple targets such as other vehicles,

people, animals, bicycles, motorcycles, and obstacles surrounding a given vehicle. Along this work, we call “ego” vehicle the one to which target tracking techniques are being applied and targets are all other objects including other vehicles being tracked. We are interested only in target tracking systems that take advantage of the VANET communication model, i.e., V2I or V2V, namely Cooperative Target Tracking systems (CTT). We use this term for both, target or multi target tracking [Liu et al., 2007] indiscriminately, being the single target tracking the specific situation when the number of targets is equal to one.

Considering the driver’s point of view, CTT techniques can be applied to augment the driver’s perception of the surrounding context and to increase the comfort and safety of the driving experience. The results of the target tracking system can be used to either automatically actuate on the vehicle such as breaking or steering, or just assist the driver to take the correct action whenever some situation occurs.

Fig.5.1 depicts some scenarios where CTT can be applied in different situations to assist the drivers. Considering the vehicle A as the ego, CTT techniques can be applied to warn driver A that vehicle D is coming on the opposite lane while keeping track its distance to other vehicles. In this situation, vehicle B might be blocking the driver’s view (B might be a large car or a truck, for instance), hence the driver of vehicle A can safely avoid any kind of risky maneuver such as overtaking B. Other scenarios can be illustrated from this figure. For example, E might be warned by D about the presence of a bicycle and a pedestrian on the same lane that E is about to share while it turns the corner. Both the bicycle and pedestrian might be detected by sensors installed in D and/or by roadside infrastructure. Similarly, F might warn C about the unexpected unleashed dog and the upcoming vehicle G.

The aforementioned examples illustrate how the CTT can increase the drivers’ confidence and, thus, the safety while driving in heavy urban traffics. Similar approaches can be derived to assist highway traffic such as lane changing, overtaking, and lane merging assistance. CTT can also be applied to assist fleet management, allowing smart rerouting and reschedule, and traffic management.

As stated by Rong-Li and Jilkov [2003], the effective extraction of useful

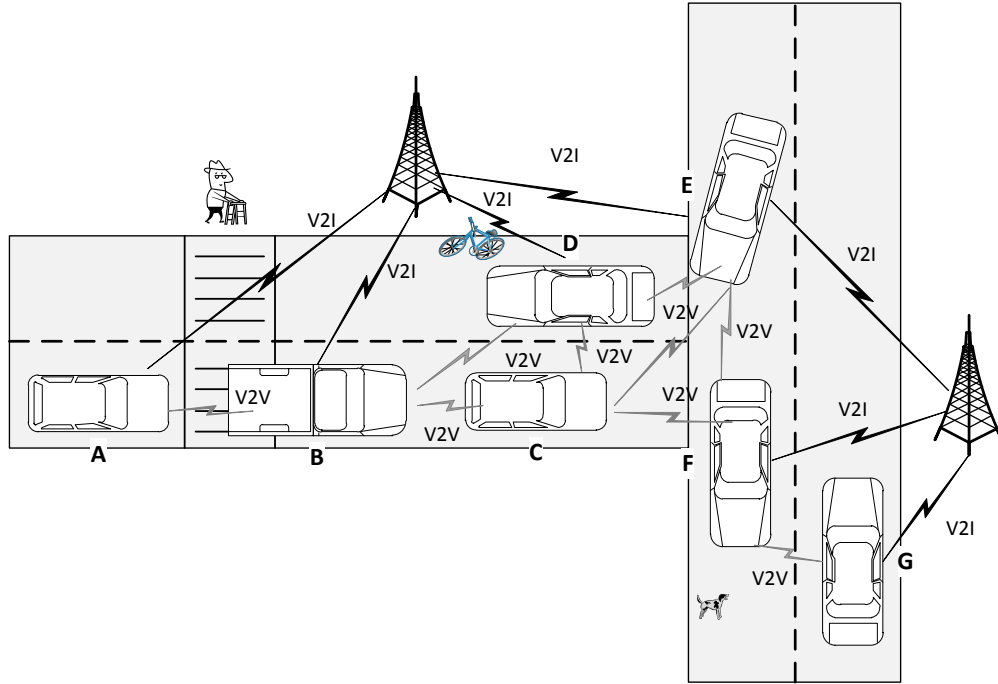


Figure 5.1: Cooperative target tracking scenarios

information about the target's state plays a crucial role in any target tracking system. By effective extraction in a VANET environment, we consider: (i) targets' motion model, and (ii) measurements of the targets' state. In a VANET scenario, communication issues and the large number of rapidly maneuvering targets influence the target state estimation quality and should be considered and further investigated. Thus, in this chapter we address the problem of how to cooperatively track a target (or multiples targets) considering the innovative aspects of vehicular ad hoc networks.

The rest of this Chapter is organized as follows: the problem we are interested in is stated in Section 5.2. The main components of the CTT systems are detailed and discussed on Section 5.3, while the Chapter remarks are present on Section 5.6.

5.2 Problem statement

Target tracking is the capacity of detecting and continuously tracking the state of a target, or a set of targets [Gustafsson et al., 2002; Hue et al., 2002]. Tracking is usually stated as an estimation problem based on a series of measurements: the main goal is to estimate the targets' state and update the estimation with measurements. Suppose that the target trajectory is described by the following discrete-time system

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (5.1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (5.2)$$

where $\mathbf{x}$, $\mathbf{u}$, $\mathbf{z}$ are the target state, input control and observation, respectively, $\mathbf{w}$ and $\mathbf{v}$ are the process and measurement noise, respectively, $\mathbf{f}$ and $\mathbf{h}$ are function vectors, and $k \geq 1$ is the measurement epoch. All variables and functions are related to the discrete time step t_k , thus, $\mathbf{x}_{k+1}$ is the target state estimate at the next time step.

The main approach to solve this problem is the Bayesian state estimation [Simon, 2006]. Its goal is to approximate the conditional probability of $\mathbf{x}_k$ based on measurements $\mathbf{z}_k$. It can be denoted as $\Pr(\mathbf{x}_k | \mathbf{z}_j, j \leq k)$, that is the probability of $\mathbf{x}_k$ conditioned on measurements $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots, \mathbf{z}_k$.

In the following, we present the main components of target tracking systems. We describe the motion models, which determine the function vector $\mathbf{f}$, the measurement models, which determine the function vector $\mathbf{h}$, and the filtering techniques, which drive how the pdf of $\mathbf{x}_k$ will be recursively computed. Other aspects such as data association and VANET communication characteristics are also discussed.

5.3 Components of cooperative target tracking systems

Target tracking systems aim at continuously detecting and estimating the state of a set of targets. The targets' state can include, among others, position, veloc-

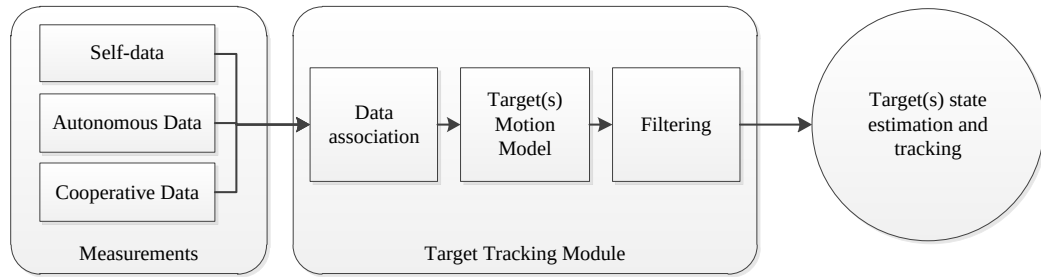


Figure 5.2: Basic components of cooperative target tracking systems

ity, acceleration, and jerk (derivative of acceleration). The set of state variables can vary to meet the application requirements and constraints. Target tracking systems typically rely on a model-based (motion and observation) Bayesian estimation framework and require: a motion model that describes the target's dynamic; measurements of the target's state; a data association algorithm that relates the measurements to the correct target; and an initial probability distribution, also known as prior knowledge of the target's state. Based on the motion model, the main task is to estimate the parameters of the model, considering the measurements. This task is usually performed by a Bayesian filter, such as the Kalman filter [Simon, 2006; Aydos et al., 2009] and its variations, and the particle filter [Simon, 2006; Arulampalam et al., 2002].

A schematic view of these components is presented in Figure 5.2. Based on these components, the target tracking system performs in two steps: (i) prediction, that uses the motion model to propagate the probability function of the target state over the time, and (ii) correction, that uses the latest measurements to update the probability density function of the target at the current time step. The main difference from the cooperative target tracking schema shown in Figure 5.2 to the autonomous target tracking, is the cooperative data, which is provided through the communication capability of the vehicles (V2V and/or V2I).

5.3.1 Motion models

The most common approach used in target tracking systems is to describe the target's dynamic as the evolution of the target's state along the time (Eq. 5.1). The main reasons of the popularity of model-based systems are that the models are almost always available, and they usually outperform any model-free system [Rong-Li and Jilkov, 2003]. The target motion uncertainty, i.e., the lack of an accurate model of the target from the tracker's viewpoint, is one of the most challenging problems in target tracking. Therefore, the choice of an expressive and tractable model, which captures the target dynamic, poses an important and challenging role for the success of target tracking systems.

There are many different motion models, with different complexity and parameters in the literature. Rong-Li and Jilkov [2003] survey a variety of motion models, while Schubert et al. [2008] compare some of them considering the vehicular target tracking context. In this section, we aim at revisiting some discrete-time motion models characteristics and discussing their possible applicability to VANET scenarios.

In this work we consider a classification similar as presented in Schubert et al. [2008], and we consider two groups of motion models: 1D models, which assume the target makes straight movements at each time step, and 2D models, which the coupling between the coordinates is considered.

5.3.1.1 1D Motion Models

The most simple motion model is one that considers a piecewise constant target velocity (CV). This model has the advantage of presenting a linear state equation, which can lead to an optimal estimation.

Hence, the model assumes straight target movements that might not be suitable for some applications. For instance, consider the case where the time elapsed between the state measurement k and $k + 1$ is large enough to the target object changes its movement performing a curved trajectory. As this model considers only straight movements, curved trajectories will not be well represented due the model restrictions. As target tracking systems correct their estimation by periodically measuring the target state, as shown in Eq. 5.2, the length of the mea-

surement interval provides an important leverage to this problem. Large measurement intervals can typically occur in VANET scenarios due to packet losses and multi-hop wireless communication delay, for instance.

The CV model is also called “nonmaneuver” because no control input u is supposedly applied, hence the velocity is considered piecewise constant over each time step. Rong-Li and Jilkov [2003] also include a small white noise acceleration model, which leads to a CV model called nearly constant velocity model. This model is reasonable to some specific scenarios where the discrete time interval is small and the velocity can be updated by measurements constantly (at each time interval). As consequence of the assumption that no control input is present, the CV model holds for such situations where only the two first terms of the Taylor expansion for the displacement equation are representative. Thus, all high order terms, greater than the second derivative, are negligible for this model.

The control input u should be incorporated to the model to improve its realism. Even though the control input is intrinsically deterministic, it is commonly represented as a random process due the lack of knowledge of its dynamic [Rong-Li and Jilkov, 2003]. Thus, the control input is frequently modeled as: (i) white noise, (ii) Markov process, or (iii) semi-Markov jump process. The most popular model of this category is the constant acceleration model (CA), which considers a piecewise constant acceleration. The CA model usually assumes that the acceleration is a random process, for example, a white noise or a process with independent increments, such as the Wiener process. The assumption is too simplistic because the acceleration is barely time-uncorrelated in real systems [Rong-Li and Jilkov, 2003]. Notwithstanding, its simplicity leads to a frequent adoption in target tracking context. Many acceleration models from the simple CA to more realist ones have been proposed and some of them are thoroughly discussed in Rong-Li and Jilkov [2003]. These models are appropriate when the vehicle’s acceleration is not negligible during the time step. Additionally, the acceleration should be measured or properly modeled.

