Exploring Cross-Layer Dependencies in Congested Wireless Ad Hoc Networks

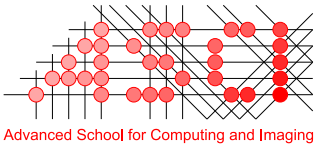
Ph.D. Thesis

Albana Gaba

VU University Amsterdam, 2014

vrije Universiteit amsterdam

This research was funded by the Vrije Universiteit Amsterdam.



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 316.

ISBN 978-90-5383-100-7

Cover design and layout by Dirk Vogt
Cover based on “Misty morning” by Ole Husby (cc-by-sa-2.0)
Printed by IPSKAMP Drukkers B.V.

VRIJE UNIVERSITEIT

Exploring Cross-Layer Dependencies in Congested Wireless Ad Hoc Networks

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op dinsdag 9 december 2014 om 11.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

ALBANA GABA

geboren te Vlorë, Albanië

promotor: prof.dr.ir. M.R. van Steen
copromotor: dr. S. Voulgaris

Mamit dhe babit

Acknowledgments

I am grateful to a lot of people who, in different ways, have supported me throughout my PhD endeavor. I take the opportunity to thank them in these pages.

First and foremost, I would like to thank my supervisor, Maarten van Steen, for