5.3.1.2 2D Motion Models

To circumvent the straight target movement assumption other models consider the rotation around the vertical axis, and consider the coupling among coordinates. They are called coordinate-coupled maneuver models, or simply 2D horizontal model (or flatland motion model). In this work we do not consider 3D models once they are more useful for other target tracking context such as for aircrafts. Coordinate-coupled target motion models depend on the underlying target kinematic motion model, and, consequently, on the choice of the state space. The simplest model on this category is the Constant Turn rate model (CT). In this model, the motion is usually described by a set of state variables such as target position (x, y) , velocity v , acceleration a , heading ϕ , and turn rate $\dot{\phi}$ (also referred as yaw rate, the derivative of heading). Thus, two models can be easily devised when the acceleration is null (CTV), and when the acceleration is constant (CTA). Those models consider that the velocity and the yaw rate are uncorrelated, leading to imprecision like the possibility of changing the yaw angle even when the target is not moving [Schubert et al., 2008]. Thus, some models include another state variable, namely the steering angle Φ , used to model the correlation between the velocity and yaw rate. Again, two derivatives can be devised considering constant steering rate and velocity (CSV) and constant steering rate and acceleration (CSA). Table 5.1 summarizes the main characteristics of each aforementioned model.

5.3.1.3 Comments about the Motion Models

The most appropriate motion model to track a maneuver target is application dependent. Observing the third column of Table 5.1, it is noteworthy that the CV model requires only a small set of state variables and measurements, and, consequently, it is simple, and requires few sensors to measure the target state. On the other hand, the most complete model herein shown, the CSA model, requires a more elaborated sensory system. Although models such as CSA and CSV are usually more accurate, they also present some undesirable effects such as drifting and skidding under the presence of small errors [Schubert et al., 2008]. Some models consider these high dynamic behaviors, as discussed in [Schubert

Table 5.1: Summary of motion models

Model	Model Type	State Space and Measurements	Characteristic
CV	1D	x, y, v	Constant velocity
CA	1D	x, y, v, a	Constant Acceleration
CTV	2D	$x, y, \phi, v, \dot{\phi}$	Constant Turn Rate and Velocity
CTA	2D	$x, y, \phi, v, a, \dot{\phi}$	Constant Turn Rate and Acceleration
CSV	2D	$x, y, \phi, v, \dot{\phi}, \Phi$	Constant Steering Rate and Velocity
CSA	2D	$x, y, \phi, v, a, \dot{\phi}, \Phi$	Constant Steering Rate and Acceleration

et al., 2008], and [Buhren and Yang, 2007], and require additional sensors to adequately measure the target state.

The most adequate choice depends on the application accuracy requirements and on the data availability. Autonomous vehicular target tracking using radar, lidar (laser-based radar) and other sensors cannot rely on some local information such as the steering angle, lateral acceleration, tire slip, among others, and simplified target motion models should be used. Conversely, cooperative target tracking systems increase the possibility of make local sensory data available to other vehicles in the neighborhood through the communication capability.

A challenging issue concerning VANET applications is the variety of different kinds of targets such as bicycles, motorcycles, pedestrians, and animals. Notwithstanding, models to describe the dynamics of such targets are not fully discussed in the literature. The CTT system might be able to identify the target and choose the adequate model to describe the target dynamic.

5.3.2 Measurements

As shown in Figure 5.2, in cooperative target tracking context, the measurements come from three different sources: (i) self-data (SD), i.e., the measured data from the ego vehicle, (ii) autonomous data (AD), i.e., the data from other objects

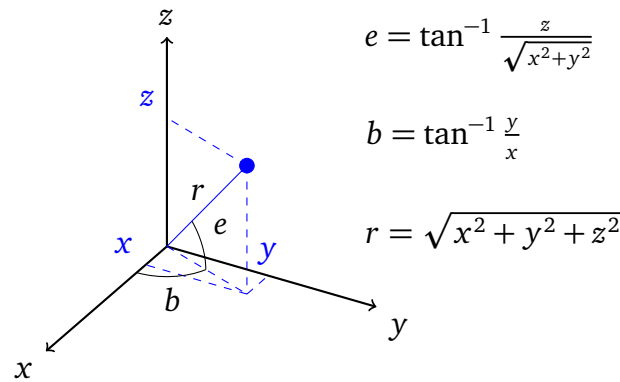


Figure 5.3: Non-linear relation between sensor and Cartesian coordinations system

collected by on-board sensors, and (iii) cooperative data (CD), i.e, the data that other vehicles transmit using the vehicle's communication capability.

Most of the sensors used for target tracking applications provide the measures in a sensor coordinate system, which in many cases is spherical in 3D and polar in 2D, with the following components: range r , bearing (or azimuth) b , elevation e , and possibly range rate (or Doppler) $\dot{r}$ [Li and Jilkov, 2001]. As target motion is best described in a Cartesian coordinate system, some possibilities arise such as tracking in mixed coordinates, in Cartesian coordinates, or in sensor coordinates [Li and Jilkov, 2001]. The challenge is that the relation between the sensor and Cartesian coordinates are highly nonlinear, as shown in Figure 5.3.

Thus, the use of Eqs. (5.1) and (5.2) with a mixed coordinate systems usually leads to a nonlinear system even when the target motion is described by a linear model as the CV motion model. A nonlinear system incurs the use of nonlinear filtering techniques that are not optimal, are hard to tune, and might be only properly handled by high computational demanding filters. Issues about linear and nonlinear filtering are discussed in Section 5.3.5. Nevertheless, the most natural solution for this problem is the use of mixed coordinates (Cartesian and sensor) with nonlinear filtering.

Li and Jilkov [2001] provide a detailed survey on AD measurements for non-cooperative target tracking systems. They show many approaches for the three aforementioned categories of motion and measurements models, and they

also discuss about other systems different from Cartesian and sensor coordinates. These other systems are more suitable for specific applications and are not considered along this work. On one hand, mixed coordinates approaches must deal with the nonlinear models, and hence, nonlinear filtering. On the other hand, other approaches strive to convert the measurements from the sensor coordinates to Cartesian coordinates (or vice-versa) to take advantage of the linear filtering properties. Although there are some proposals on this direction, the linear filter is not anymore optimal for those cases, once the noise is highly non-Gaussian and is state dependent. Their survey shows techniques that cope with this problem by finding statistical properties of the noise measurement such as the first and second moment. Monte-Carlo method, or the computational friendly Quasi-Monte-Carlo method, may be alternatively used to estimate the noise statistics for more complex situations, for example when the noise is modeled by less tractable distributions. Motion models based on the sensor coordinates are also highly nonlinear and it seems to be much more difficult than the use of Cartesian coordinates [Li and Jilkov, 2001].

Other sources of data are available in CTT systems due to their communication ability. In this scenario, the vehicles can broadcast their self-measurements to their neighborhood. Hence, the SD data can be made available for all neighbor vehicles, and, depending on the application, it can be made available for any reachable vehicle in the network by multi hop data dissemination. For instance, vehicles can broadcast their GPS, compass, speedometer and gyroscope data to their neighbors. Thus, a mixture of data collected on Cartesian and sensor coordinates is available. The CTT system must be able to cope with this new data.

In VANET environments the measurements is challenging once the target behavior is very dynamic, usually the number of targets is unknown and the targets typically perform quick maneuvers. Thus, a good combination of motion and measurement models is important to achieve good tracking results.

5.3.3 Data association

In a presence of multiple targets and/or multiple sensory data, CCT systems might face the data association problem. This problem occurs when the system

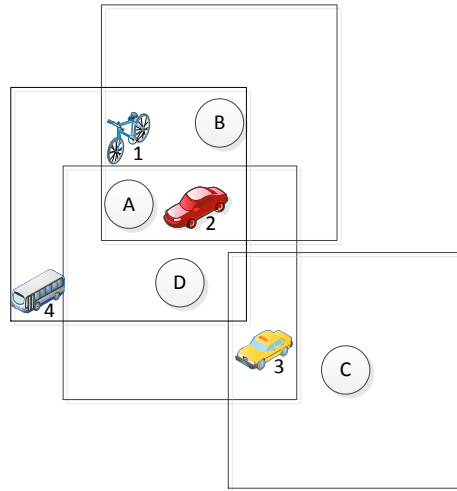


Figure 5.4: Data association problem

has to associate the observed measurements to known targets at each time step. In VANET scenarios, this issue arises in any of the following situations: (i) multiple autonomous sensors, and (ii) autonomous and cooperative sensory data are used together, in a presence of multiple targets. There is no problem in identifying the measured data when only cooperative data is used because the origin of these data is identified by the network address. The data association problem is also known as measurement-to-track association, is shown to be NP-hard [Smith and Singh, 2006], and is one of the most challenging problems for multi-target and/or multi-sensor tracking systems.

To illustrate, suppose that vehicles C and D in Figure 5.1 detect the bicycle using their autonomous sensor (e.g., radar). When they broadcast these data to vehicle E, for instance, the problem is how the tracking system of E knows whether these two measurements (from C and D) originate from the same target or not.

Figure 5.4 depicts the data association problem. Objects 1, 2, 3, and 4 represent general targets (e.g., other vehicles, bicycles, and pedestrians) detected by an ego vehicle; circles A, B, C, and D represent measurements obtained from the local sensors and/or cooperative data. The rectangles around the measure-

ments represent the system belief on the specific measurement, also referred as prediction window. Those windows are usually determined by the accuracy of the involved sensors.

Firstly consider a single sensor capturing all measures. This is a typical aircraft monitoring situation where radar is used to track multiple targets. A simple heuristic to this problem is to associate measure A to target 4, as no other measure can be associated to it. Hence, measure B can be associated to target 1, measure D can be associate to the target 2, and finally, measure C can be associate to the target 3. This specific scenario could be solved in a simple way, and all measures are associated to the most plausible target. Furthermore, this problem can be much harder to solve in a dense and cluttered scenario. Actually, the number of associations is factorial in the number of targets [Smith and Singh, 2006]. However, if we consider that the measures can be captured by multiple sensors, even this simple scenario can be hard to solve. For instance, we cannot eliminate the hypothesis of measures D and B are associated to target 2, as well as D and C are associated to target 3. Thus, the data association problem for multiple targets and multiple sensors is even harder to be solved. This problem became harder in real world environments where false alarm and missed measurements are likely [Liu et al., 2007]. In the context of a VANET, clutter and dense scenarios are likely, especially considering urban applications. Thus, the data association problem must be addressed when cooperative and non-cooperative data are used at same time.

Some heuristics have been presented in the literature to solve this NP-hard problem. Smith and Singh [2006], and Liu et al. [2007] discuss exponential solutions and polynomial heuristics for this problem.

5.3.4 Communication

The communication model plays a crucial role on CTT systems once all cooperative data are exchanged through a wireless channel either using V2V or V2I communication. There are issues regarding the communication channel that can influence the CTT performance such as delay and packet loss.

Different access methods of communication such as WiFi, cellular, and

WiMax can be used for VANETs. However the DSRC/WAVE protocol is the most promising solution since it was devised specifically for vehicular communications. Dedicated short-range communications is a standard also referred as IEEE 802.11p. It is a recently approved amendment to IEEE 802.11 standard that adds Wireless Access in Vehicular Environments (WAVE). This standard defines the use of the licensed 5.9 GHz band dedicated for V2V and V2I ITS communication. The frequency range 5.850 – 5.925 GHz is divided into seven channels of 10 MHz each, reaching high communication rates of order of 6 to 27 Mbps. The channels are half-duplex and the typical communication range is 300 m (up to 1000 m). The standard is designed to have an expected low latency in the order of 50 ms, and features eight priority levels. The channel allocation is designed in a way that most central channel is the control channel, which is restricted to safety communication only, the two channels at the edges of the spectrum are reserved for future advanced accident avoidance applications, and the rest is designed for general use [Jiang et al., 2006].

There are two main units: On-Board Unit (OBU), and Road-Side Unit (RSU). The RSU is designed to broadcast messages about 10 times per second, announcing warning messages and promoting applications. The OBU unit listens to the control unit, authenticates the RSU and executes applications. A public key infrastructure is used to provide the necessary security and authentication services.

Communication characteristics such as low latency, reasonable communication range, priority levels to provide the desired quality-of-service, and licensed band are indicatives that this technology will provide suitable conditions for CTT applications. As the standard was approved in July 2010, it is quite new and lacks of thorough assessments.

CTT systems do not require a large amount of data to be communicated. Basically, the nodes need to send their state variables set to their neighbors. This set typically varies from 3 to 7 variables (float) (see Table 5.1), and thus, the payload varies from 96 to 224 bytes, plus a timestamp. However, communication issues present in VANETs can pose an important role in the design of a CTT system. For example, collision detection systems demand high accurate estimates of the vehicles' state. To do so, the vehicles should broadcast their state variables set very

frequently. This leads from the fact that the estimation process error increases when the time elapsed between measures increases. Furthermore, even in presence of packet losses and temporary network disconnections, the CTT system still can be able to use the estimated data instead of measures, but paying the cost of loss of accuracy. A thorough evaluation on how the VANET communication issues should impact on CTT systems is still missing in the literature.

5.3.5 Filtering

The filtering component of target tracking systems is responsible for defining how the pdf of the target's state at time step k is recursively calculated by a Bayesian approach. The Kalman filter [Aydos et al., 2009; Iqbal et al., 2010; Baheti, 1986; Rawicz et al., 2003] is a particular case of Bayesian filtering under the assumption that the target dynamic f , described by the motion model and the measurement model h are both linear; moreover, the uncertainties w and v in both models are assumed to be Gaussian and uncorrelated. If all those assumptions hold, the posterior probability $\Pr(\mathbf{x}_k | \mathbf{z}_k)$ is also Gaussian, and the Kalman filter recursively updates its mean and covariance based on the measurements updates [Liu et al., 2007]. The Kalman filter presents some interesting properties such as being the optimal estimator when the noise is Gaussian, and is the linear optimal estimator even when the noises are not Gaussian [Simon, 2006]. This is optimal in the sense that it provides an unbiased minimum variance state estimation.

The Kalman filter is the best option when all assumptions hold. However, system linearity and precise knowledge of the system properties (motion model, measurement model, and noises covariance) are not always granted for real systems. Some techniques can be used to relax some assumptions. For instance, the fading-memory filter can be used to place more emphasis on recent measurements and make the filter more robust to modeling error, paying the cost of having a suboptimal filter.

Variations have also been proposed for relaxing the linearity assumption. The Extended Kalman Filter (EKF) [Hue et al., 2002] is a popular approach to implement nonlinear filters. The rationale behind the EKF is that the state distribution is approximated by a Gaussian law, whose density is then propagated using

the first-order Taylor series expansion to linearize the system. Although EKF is a common nonlinear filter technique, the Unscented Kalman Filter (UKF) [Aydos et al., 2009] is a more recent variation of the Kalman filter and represents a great improvement to EKF. UKF uses a deterministic minimal set of sample points to represent the Gaussian random variable (with same mean and covariance) that approximates the target's state. Those points are propagated through the true nonlinear system. This approach is known to capture the posterior mean and covariance accurately to the third order of the Taylor series expansion for any nonlinearity [Simon, 2006]. UKF is known to greatly improve the performance for nonlinear systems when compared to EKF, because it does not have to deal with linearization errors. Another issue is that EKF can be difficult to tune and is able to cope only with slight nonlinearities.

Although UKF improves the nonlinear filter performance, it is still an approximate nonlinear estimator. Thus, even UKF can present bad accuracy and divergence issues when the system nonlinearities are severe. Particle filter [Hue et al., 2002; Rockl et al., 2008; Arulampalam et al., 2002; Gustafsson et al., 2002; Lu et al., 2010] represents another class of filtering that estimates the target's state through a brute-force approach. Particle filter can often cope with nonlinearity and with non-Gaussian noise when Kalman filter approaches do not perform well. The key idea is to represent the posterior function $\Pr(\mathbf{x}_k | \mathbf{z}_k)$ by a set of random samples, called particles, which are sequentially propagated over time. At each time step, some particles that present low posterior probability are discarded by a process called resampling. To each particle it is associated a weight indicating its quality, thus, the estimate is the resulting of the weighted sum of all particles [Nakamura et al., 2007].

Generally speaking, the Kalman filter is the optimal choice when the system is linear with Gaussian noise. Particle filter can outperform the Kalman filter especially for the nonlinear case, with the cost of additional computational effort, because it typically requires a large number of particles to present accurate results. UKF is an intermediate solution between the Kalman filter and the particle filter. It usually provides reasonable accuracy with low computational cost. For a VANET scenario, the computational constraint is not an important issue, thus particle filter represents a good candidate. However, if the number

of required particles is too high, the computational time can impose a great constraint especially when time critical applications are considered, such as collision warning/avoidance systems. Thus, a careful choice of the most appropriate filter should be considered. There are other filters that can be applied to target tracking, as presented in Simon [2006].

Regardless the filter choice, this process must cope with the particularities of the CTT data type. Self- and autonomous-data generally are more controlled and are always timely available. However, cooperative data, as described in Section 5.3.2, may suffer from data loss or delay that should be processed by the filter. Though the communication channel features interesting QoS aspects (e.g., priority queues and low latency – see Section 5.3.4), applications such as collision detection/avoidance systems, require strict time constraints and may suffer with the networked data behavior. Thus, the filter must be able to tackle the two most common data problems in networked data, loss and delay.

Some techniques to deal with missing data can be borrowed from the time series theory. Most of them can be classified as: (i) interpolation, (ii) state space models, and (iii) stochastic models. The first is a deterministic approach and sounds reasonable only when the vehicle is performing straight movements, but may not be appropriated when the vehicle performs abrupt maneuvers such as turning a corner. This problem may be exacerbated by the fact that the data is more likely to be lost or delayed in sequence (burst loss/delay), and thus, important measures of the maneuvers can be missing. State space models are used when the predict step of the filters is suitable to represent the missing data. These models may suffer from the same kind of weakness of the deterministic models once the motion model typically cannot represent abrupt maneuvers without appropriate measures. Stochastic models can be carefully devised to try to accommodate those characteristics and cope with those problems. The choice of the best method to handle missing data in CTT applications is a topic that still deserves more attention.

The accuracy of CCT systems depends mostly on how the modules presented in Figure 5.2 are designed. The choice of the motion model, the quality of the available measures, and the design of the filtering technique can dramatically influence the accuracy. For cooperative data, the frequency that the localization

data is transmitted to the neighbors also influences the prediction accuracy. For the sake of exemplification, Lu et al. [2010] adopt a particle filtering-based target tracking scheme to implement a WIFI fingerprint-based localization system. They achieved root mean square error ranging between 14.5 to 24.8 m. A comprehensive evaluation of the prediction error is still missing.

5.4 Case study: data dissemination in VANETs

In VANET scenarios, CTT systems can be used in a wide variety of applications. The most common applications are those that CTT is used as main role, i.e., the application itself is directly related to CTT. For example, collision warning/avoid systems can take much advantage of CTT as it can use the vehicles' state and predictions from a CTT module. CTT can also be seen as a main role for other aforementioned applications such as fleet tracking and control, smart adaptive cruise control, autonomous driveless vehicle, blind spot detection, and cooperative lane changing, mostly because those applications take immediate advantage of CTT technique in order to work properly.

CTT can also assume another important role for vehicular applications when it is used as a support to improve other functions. For instance, CTT can improve network functions as routing, data dissemination and media access, for instance. In those applications, CTT is useful for managing the highly dynamic topology presented in VANETs. With a CTT module, the nodes can store their neighbors state and can predict near-future states, and thus, have a reasonable view of the network topology. In this Section, we present a data dissemination algorithm that uses a CTT module in order to improve the data dissemination process. In our approach, the nodes only stores the states of the neighbors that are directly involved in the data communication process. Our algorithm uses the neighbors' location and near-future predictions in order to drive the relay selection process. We show that the use of CTT improves the data dissemination performance. We also developed a video unicast protocol on top of our data dissemination algorithm, where we show that the use of our algorithm decreases the delay and the frame loss when compared to baseline solutions. Next Sections present a review of the literature on this problem, the details and the evaluation

of our algorithm.

5.4.1 CTT-based data dissemination algorithm

The CTT-based Data Dissemination algorithm (CTTDD) herein presented is a reactive solution as its forwarding mechanism is receiver based. It means that instead of forwarding nodes choosing the next hop in the route, receiving nodes are responsible for determining if they are suppose to relay the packet. This is a well-known technique and we have adopted a variation of the common greedy geographic method where all receiving node schedule themselves to forward further the packet with a time proportional to how much closer they are to the destination. If a node overhears a node closer to the destination forwarding the same packet that it has previously scheduled before its own waiting time expires, it drops the packet. For CTTDD works, it is suppose that all vehicles are equipped with GPS (Global Positioning System) or similar localization mechanism.

The problem with receiver-based solutions is that the waiting time for each hop leads to an excessive end-to-end delay. We have tackled this issue through the perspective that in each hop there is a competition in order to determine the most suitable node to forward the packet. Existing solutions conduct this competition for every single transmission. However many successive packets are transmitted in a connection, thus, repeated competitions are redundant. For this reason, in CTTDD, whenever a node *wins* as the most suitable candidate, it supposes that it continues to be the best option to relay the packet for a limited time window. Within this time window, any packet received is immediately forwarded. This solution does not impact CTTDD's ability to react to path breaks as if a node that has *won* the competition is no longer available, one of the other neighboring nodes eventually reaches its waiting time, forwards the packet, and consider itself as the best suitable forwarding candidate for further packet transmissions.

CTTDD employs a location tracking mechanism able to estimate near-future locations of nodes so it can predict the contact time between neighbor nodes. This estimation guides the time window length by which nodes relay packets without delay. The data dissemination is preceded by a request. Thus, the client requests data from the server by sending its location piggybacked into the request packet.

Upon receiving the request, the server starts sending the data to be disseminated towards the client's location. Both the client and the server need to periodically send their location information in order to feed the target tracking system of each other with their GPS measurements.

The forwarding process is guided by the modified greedy geographic method from CTTDD. As a receiver-based approach, nodes do not need to store any kind of neighboring table and do not exchange beacon. All the required information is piggybacked into the data and request packets. This forwarding mechanism leads the CTTDD approach to be highly reactive to topology change, since, as long there is an eligible receiving node, the packet is always forwarded.

In CTTDD forwarding mechanism, only nodes that are inside a *forwarding zone* are eligible to forward the data. The forwarding zone is defined as a sector of the communication radius of the last transmitter. Figure 5.5 illustrates the forwarding mechanism. The source node sends a packet towards the destination. The forwarding zone is the gray circle sector. Nodes inside the forwarding zone schedule themselves to retransmit the packet after a *waiting time* (γ_{geo}) proportional to the distance to the destination. Thus, the node closer to the destination (black node inside the forwarding zone) have the timer expired before all others. This node forwards the packet. Upon overhearing the packet already scheduled, all other nodes (gray nodes inside the forwarding zone) cancel the transmission of this packet. This process is repeated hop-by-hop until the packet reaches the destination. In CTTDD, the black nodes elect themselves to be the default transmitter, without delay, of further packets from the same connection for a specific time interval. We refer to this time interval as *reservation time* (λ). White nodes outside the forwarding zone discard the packet and do not participate on the forwarding process.

By using the forwarding zone scheme, CTTDD alleviates one of the major drawbacks of the traditional receiver-based solutions, that is when nodes are not able to overhear the forwarder, and thus, cancel its schedule transmission. This leads to concurrent transmissions that may overload the network channels. This failure in the canceling processing mostly happens due to hidden terminals. The angle θ defines the forwarding zone of each transmitter. The smaller the θ , the smaller the probability of having eligible nodes to forward the packet. On the

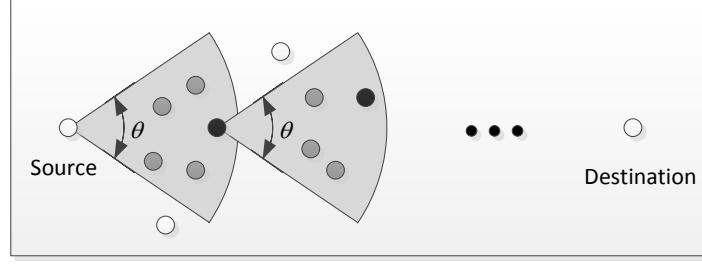


Figure 5.5: Forwarding zone

other hand, it is straightforward to show that if $\theta > 60^\circ$ the forwarding zone may not be able to avoid hidden terminals.

The waiting time of the CTTDD forwarding mechanism is defined as:

$$\gamma_{geo} \triangleq \left(1 - \frac{\delta(v_{lh}, v_d) - \delta(v_r, v_d)}{r} \right) \Gamma \quad (5.3)$$

where $\delta(\bullet)$ denotes the distance between two nodes, v_{lh} is the last hop node, v_d is the destination, v_r is the receiving node, r is the communication radius, and Γ is the maximum waiting time. In real-world scenarios, r is approximated by the theoretical maximum communication radius.

The reservation time is defined as:

$$\lambda \triangleq \min \{ \kappa(v_{lh}, v_r), \Lambda \} \quad (5.4)$$

where Λ is the maximum reservation time, and $\kappa(\bullet)$ is the estimate of the contact time between two nodes. This metric is estimated by using the prediction step of the target tracking mechanism.

By using the reservation time, CTTDD overcomes another inconvenience of the receiver-based approaches, that is the large delay imposed by the successive waiting time competition for every single packet. Thus, CTTDD identifies flows and reserves the channel for a specific time windows where packets are transmitted without delay.

The forwarding zone alleviates but does not completely avoid the multi-

ple forwarders effect (concurrent transmissions) typical of receiver-based mechanism. This happens due to two main reasons: (i) the forwarding zone defines a theoretical area where all nodes can reach each other, but in real world scenarios, obstacles and fading may lead to hidden nodes even inside the forwarding zone, and (ii) nodes that are very close to each other have very similar waiting time, and thus, may not overhear the transmission of the node closer to the destination timely. While the former is unavoidable¹, CTTDD employs a mechanisms to alleviate the effect of the latter. All nodes include into the packet the value of the hop level and the γ_{geo} . Thus, even if a node that has reserved the channel receives a packet that indicates a waiting time lower than its for the same connection and hop level, it cancels the channel reservation alleviating the multiple forwarders effect.

The next section details the target tracking mechanism employed by CTTDD.

5.4.2 Target tracking mechanism

In CTTDD approach, nodes receive the location measurements from their own GPS (self data) and from the the transmitting vehicles' GPS piggybacked into data packets (cooperative data). Other type of measurement can also be considered, such as on-board proximity sensors (autonomous data) in order to cope with GPS outages, but they are not considered in this work. Hence, no data-association algorithm is necessary for this specific case once networked data is easily identified. We adopted a motion model that assumes a nearly constant velocity model [Rong-Li and Jilkov, 2003], also known as white noise acceleration model. This model considers a small acceleration ($\mathbf{w}_k$) modeled by a white noise [Schubert et al., 2008]. For this model, the Equation (5.1) can be rewritten as

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + G\mathbf{a}_k, \quad (5.5)$$

where $\mathbf{a}_k = [\alpha_{x,k} \ \alpha_{x,k} \ \alpha_{y,k} \ \alpha_{y,k}]^T$ is the noisy acceleration, with α_x and α_y independent and identically distributed, $\alpha_x, \alpha_y \sim N(0, \sigma_a^2)$. Let us expand the

¹Although multipath flows originated by this reason are likely to be pruned along the path because they tend to follow paths close to each other

Equation (5.1) as

$$\begin{bmatrix} x_{k+1} \\ v_{x,k+1} \\ y_{k+1} \\ v_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x,k} \\ y_k \\ v_{y,k} \end{bmatrix} + \begin{bmatrix} \Delta t^2/2 & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t^2/2 & 0 \\ 0 & 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_{x,k} \\ a_{x,k} \\ a_{y,k} \\ a_{y,k} \end{bmatrix} \quad (5.6)$$

where $\mathbf{x}_{k+1}$ is the state vector at time step $k+1$, which consists of the position coordinate (x, y) and velocity (v_x, v_y) vector, Δt represents the sampling interval, and $\mathbf{w}_k \sim N(0, \mathbf{Q}) = \mathbf{G}\mathbf{a}_k$, is the process noise that simulates the noisy acceleration, with covariance matrix $\mathbf{Q} = E(\mathbf{w}_k \mathbf{w}_k^T) = \mathbf{G}\mathbf{G}^T \sigma_a^2$. The process noise $\mathbf{w}$ is modeled as a white noise as described in [Rong-Li and Jilkov, 2003]. Note that the control input $\mathbf{u}$ presented in Equation (5.1) is null in this model.

In this work we are considering only GPS measures, which can be obtained from a CO-GPS module like the one presented in the Chapter 4, or any other GPS solution. These measures are usually in the same coordinate system than the system state. Hence, the output equation (see Equation (5.2)) can be rewritten as:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (5.7)$$

where $\mathbf{H} = \mathbb{1}$ is the system output matrix, $\mathbf{v}_k \sim N(0, \mathbf{R})$, $\mathbf{R} = E(\mathbf{v}_k \mathbf{v}_k^T)$ represents the measurement noise, in our system, the GPS error. Considering the GPS error as normally distributed [Diggelen, 2007] with mean 0 and standard deviation σ_{gps} , we have $\mathbf{R} = \begin{bmatrix} \sigma_{gps}^2 & 0 \\ 0 & \sigma_{gps}^2 \end{bmatrix}^T$.

Equations (5.6), and (5.7) completely describe the system motion and measures. With them, we are ready to choose the most appropriate filtering technique in order to complete the target tracking system as shown in Figure 5.2.

In Section 5.3.5 we discussed the main Bayesian filtering approaches used for target tracking. In the scenario we are considering for CTTDD, we can observe that Equations (5.6), and (5.7) accomplish all conditions for the use of the Kalman filter. Nevertheless, two situations can impair or difficult its usage: (i) near-future vehicles are likely to be equipped with on-board sensors and have autonomous data to feed the target tracking system, thus, the system can become nonlinear and Kalman filter is not appropriate to it. Moreover, (ii) particle



Figure 5.6: Example of multi-modal hypothesis for vehicular state estimation

filter is able to handle multi-modal hypothesis likely to happen on intersections, again, Kalman filter is not appropriate for this situation. Figure 5.6 illustrates how multi-modal hypothesis may appear in vehicular state estimation. Let's assume that a vehicle is approaching an intersection at time T1. Let's also assume that no measurement arrive between times T1 and T2. Thus, there three equally likely different positions for the vehicle at time T2, as represented in Figure 5.6. This situation can be easily represented by particle filter approach, while it is not well represented by a Kalman filter approach.

As we know from Section 5.3.5, Kalman filter is the optimal approach when the system is linear and the errors are uncorrelated and Gaussian. Thus, Kalman filter is the technique that presents the average value of the estimated state equal to the average value of the true state, i.e., the estimate is not biased, and additionally, the Kalman filter is the technique that presents the smallest error variance. There are many alternatives to the Kalman filter equations, all of them are equivalents. One possible formulation is given as follows [Simon, 2006]:

$$K_k = FP_k H^T (HP_k H^T + Q)^{-1}, \quad (5.8)$$

$$\hat{x}_{k+1} = (F\hat{x}_k + Gu_k) + K_k(y_{k+1} - H\hat{x}_k), \quad (5.9)$$

$$P_{k+1} = FP_k F^T + R - FP_k H^T Q^{-1} HP_k F^T, \quad (5.10)$$

where $\mathbf{H}$, $\mathbf{F}$ and $\mathbf{G}$ were defined in Equations (5.5) and (5.7), $\mathbf{Q}$ and $\mathbf{R}$ are the process and measurements covariance noises, as defined before. The $\mathbf{K}$ matrix is called the Kalman gain, the $\mathbf{P}$ matrix is called the estimation error covariance, while $\hat{\mathbf{x}}$ is the system state estimate and $\mathbf{y}$ represents the measurements. The first term of the system state estimate comes from the motion model, while the second term is called the correction term and represents the amount of correction to be applied to the propagated state estimate due the measurement. Observe that, if the measurement noise is large, $\mathbf{K}$ is small and the filter does not give much credibility to it when computing the next $\hat{\mathbf{x}}$. Conversely, if the measurement noise is small, the filter will give more credibility to the measurement when computing the next $\hat{\mathbf{x}}$.

Particle filter can be more suitable for estate estimation in some situations that Kalman filter does not perform well, as discussed above. The rationale behind the particle filter is to represent the posterior function $\Pr(\mathbf{x}_k | \mathbf{z}_k)$ by a set of random samples, called particles, which are sequentially propagated over time. At each time step, some particles that present low posterior probability are discarded by a process called resampling. To each particle it is associated a weight indicating its quality, thus, the estimate is the resulting of the weighted sum of all particles [Nakamura et al., 2007]. We adopted a Sequential Importance Resampling (SIR) [Arulampalam et al., 2002] particle filter, where the pdf of system state $\mathbf{x}_k$ is approximate to:

$$\Pr(\mathbf{x}_k | \mathbf{z}_j, j \leq k) \approx \sum_{i=1}^N \omega_k^i \delta_{\mathbf{x}_k^i}$$

where δ is the Dirac function and $\{\omega_k^i, \mathbf{x}_k^i\}_{i=1}^N$ denotes a set of weights and particles, N is the number of particles. The weights are normalized such that $\sum_i \omega_k^i = 1$. The SIR algorithm can be implemented in four steps:

Initialization: the set of particles is created from the initial distribution $\{\mathbf{x}_0^i\}_{i=1}^N \sim \Pr(\mathbf{x}_0)$ and the weights are initially set to $1/N$.

Prediction: particles move forward according to the system model described in Equation (5.6), resulting a new state hypothesis for each particle.

Weight assignment: uses the information from the measurements to update the particle weights by using the principle of importance sampling [Arulampalam et al., 2002]. Weights are calculated as $w_k^i \propto w_{k-1}^i \Pr(\mathbf{z}_k | \mathbf{x}_k^i)$, and normalized to $w_k^i = w_k^i / \sum_i w_k^i$.

Resampling: particle filters are known to quickly deplete the quality of the particles. After some iterations, some particles present low weights and no longer contribute to the state estimation. Thus, we adopted the effective sample size (ESS) measure to indicate the quality of the sample, in terms of the percentage of particles that are not contributing to the process. The ESS measure can be calculated as $\text{ESS}_k = N / (1 + cv_k^2)$, where cv is the coefficient of variation defined as $cv_k^2 = \text{var}(\omega_k^i) / E^2(\omega_k^i)$. Thus, when ESS value drops below a certain threshold, then the particle population is resampled. There are different resampling approaches, and this step is usually the most computation demanding of the SIR method. We use a approach that runs in linear time, initially proposed in [Carpenter et al., 1999] and also discussed in [Rekleitis et al., 2003].

The next Section presents a performance evaluation of the CTTDD algorithm.

5.4.3 Performance evaluation

It is worth to notice that the efficiency of CTTDD data dissemination mechanism is more noticeable when long data flows are transmitted. This happens because CTTDD uses a channel reservation mechanism based on the target tracking module predictions, and thus, it is able to alleviate some drawbacks of the greedy forwarding process. Thus, we developed the Video Reactive Tracking-based Unicast protocol (VIRTUS) [Rezende et al., 2012], a resilient location-aware video unicast scheme for vehicular networks, built on top of the CTTDD data dissemination mechanism to evaluate the

VIRTUS is developed with a modified waiting time calculation in order to meet the video streaming requirements. As video streams usually present long-term data connection where several packets are transmitted, we observed the the

best candidate to forward the stream packet would be the one that is not necessarily the closest to the destination. Instead, we choose the one that represents a balance between efficient, i.e, the shortest distance to the destination, and link stability, i.e., the highest contact time with the packet sender. Thus, let us define the waiting time referent to the link stability part as $\gamma_{stab} \triangleq (1 - \lambda/\Lambda)\Gamma$. The VIRTUS waiting time can now be defined as:

$$\gamma_{virtus} \triangleq \alpha\gamma_{stab} + (1 - \alpha)\gamma_{geo}, \quad (5.11)$$

where α is the weighting factor between efficiency and stability.

We envisioned a highway emergency response application for the evaluation of the proposed scheme. We used four different scenarios which vary the mobility pattern of source and destination: (i) both are static, (ii) static source and a moving destination, (iii) moving source and a static destination, and (iv) both source and destination are moving. Those scenarios represent from the most static scenario when doctors at hospital want to have access to *in loco* cameras of an accident in a highway, to the most moving scenario where two ambulances cooperate with each other. Other intermediate scenarios are also consider.

These environments were simulated using the Freeway+ Mobility Model [Boukerche et al., 2009] to represent vehicles' movement on a 12 km highway with 3 lanes in each of the two opposing directions with speeds varying from 5 to 40 m/s (with different ranges for each lane). The density is of 50 nodes/km and the two extreme kilometers are avoided to prevent the use of results biased due to vehicles reaching the road extremes. In the three first scenarios, source and destination are place respectively at 1,000 m and at 11,000 m and whenever they are not static, they move at 30 m/s towards the other extreme. For the scenario with two moving ambulances, the ambulance that streams the video starts at 3,500 m and the receiving ambulance starts at 8,500 m, both moving apart at 30 m/s.

We used EvalVid —*A Video Quality Evaluation Tool-set* Klaue et al. [2003] —in order to get results relevant to video streaming. The video transmitted in our simulations is well-known and widely available online (akiyo_cif). It is a MPEG video with resolution of 360x486 composed of 300 frames that could be

fitted into 353 packets each with a payload of 1000 bytes.

We evaluated protocols' performance according to frame loss and delay. Besides that, we compared their feasibility in terms of cost by measuring the rate between the total number of transmissions and packets sent. Frame loss is the percentage of frames that could not be reproduced at the receiver. Delay is the average end-to-end latency of all received frames. For the cost estimation, we used the total number of transmissions at the VIRTUS level divided by the number of packets sent by VIRTUS at the source.

The performance of VIRTUS and two other baseline solutions were conducted using Network Simulator-2². We used the Two Ray Ground propagation model, the IEEE 802.11b MAC with a communication range of 300 meters. All plotted results are averages of from 20 up to 32 independent results each with its own topology instance of the same mobility model. Confidence intervals represents the confidence level of 95% (Student T test was used). VIRTUS' and the other solutions' parameters are specified in Table 5.2.

We compare our solution to an adaptation of a flooding mechanism which can offer lower bounds to delay even if it incurs into high overhead. We call this solution as Guided Gossiping (GG) and it consist of a dissemination directed towards the destination. Any node relaying a packet through GG updates its header with its own position so receiving nodes check if they are in its forwarding zone (the same as VIRTUS forwarding zone). If they are, they forward the packet with a probability ρ . We evaluated GG for different values of ρ and the one with the best performance was with $\rho = 50\%$.

Another baseline is the geographic receiving-based solutions based solely on a geographic greedy approach. This solution, namely GRECV, requires all nodes to schedule themselves for forwarding using the waiting time γ . Receiving nodes schedules themselves for all packets they receive while they are on the last relaying node's forwarding zone. Through exhaustive evaluations we observed that $\Gamma = 200$ ms is the value that GRECV performed better.

VIRTUS performance is influenced by two parameters: α and Γ . Parameter α determines how much the calculation of the waiting time γ_{virtus} is balanced towards stability in preference of efficiency. The maximum waiting time Γ impacts

²ns-2, "The network simulator," <http://www.isi.edu/nsnam/ns>

Table 5.2: Solutions parameters

Parameter	Value
Weighting factor (α)	50%
VIRTUS Γ	100ms
Max Reservation Time (Λ)	5s
GG prob (ρ)	50%
GRECV Γ	200ms
Max Angle θ	45°

in the following situations: (i) nodes have not observed any transmission for long period of time, (ii) the previous relay node expired with reservation time λ , or (iii) the relay node moved out of the transmission range. These situations are rare, therefore, the impact of Γ is less severe than the influence of α . Through exhaustive evaluations of the pair $\nu(\alpha, \Gamma)$, the best performance was observed with $\nu(50\%, 100 \text{ ms})$.

We varied the time between consecutive transmission of packets at the source node in order to analyze the scalability of the solutions in terms of a lower or higher demand of real-time video quality at the receiver. The data rate by which these packets were sent ranged from 80 kbps to 1600 kbps (which almost saturates the 2 Mbps bandwidth used).

The different movement patterns of source and destination did not incur into a noticeable impact on the results (Figures 5.7, 5.8 and 5.9). This is due to the facts that all solutions are reactive to topology changes and because there is an exchange of location/movement updates between source and destination. Any solution based on local neighboring tables would likely be severely degraded with the mobility of the end-points.

VIRTUS great achievement is in reducing significantly the percentage of frames lost, especially for data rates up to 800 kbps, as shown in Figure 5.7. GG suffers from using an excessive number of nodes to relay packets which causes many collisions preventing them to reach the destination. GRECV suffers from a low delivery ratio because the high end-to-end delay imposed by the selection process over many hops demands that intermediary nodes buffer incoming packets and forward them in bursts. Besides that, under high data rate and the high

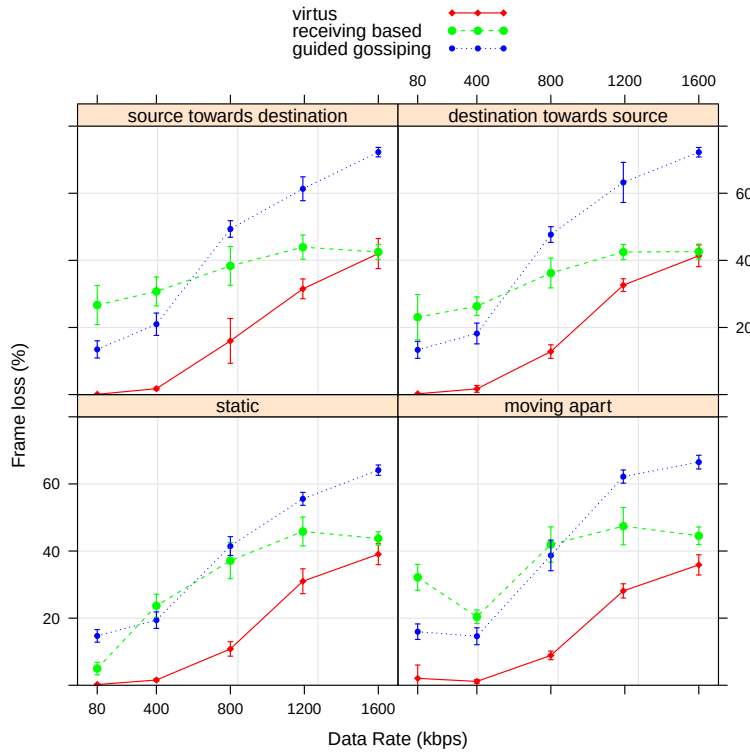


Figure 5.7: Frame loss

delay, nodes will drop packets due to buffer overflow.

In Figure 5.8, we have in log-scale the average delay of each solution. It is important to notice that the delay is calculated only of frames that were received, therefore for higher data rates, where frame loss is high (see Figure 5.7), the delay is biased to the fact that frames with higher delays are more prone to be lost. Independently of that, we can observe that GG has the lowest delay, as expected. It offers a lower bound in delay since none of the transmissions suffer from a waiting time as the receiving-base solutions (both VIRTUS and GRECV). VIRTUS does not only offer the best delivery ratio but it also delivers frames with a delay close to the one observed by GG. This happens due to the fact that most of the transmissions by intermediaries nodes in VIRTUS do not need to wait to forward packets, in case they have already transmitted a previous packet recently under similar conditions. The end-to-end latency achieved by GRECV is prohibitive to real-time video streaming and it reflects the performance of receiving-based

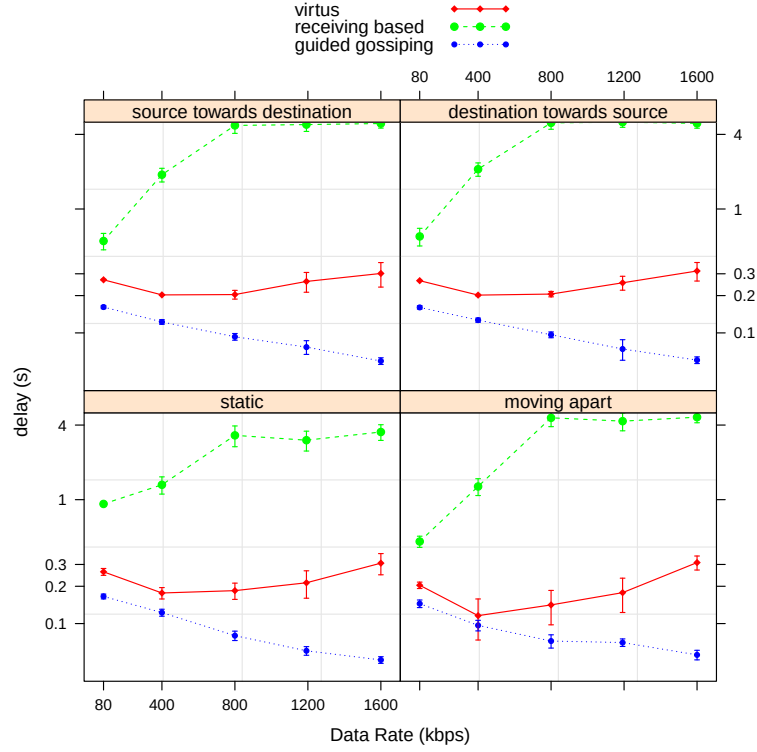


Figure 5.8: Delay

solutions that do not use our mechanism of defining a time window within which nodes consider themselves the best forwarding candidates.

We can observe in Figure 5.9 that all solutions have a reasonable amount of transmissions per packets with the exception of GREC for data rates of 800 kbps or higher (once again the graph has its y axis in log-scale). The smallest paths used between source and destination vary between 35 to 40 hops (for the scenario with end-points moving apart the paths are half the size since they are physically separated for half the distance). Therefore, we observe that VIRTUS' and GG's number of transmissions have space to improve, we notice that between 3 and 5 transmissions are performed per path hop. It is important to notice that VIRTUS always explore better the trade-off of cost (transmissions) and benefit (lower frame loss).

VIRTUS shown to outperform baseline solutions as it is able to deliver a higher percentage of packets resulting in fewer frames lost at the receiver. The

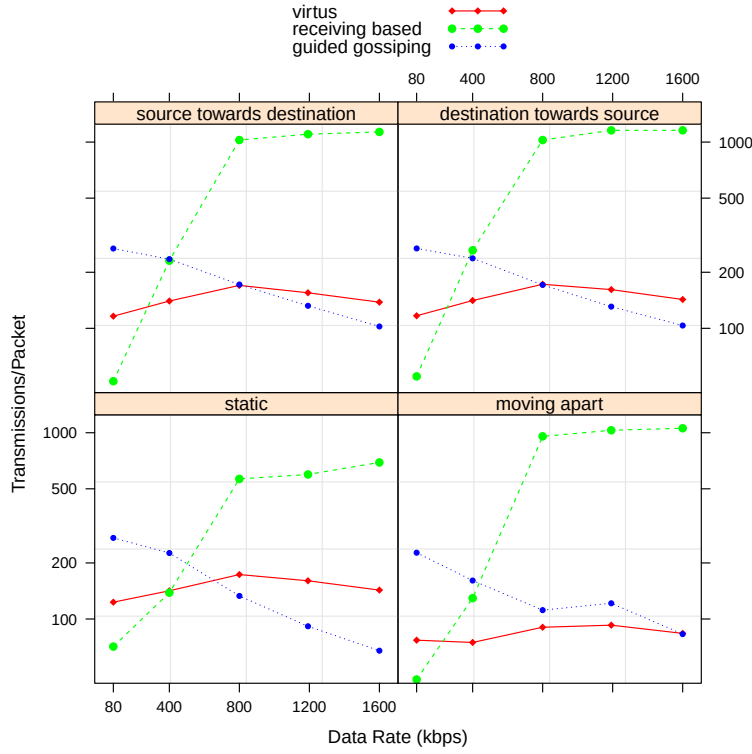


Figure 5.9: Cost

delay level achieved is within the quality of service requirements of real-time video streaming. Although the amount of resources used is not ideal, it is still reasonable and within feasible real life scenarios.

5.5 Related work

There is a wide variety of unicast communication protocols tailored for mobile ad hoc networks. Some of them are proactive such as DSDV and OLSR, reactive like AODV and DSR, geographic like GPSR, and hybrid geographic [Lee and Gerla, 2010]. Those protocols cannot be directly used in VANETs because they were not designed taking into consideration such high dynamic topology.

Due to the high mobility and intermittent connectivity, geographic approaches are preferred [Lee and Gerla, 2010; Wahid et al., 2010] over proactive and reactive approaches in VANETs scenarios. Wahid et al. [2010] survey

some geographic unicast protocols for VANETs and classify those protocols as: (i) greedy, (ii) opportunistic, and (iii) trajectory based. The greedy strategy is the most common approach, where nodes forward the packets to their farthest neighbors towards the destination. Opportunistic strategies employ the carry-forward technique in order to opportunistically deliver the data to the destination. Finally, trajectory-based strategies use a digital map to calculate possible paths to the destination and deliver the data through nodes along one or more of those paths.

Opportunistic strategies are not considered in this work because they tend to present a high delay due to the carry-forward process and do not meet strict requirements for video applications. Therefore, we are more interested in greedy strategies or, possibly, an hybrid approach where nodes greedily forward packets along the paths calculated by trajectory strategies.

GPSR is a well-known greedy geographic unicast protocol tailored for MANETs. It assumes that all nodes know their own location and of the destination as well. When a packet reaches a region that greedy forwarding leads to a local maximum and the process is interrupted, GPSR recovers by routing around the perimeter of the region. As soon as possible, the greedy forwarding process is reestablished. GPSR requires the knowledge of the immediate neighbors, and thus, all nodes broadcast beacon packets with their positions.

Movement prediction routing (MOPR) [Menouar et al., 2007a] employs a mechanism to predict vehicle's movement in order to create more stable connections. Those stable connections are used to decrease the number of path recovery processes needed to handle broken connections due to topology changes. To do so, they calculate the link stability metric based on the prediction of the vehicle's movements. The authors extended the AODV protocol to incorporate this feature. In a second work [Menouar et al., 2007b], MORP was integrated to GPSR protocol, and thus, features a greedy geographic unicast scheme that uses the link stability concept. This is one of the first works to use the link stability idea in order to choose the best relay for unicast communication. MOPR employs an unrealistic motion model (constant velocity motion model [Schubert et al., 2008]) that can lead to estimation errors. Moreover, MORP chooses the relay as the node that presents the highest link stability metric. This such assumption can lead to

an excessive number of hops along the communication path. Another problem is that both versions need to maintain an updated neighbors table that is a hard task to be performed in a highly dynamic network.

Distance Based Relay Selection (DBRS) Kim et al. [2007] is a simple and efficient strategy used to disseminate information over a network. Upon receiving a packet, the node holds it for a time interval that is proportional to the reciprocal of the distance to the transmitting node. Thus, nodes situated further to the transmitting node will be preferable to disseminate the information. Nodes that hear the transmission of the packet that is already scheduled cancel its transmission to avoid the broadcast storm problem. This approach is efficient in the way that it can handle the broadcast storm, but it's prone to two problems: (i) the delay can be high once there is no guarantee of the existence of nodes close to the communication radius (the ones that will transmit with lowest delay), and (ii) the coverage can be low once nodes will cancel their transmission indiscriminately upon hearing the transmission of the same packet.

Apart from other works in the literature, CTTDD adopts a receiving-based approach that does not require any kind of neighbors table nor the use of beacons. Instead, CTTDD includes on the data packet header all the information it needs to perform the data forwarding process. CTTDD uses a data forwarding process similar to DBRS, however, it uses a Bayesian state estimation approach in order to track and predict the vehicle's state (position, velocity and bearing). This prediction is employed to make channel reservation and decrease the delay. There are three main reasons that make CTTDD stand out from others. First, it is based on the receiver based forwarding scheme, which provides greater adaptability for the highly mobile nature of VANETs. Second, it incorporates link durability into greedy strategy in order to support continuous video transmission. Third, it proposes a forwarding zone, reservation time and a multiple forwarders prevention method to alleviate the drawbacks of a receiver based forwarding scheme.

5.6 Chapter remarks

In this Chapter we presented and discussed the main aspects, and new challenges for cooperative target tracking in vehicular ad hoc network context. We divided

the problem into four components: motion models, measurement models, data association, and filtering. The main aspects of each component were discussed according to the VANET scenarios. For example, expressive models to represent other possible targets for urban scenarios such as animals, pedestrian, and bicycle, are seldom found in the literature.

When multiple sensors are used, or when both autonomous and cooperative data are considered, the data association problem represents one of the most important challenges due to the density and clutter characteristics of urban scenarios. Sensors capable of capturing other characteristics to identify the target are desirable to facilitate the data association algorithms. For instance, video cameras can capture targets' color and length, narrowing the search space.

The choice of the most appropriate filter is always a challenging task for all tracking applications. In a VANET, energy and computational power constraints are not relevant issues, thus, techniques such as particle filter are feasible candidates. The filter must also be able to cope with missing data either due to packet loss or high delays. In some cases, the application also requires hard time limits, and the filter must be able to deal with them. The performance of those filters should be thoroughly evaluated under all those conditions.

Finally we present VIRTUS, a unicast data dissemination video scheme based on CTTDD approach. VIRTUS is resilient to topology changes by using nodes locations and estimates of their future positions in order to achieve delay levels close to the baseline lower bound. It requires a slightly high number of transmissions but it scales to data rates up to 800 kbps in a channel of 2 Mbps. Our future directions are twofold into trying to achieve even lower frame loss rates and reducing the number of transmissions. We intend to use density-based mechanisms that might prevent the number of collisions at high data rates. Besides that, we strongly believe that network coding[Sengupta et al., 2010] is able to dramatically reduce the frame loss when it is already under 15%. Furthermore, we want to evaluate some heuristics to select relaying nodes that lead to shorter paths with the same stability.

Final remarks

“There is no real ending. It’s just the place where you stop the story”

Frank Herbert

This chapter presents the final thoughts about this thesis. In Section 6.1, we reinforce the achieved contributions while we provide our view about future research directions that can follow from this work. Section 6.2 shows the list of publications we achieved during this work.

6.1 Conclusions and outlook

The main objective of this thesis was the investigation of topological features in the design and operation of wireless ad hoc networks and the proposal of novel algorithms that improve the network performance by taking advantage of the topology awareness . We were also interested in estimating useful topological features and their application in the design of network protocols. This thesis advanced the state-of-the-art by proposing some solutions to deal with the topology

problem of ad hoc networks. This is mainly because the solutions are problem-driven and there is no general way to solve this problem. We started our work by investigating different kinds of wireless sensor network topologies because we seldom find a study that considers any kind of topology different from the uniformly random placement of nodes. We came up with M^2P^2 , an expressive and general topology model for wireless sensor networks that is able to represent a wide variety of topologies. This study allowed us to propose the sink betweenness metric, i.e., a variation of the betweenness. We show that sink betweenness outperforms the latter to represent the characteristics and traffic patterns of wireless sensor networks. We also proposed a routing algorithm that benefits from the sink betweenness awareness to improve the energy balancing, and thus, improve the lifetime of a wireless sensor network.

We started the study of mobile networks and we observed that, for this case, the challenges are much greater. As the topology changes rapidly, it is much harder to find topology-related metrics or models that can be used by network protocols to improve their performance. We observed that localization plays a key role, thus, motivated by the possibility of offloading GPS processing to the cloud, we propose a novel embedded GPS sensing approach called CO-GPS. By using a coarse-time navigation technique and leveraging information that is already available on the web, such as satellite ephemeris and velocities and Earth elevations, we show that 2 ms of raw GPS signals is enough to reach a location fix. By averaging multiple such short chunks over a short period of time, CO-GPS can achieve 45 m location accuracy using 10 ms of raw data (40 kB). Without the need to do satellite acquisition, tracking and decoding, the GPS receiver can be simple and aggressively duty cycled. This solution is well adapted to mobile ad hoc networks where devices are prone to deplete their batteries quickly when using GPS.

Finally, we moved our studies to vehicular networks. In order to cope with the rapid changes in the highly dynamic topologies of these networks, we proposed a cooperative target tracking module, where vehicles are able to track the topology changes of a set of nodes of interest. To show its feasibility, we proposed CTTDD, a data dissemination algorithm that takes advantage of the target tracking module to alleviate the drawbacks of geographic greedy forwarding

mechanism. On top of it, we developed VIRTUS, a unicast video dissemination scheme.

This work presents several possibilities for future research. For instance, M^2P^2 and the sink betweenness are able to represent static networks only. The development of models and metrics for mobile networks is a challenging task. For example, Peres et al. [2011] have been successfully modeling mobile networks in a different perspective. Adapting M^2P^2 and the proposal of topological metrics to improve the topology awareness of mobile networks are still open and challenging subjects. The SBet metric can be further explored, for example, we applied it to develop a data collection protocol that improves the data fusion [Oliveira et al., 2010] and obtained promising results. However, there are different network functions and applications that can benefit from the SBet metric and wasn't explored in this work. For instance, the quantification of the relationship between the SBet and fault-tolerance, latency and network lifetime properties are still open to be investigated. The SBet can also be applied to topology control schemes to diminish the possibility of interference on nodes that were attractively deployed around H-sensors and the sink. It means that there are plenty of room to conduct further investigation regarding both the M^2P^2 and the SBet metric.

Localization is a hot topic and is still demanding lots of attention from the academic point-of-view. Although CO-GPS represents a great improvement on the energy consumption of GPS-based location tags, it is still challenging to localize mobile nodes due to GPS outages and indoors nodes. Thus, a collaborative localization system based on CO-GPS, where mobile nodes may help other nodes that are facing difficulties to estimate their position is a promising approach. CO-GPS can also be embedded into tiny devices such as environmental sensors. In this case, we can also exploit various compression techniques, especially those based on compressive sensing principles, to further reduce the storage requirements. On the web service side, we can explore parallelization of CO-GPS processing to improve speed.

In this work, the cooperative target tracking module was exploit to develop CTTDD to improve the data dissemination in networks that change the topology quickly. We can also explore the cooperative target tracking to investigate how to improve the performance of other network functions like fault tolerance,

medium access and congestion control, for instance. We can also investigate the use of the cooperative target tracking module as main role, for example in collision avoidance systems. This is a promising subject that arouses the interest of intelligent transportation systems providers.

In VIRTUS context, we are working on new approaches, where in collaboration with a group in the University of Ottawa, we are including network coding capabilities to decrease the frame loss and improve performance.

6.2 Publications

In the following sections we list the publications achieved during this work. The list is divided into four categories: (i) periodical papers, (ii) conference papers, (iii) under submission, and (iv) short course. Papers that start with a (☆) mark are related to a direct contribution of this work, while other papers indicate collaboration.

6.2.1 Periodicals

Nakamura, E. F., Ramos, H. S., Villas, L. A., de Oliveira, H. A., de Aquino, A. L., and Loureiro, A. A. F. (2009). A reactive role assignment for data routing in event-based wireless sensor networks. *Computer Networks*, 53(12):1980--1996.

Frery, A. C., Ramos, H. S., Alencar-Neto, J., Nakamura, E. F., and Loureiro, A. A. F. (2010). Data Driven Performance Evaluation of Wireless Sensor Networks. *Sensors (Basel)*, 10(3):2150--2168.

☆ Ramos, H. S., Boukerche, A., Pazzi, R. W., Frery, A. C., and Loureiro, A. A. (2012). Cooperative Target Tracking in Vehicular Sensor Networks. *Wireless Communications Magazine*, to appear.

Villas, L., Boukerche, A., Ramos, H. S., de Oliveira, H. A., Araujo, R., and Loureiro, A. A. (2012). DRINA: A Lightweight and Reliable Routing Approach for in-Network Aggregation in Wireless Sensor Networks. *IEEE Transactions on Computers* (PrePrint).

6.2.2 Conferences

- Frery, A. C., Ramos, H. S., Alencar-Neto, J., and Nakamura, E. F. (2008). Error estimation in wireless sensor networks. In *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, volume 3, page 1923, New York, New York, USA. ACM Press.
- Villas, L. A., Ramos, H. S., and Loureiro, A. A. F. (2009). Um algoritmo de Roteamento Ciente de Agregação de Dados para Redes de Sensores sem Fio. In *SBRC'09 - Simpósio Brasileiro de Redes de Computadores*, pages 233--246.
- Cunha, F. D., Ramos, H. S., and Loureiro, A. A. F. (2010). Roteamento Oportunístico em Redes de Sensores Tolerantes a Atrasos e Desconexões. In *SBRC'10 - Simpósio Brasileiro de Redes de Computadores*, pages 423--436.
- ☆ Oliveira, E. M. R., Ramos, H. S., and Loureiro, A. A. F. (2010). Centrality-based routing for Wireless Sensor Networks. In *2010 IFIP Wireless Days*, pages 1--5. IEEE.
- ☆ Ramos, H. S., Almiron, M. G., Frery, A. C., Nakamura, E. F., and Loureiro, A. A. F. (2010). Node Deployment by Stochastic Point Processes in Wireless Sensor Networks. In *ITS' 2010: Proceedings of the International Telecommunications Symposium*, Manaus, Brazil.
- ☆ Ramos, H. S., Zhang, T., Liu, J., Priyantha, N. B., and Kansal, A. (2011). LEAP: a low energy assisted GPS for trajectory-based services. In *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp'11*, pages 335---344, New York, New York, USA. ACM Press.
- ☆ Ramos, H. S., Guidoni, D., Boukerche, A., Nakamura, E. F., Frery, A. C., and Loureiro, A. A. F. (2011). Topology-related modeling and characterization of wireless sensor networks. In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks - PE-WASUN'11*, page 33, New York, New York, USA. ACM Press.
- ☆ Ramos, H. S., Oliveira, E. M. R., Boukerche, A., and Loureiro, A. A. F. (2012). Characterization and Mitigation of the Energy Hole Problem of Many-to-One

Communication in Wireless Sensor Networks. In *Proceeding of the IEEE International Conference on Computing, Networking and Communications - ICNC'12*.

☆ Rezende, C. G., Ramos, H. S., Pazzi, R. W., Boukerche, A., Frery, A. C., and Loureiro, A. A. F. (2012). VIRTUS: A resilient location-aware video unicast scheme for vehicular networks. In *ICC '12: IEEE International Conference on Communications*, pages 1--5, Ottawa, Canada. IEEE Comput. Soc.

Wang, R., Rezende, C. G., Ramos, H. S., Pazzi, R. W., Boukerche, A., and Loureiro, A. A. F. (2012). LIAITHON: A location-aware multipath video streaming scheme for urban vehicular. In *ISCC'2012: 17th IEEE Symposium on Computers and Communications*, pages 1--5.

☆ Liu, J., Hart, T., Priyantha, N. B., Ramos, H. S., Loureiro, A. A. F., and Wang, Q. (2012). Energy-Efficient GPS Sensing with Cloud Offloading. In *SenSys'12 10th ACM Conference on Embedded Network Sensor Systems*, pages 1--14, Toronto, CA.

Villas, L., Ramos, H. S., Boukerche, A., Guidoni, D., Araújo, R., and Loureiro, A. A. F. (2012). An Efficient and Robust Data Dissemination Protocol for Vehicular Ad Hoc Networks. In *Proceedings of the 9th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks - PE-WASUN '12*, pages 1--5, Paphos, Cyprus Island.

6.2.3 Under Submission

☆ Ramos, H. S., Boukerche, A., Frery, A. C., Oliveira, E. M. R., and Loureiro, A. A. F. (2012). Energy-Aware Topology Modeling of Wireless Sensor Networks. *IEEE Transactions on Computers*, (under submission).

☆ Ramos, H. S., Oliveira, E. M. R., Boukerche, A., Frery, A. C., and Loureiro, A. A. F. (2012). Topology-Aware Design of Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, (January):(under submission).

Almulla, M., Wang, R., Rezende, C. G., Ramos, H. S., Pazzi, R. W., Boukerche, A., and Loureiro, A. A. F. (2012). Towards High Quality Video Streaming over

Urban Vehicular Networks Using A Location-Aware Multipath Scheme. *IEEE Transactions on Computers*, (under submission).

6.2.4 Short Course

Loureiro, A. A. F., Frery, A. C., Mini, R., Aquino, A. L. L., Ramos, H. S., and Almiron, M. G. (2010). *Redes Complexas na Modelagem de Redes de Computadores*. Minicursos do SBRC'10 - Simpósio Brasileiro de Redes de Computadores.

6.2.5 Awards

2012 – Best Paper Award at the ACM Conference on Embedded Networked Sensor Systems (SenSys'12), paper entitled 'Energy-Efficient GPS Sensing with Cloud Offloading', ACM.

2010 – Microsoft Research PhD Fellowship Award, Microsoft Research.



Bibliography

- Akopian, D. and Syrjarinne, J. (2009). A Fast Positioning Method Without Navigation Data Decoding for Assisted GPS Receivers. *IEEE Transactions on Vehicular Technology*, 58(8):4640--4645.
- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393--422.
- Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6--28.
- Almiron, M., Almeida, E. S., and Miranda, M. (2009). The Reliability of Statistical Functions in Four Software Packages Freely used in Numerical Computation. *Brazilian Journal of Probability and Statistics*, 23(2):107--119.
- Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537--568.
- Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174--188.

- Aydos, C., Hengst, B., and Uther, W. (2009). Kalman filter process models for urban vehicle tracking. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1--8.
- Baccelli, F. (2009). Stochastic Geometry and Wireless Networks: Volume I Theory. *Foundations and Trends in Networking*, 3(3-4):249--449.
- Baddeley, A. (2006). Spatial point processes and their application. In Weil, W., editor, *Stochastic Geometry*, volume 1892 of *Lecture Notes in Mathematics*, pages 1--75. Springer, Berlin.
- Baheti, R. (1986). Efficient Approximation of Kalman Filter for Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(1):8--14.
- Bavelas, A. (1948). A mathematical model for Group Structure. *Human Organizations*, 7:16--30.
- Beauchamp, M. A. (1965). An improved index of centrality. *Behavioral Science*, 10:161--163.
- Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113--120.
- Bonacich, P. (1987). Power and Centrality: A Family of Measures. *The American Journal of Sociology*, 92(5):1170--1182.
- Bonett, D. G. and Wright, T. A. (2000). Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometric*, 65(1):23--28.
- Borre, K., Akos, D. M., Bertelsen, N., Rinder, P., and Jensen, S. H. (2006). *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*. Birkhäuser.
- Boukerche, A., Oliveira, H., Nakamura, E., and Loureiro, A. (2008). Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems. *Computer Communications*, 31(12):2838--2849.

- Boukerche, A., Rezende, C., and Pazzi, R. (2009). Improving neighbor localization in vehicular ad hoc networks to avoid overhead from periodic messages. *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1-6.
- Boulis, A. (2009). Castalia: A simulator for Wireless Sensor Networks. Web Page: <http://castalia.npc.nicta.com.au/>. Last visited in 2009, september.
- Brin, S. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Computer Networks and ISDN Systems*, volume 30, pages 107--117.
- Buhren, M. and Yang, B. (2007). A Global Motion Model for Target Tracking in Automotive Applications. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages II--313--II--316. IEEE.
- Burt, R. S. (2004). Structural Holes and Good Ideas. *American Journal of Sociology*, 110(2):349--399.
- Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146(1):2.
- Celebi, H. and Arslan, H. (2007). Utilization of Location Information in Cognitive Wireless Networks. *IEEE Wireless Communications*, 14(4):6--13.
- Cui, J.-H., Kong, J., Gerla, M., and Zhou, S. (2006). The challenges of building mobile underwater wireless networks for aquatic applications. *IEEE Network*, 20(3):12--18.
- Culler, D., Estrin, D., and Srivastava, M. (2004). Overview of Sensor Networks. *IEEE Computer*, 37(8):41--49.
- David L. Mills (2006). *Computer Network Time Synchronization: The Network Time Protocol*. Taylor & Francis, 1 edition.
- Diggelen, F. (2007). GNSS Accuracy: Lies, Damn Lies, and Statistics. *GPS World*, 18(1):126---33.
- Dolev, S., Elovici, Y., and Puzis, R. (2010). Routing betweenness centrality. *Journal of the ACM*, 57(4):1--27.

- Ebrahimi, N., Soofi, E. S., and Soyer, R. (2010). Information Measures in Perspective. *International Statistical Review*, pages no--no.
- Erdős, P. and Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290--297.
- Fernandes, P. and Nunes, U. (2007). Vehicle Communication: a short survey. *IADIS Telecommunications, Networks and Systems, Lisboa, Portugal*, 1556(Jones 2005):134--138.
- Forouzan, B. A. (2007). *Data Communications and Networking*. McGraw-Hill Higher Education, 4th edition.
- Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35--41.
- Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215--239.
- Frery, A. C., Ramos, H. S., Alencar-Neto, J., Nakamura, E. F., and Loureiro, A. A. F. (2010). Data Driven Performance Evaluation of Wireless Sensor Networks. *Sensors (Basel)*, 10(3):2150--2168.
- Goh, K.-I., Kahng, B., and Kim, D. (2001). Universal Behavior of Load Distribution in Scale-Free Networks. *Physical Review Letters*, 87(27).
- Guan, L. (2008). Data Gathering Algorithm Based Load Balance for Wireless Sensor Networks. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, pages 1--5. IEEE.
- Guidoni, D. L., Mini, R. A. F., and Loureiro, A. A. F. (2008). On the design of heterogeneous sensor networks based on small world concepts. In *MSWiM '08: Proceedings of the 11th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 309--314. ACM.
- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425--437.

- Haenggi, M., Andrews, J., Baccelli, F., Dousse, O., and Franceschetti, M. (2009). Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(7):1029--1046.
- Hage, P. and Harary, F. (1995). Eccentricity and centrality in networks. *Social Networks*, 17(1):57--63.
- Hedetniemi, S. M., Hedetniemi, S. T., and Liestman, A. L. (1988). A Survey of Gossiping and Broadcasting in Communication Networks. *Networks: An International Journal*, 18(4):319--349.
- Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660--670.
- Helmy, A. (2003). Small Worlds in Wireless Networks. *IEEE Communications Letters*, 7(10):490--492.
- Hoydis, J., Petrova, M., and Maehoenen, P. (2009). The Effects of Topology on the Local Throughput of Ad Hoc Networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):337--347.
- Hue, C., Le Cadre, J.-P., and Perez, P. (2002). Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2):309--325.
- Iqbal, U., Georgy, J., Korenberg, M. J., and Nouredin, A. (2010). Augmenting Kalman Filtering with Parallel Cascade Identification for Improved 2D Land Vehicle Navigation. In *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1--5. IEEE.
- Jiang, D., Taliwal, V., Meier, A., Holfelder, W., and Herrtwich, R. (2006). Design of 5.9 ghz dsrc-based vehicular safety communication. *IEEE Wireless Communications*, 13(5):36--43.
- Kaplan, E. D. and Hegarty, C. J. (2005). *Understanding GPS: Principles and Applications, Second Edition*. Artech House Publishers, 2nd edition.

- Kiess, W. and Mauve, M. (2007). A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks*, 5(3):324--339.
- Kim, T.-h., Adeli, H., Choi, S.-Y., Kim, J.-S., Han, S.-J., Choi, J.-H., Rim, K.-W., and Lee, J.-H. (2010). Dynamic Routing for Mitigating the Energy Hole Based on Heuristic Mobile Sink in Wireless Sensor Networks. In *Advances in Computer Science and Information Technology*, volume 6059 of *Lecture Notes in Computer Science*, pages 159--174, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Kim, T.-H., Hong, W.-K., Kim, H.-C., and Lee, Y.-D. (2007). An effective data dissemination in vehicular ad-hoc network. In *Data Dissemination Techniques in Vehicular Ad Hoc Network*, volume 5200 of *Lecture Notes in Computer Science*, pages 295--304, Berlin/Heidelberg. Springer.
- Kirkpatrick, S., Gelatt, C. D., and Vechhi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598):671--680.
- Klaue, J., Rathke, B., and Wolisz, A. (2003). Evalvid – a framework for video transmission and quality evaluation. In Kemper, P. and Sanders, W., editors, *Computer Performance Evaluation. Modelling Techniques and Tools*, volume 2794 of *Lecture Notes in Computer Science*, pages 255--272. Springer Berlin / Heidelberg.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604--632.
- Koschützki, D., Lehmann, K. A., Peeters, L., Richter, S., Podehl, D. T., and Zlotowski, O. (2005). *Centrality Indices*, volume 3418/2005 of *Lecture Notes in Computer Science: Network Analysis*, chapter Part I: El, pages 16--61. Springer.
- Krishnamachari, B., Estrin, D., and Wicker, S. B. (2002). The Impact of Data Aggregation in Wireless Sensor Networks. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575--578, Washington, DC, USA. IEEE Computer Society.
- Leavitt, H. J. (1951). Some effects of communication patterns on group performance. *Journal of Abnormal and Social Psychology*, 46:38--50.

- Lee, U. and Gerla, M. (2010). A survey of urban vehicular sensing platforms. *Computer Networks*, 54(4):527--544.
- Leu, F. F.-Y., Wu, W.-C. W., Wen-Chin, and Huang, H.-W. (2009). A Routing Scheme with Localized Movement in Event-Driven Wireless Sensor Networks. In *Proceedings of the Joint International Conferences on Advances in Data and Web Management, APWeb/WAIM '09*, pages 576--583, Berlin, Heidelberg. Springer-Verlag.
- Li, J. and Mohapatra, P. (2007). Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive and Mobile Computing*, 3(3):233--254.
- Li, X. R. and Jilkov, V. P. (2001). Survey of maneuvering target tracking: III. Measurement models. In *Proceedings of SPIE*, pages 423--446. Spie.
- Lin, K., Kansal, A., Lymberopoulos, D., and Zhao, F. (2010). Energy-accuracy trade-off for continuous mobile device location. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 285--298, New York, NY, USA. ACM.
- Liu, A.-F., Wu, X.-Y., Chen, Z.-G., and Gui, W.-H. (2010). Research on the energy hole problem based on unequal cluster-radius for wireless sensor networks. *Computer Communications*, 33(3):302--321.
- Liu, J., Chu, M., and Reich, J. (2007). Multitarget Tracking in Distributed Sensor Networks. *IEEE Signal Processing Magazine*, 24(3):36--46.
- Liu, J., Hart, T., Priyantha, N. B., Ramos, H. S., Loureiro, A. A., and Wang, Q. (2012). Energy-Efficient GPS Sensing with Cloud Offloading. In *SenSys'12 10th ACM Conference on Embedded Network Sensor Systems*, pages 1--13, Toronto, CA.
- Liu, J., Reich, J., and Zhao, F. (2003). Collaborative In-Network Processing for Target Tracking. *EURASIP Journal on Advances in Signal Processing*, 2003(4):378--391.

- Lu, H., Zhang, S., Liu, X., and Lin, X. (2010). Vehicle Tracking Using Particle Filter in Wi-Fi Network. In *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1--5. IEEE.
- Luciano, Rodrigues, F. A., Travieso, G., Boas, V. P. R., Costa, L. d. F., and Villas Boas, P. R. (2007). Characterization of complex networks: a survey of measurements. *Advances in Physics*, 56(0001-8732):167--242.
- Machado, M., Goussevskaia, O., Mini, R., Rezende, C., Loureiro, A., Mateus, G., and Nogueira, J. (2005). Data dissemination in autonomic wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(12):2305--2319.
- Mandala, D., Dai, F., Du, X., and You, C. (2006). Load Balance and Energy Efficient Data Gathering in Wireless Sensor Networks. In *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 586--591. IEEE.
- Menouar, H., Lenardi, M., and Filali, F. (2007a). Improving Proactive Routing in VANETs with the MOPR Movement Prediction Framework. In *2007 7th International Conference on ITS Telecommunications*, pages 1--6. IEEE.
- Menouar, H., Lenardi, M., and Filali, F. (2007b). Movement Prediction-Based Routing (MOPR) Concept for Position-Based Routing in Vehicular Networks. In *2007 IEEE 66th Vehicular Technology Conference*, pages 2101--2105. IEEE.
- Misra, P. and Enge, P. (2006). *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press.
- Mohapatra, P. (2005). An analytical model for the energy hole problem in many-to-one sensor networks. In *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, volume 4, pages 2721--2725. IEEE.
- Nakamura, E. F., de Oliveira, H. A. B. F., Pontello, L. F., and Loureiro, A. A. F. (2006). On Demand Role Assignment for Event-Detection in Sensor Networks. In *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pages 941--947, Washington, DC, USA. IEEE.

- Nakamura, E. F., Loureiro, A. A. F., and Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3):9.
- Nakamura, E. F., Ramos, H. S., Villas, L. A., de Oliveira, H. A., de Aquino, A. L., and Loureiro, A. A. F. (2009). A reactive role assignment for data routing in event-based wireless sensor networks. *Computer Networks*, 53(12):1980--1996.
- National Institute of Standards and Technology (2011). *NIST time and frequency radio stations: WWW, WWVH, and WWVB*. Books LLC, Reference Series.
- Newman, M. (2003). The Structure and Function of Complex Networks. *SIAM Review*, 45:167--256.
- Niculescu, D. and Nath, B. (2003). Trajectory based forwarding and its applications. In Johnson, D. B., Joseph, A. D., and Vaidya, N. H., editors, *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, MOBICOM*, pages 260--272, San Diego, CA, USA. ACM.
- Nordio, A., Chiasserini, C., and Viterbo, E. (2007). The impact of quasi-equally spaced sensor layouts on field reconstruction. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 274--282, New York, NY, USA. ACM.
- Olariu, S. and Stojmenovic, I. (2006). Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1--12. IEEE.
- Oliveira, E. M. (2010). *Algoritmo de Roteamento Baseado em Características Topológicas para Redes de Sensores sem Fio*. Mestrado, Universidade Federal de Minas Gerais.
- Oliveira, E. M. R., Ramos, H. S., and Loureiro, A. A. F. (2010). Centrality-based routing for Wireless Sensor Networks. In *2010 IFIP Wireless Days*, pages 1--5. IEEE.

- Paek, J., Kim, J., and Govindan, R. (2010). Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 299, New York, New York, USA. ACM Press.
- Parsa, M. and Garcia-Luna-Aceves, J. (1998). An iterative algorithm for delay-constrained minimum-cost multicasting. *IEEE/ACM Transactions on Networking*, 6(4):461--474.
- Peres, Y., Sinclair, A., Sousi, P., and Stauffer, A. (2011). Mobile Geometric Graphs: Detection, Coverage and Percolation. In *SODA '11: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 412--428, San Francisco, California.
- Perillo, M. and Heinzelman, W. (2009). An Integrated Approach to Sensor Role Selection. *IEEE Transactions on Mobile Computing*, 8(5):709--720.
- R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ramanathan, R. and Redi, J. (2002). A brief overview of ad hoc networks: challenges and directions. *IEEE Communications Magazine*, 40(5):20--22.
- Ramos, H. S., Guidoni, D., Boukerche, A., Nakamura, E. F., Frery, A. C., and Loureiro, A. A. (2011a). Topology-related modeling and characterization of wireless sensor networks. In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks - PE-WASUN'11*, page 33, New York, New York, USA. ACM Press.
- Ramos, H. S., Oliveira, E. M. R., Boukerche, A., and Loureiro, A. A. (2012). Characterization and Mitigation of the Energy Hole Problem of Many-to-One Communication in Wireless Sensor Networks. In *ICNC'12: Proceeding of the IEEE International Conference on Computing, Networking and Communications*.
- Ramos, H. S., Zhang, T., Liu, J., Priyantha, N. B., and Kansal, A. (2011b). LEAP: a low energy assisted GPS for trajectory-based services. In *Proceedings of the 13th*

- international conference on Ubiquitous computing - UbiComp'11*, pages 335--344, New York, New York, USA. ACM Press.
- Rawicz, P. L., Kalata, P. R., Murphy, K. M., and Chmielewski, T. A. (2003). Explicit formulas for two state Kalman, H-2 and H-infinity target tracking filters. *Ieee Transactions On Aerospace And Electronic Systems*, 39(1):53--69.
- Rekleitis, I., Dudek, G., and Milios, E. (2003). Probabilistic cooperative localization and mapping in practice. In *IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pages 1907--1912. IEEE.
- Ren, F., Zhang, J., He, T., Lin, C., and Ren, S. K. D. (2011). EBRP: Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(12):2108--2125.
- Rezende, C. G., Ramos, H. S., Pazzi, R. W., Boukerche, A., Frery, A. C., and Loureiro, A. A. (2012). VIRTUS: A resilient location-aware video unicast scheme for vehicular networks. In *ICC'12: IEEE International Conference on Communications*, pages 1--5, Ottawa, Canada. IEEE Comput. Soc.
- Robert, C. P. and Casella, G. (2000). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, New York.
- Rockl, M., Strang, T., and Kranz, M. (2008). V2V Communications in Automotive Multi-Sensor Multi-Target Tracking. In *2008 IEEE 68th Vehicular Technology Conference*, pages 1--5. IEEE.
- Rong-Li, X. and Jilkov, V. (2003). Survey of maneuvering target tracking. Part I: dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333--1364.
- Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581--603.
- Schubert, R., Richter, E., and Wanielik, G. (2008). Comparison and evaluation of advanced motion models for vehicle tracking. In *11th International Conference on Information Fusion*, pages 1--6. IEEE.

- Sengupta, S., Rayanchu, S., and Banerjee, S. (2010). Network Coding-Aware Routing in Wireless Networks. *IEEE/ACM Transactions on Networking*, 18(4):1158--1170.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.
- Sinalgo (2008). Simulator for Network Algorithms.
- Sirola, N. (2001). A Method for GPS Positioning Without Current Navigation Data. Master's thesis, Department of Electrical Engineering. Tampere University of Technology.
- Smith, D. and Singh, S. (2006). Approaches to Multisensor Data Fusion in Target Tracking: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1696--1710.
- Somasundara, A. A., Kansal, A., Jea, D., Estrin, D., and Srivastava, M. B. (2006). Controllably Mobile Infrastructure for Low Energy Embedded Networks. *IEEE Transactions on Mobile Computing*, 5(8):958--973.
- Song, C., Cao, J., Liu, M., Zheng, Y., Gong, H., and Chen, G. (2008). Mitigating energy holes based on transmission range adjustment in wireless sensor networks. In *Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, QShine '08, pages 32:1----32:7.
- Sun, G., Chen, J., Guo, W., and Liu, K. J. R. (2005). Signal Processing Techniques in Network-Aided Positioning. *IEEE Signal Processing Magazine*, pages 12--23.
- Syrjarinne, J. (2000). Time recovery through fusion of inaccurate network timing assistance with GPS measurements. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, volume 2, pages WED5/3 -- WED510 vol.2.
- Tselishchev, Y., Boulis, A., and Libman, L. (2010). Experiences and Lessons from Implementing a Wireless Sensor Network MAC Protocol in the Castalia Simu-

- lator. In *IEEE Wireless Communications & Networking Conference 2010 (WCNC 2010)*, Sydney/Australia.
- van Diggelen, F. (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, Boston/London.
- Wahid, A., Yoo, H., and Kim, D. (2010). Unicast geographic routing protocols for inter-vehicle communications. In *Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N '10*, page 17, New York, New York, USA. ACM Press.
- Wang, D. M., Xie, B., and Agrawal, D. P. (2008). Coverage and Lifetime Optimization of Wireless Sensor Networks with Gaussian Distribution. *IEEE Transactions on Mobile Computing*, 7(12):1444--1458.
- Wasserman, S. and Faust, K. (1994). *Social Network Analysis: methods and applications*. Cambridge University Press.
- Wu, X., Chen, G., and Das, S. K. (2008). Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution. *IEEE Transactions on Parallel and Distributed Systems*, 19(5):710--720.
- Xi, F. and Liu, Z. (2009). Small World Topology-Aware Geographic Routing in Wireless Sensor Networks. In *CMC '09: Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, pages 116--120, Washington, DC, USA. IEEE Computer Society.
- Yarvis, M. D., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y., and Singh, S. (2005). Exploiting heterogeneity in sensor networks. In *INFOCOM '05: Proceedings of the 24th IEEE International Conference on Computer Communications*, pages 878--890. IEEE.
- Yiwei, F., Wentian, C., and Huahai, Y. (2006). Identification of Important Actors in Edge-Weight Social Networks. In *2006 International Conference on Service Systems and Service Management*, volume 2, pages 1643--1647. IEEE.

- Yoon, S., Veerarittiphan, C., and Sichitiu, M. L. (2007). Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Transactions on Sensor Networks*, 3(2):8--es.
- Younis, M. and Akkaya, K. (2008). Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621--655.
- Younis, O. and Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366--379.
- Younis, O., Krunz, M., and Ramasubramanina, S. (2006). Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges. *IEEE Network*, 20(3):20--25.
- Yousefi, S., Mousavi, M., and Fathy, M. (2006). Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. In *2006 6th International Conference on ITS Telecommunications*, pages 761--766. IEEE.
- Zhang, Y., Nath, B., and Kang, J. (2007). TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks. *IEEE Transaction on Parallel and Distributed Systems*, 18(7):919--931.
- Zhao, Y., Wu, J., Li, F., and Lu, S. (2012). On Maximizing the Lifetime of Wireless Sensor Networks Using Virtual Backbone Scheduling. *IEEE Transactions on Parallel and Distributed Systems*, 23(8):1528--1535.
- Zhuang, Z., Kim, K.-H., and Singh, J. P. (2010). Improving energy efficiency of location sensing on smartphones. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 315--330, New York, NY, USA. ACM.
- Zuniga, M. and Krishnamachari, B. (2004). Analyzing the transitional region in low power wireless links. In *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, pages 517--526.

This document was prepared with $\text{\LaTeX}$.
The bibliography was managed with $\text{\BibTeX}$.
The font typeface is the Bitstream Charter (Math Design project).
The font size is 12 pt.
Pictures and plots were prepared with PGF/Tikz, R, and Microsoft Visio.

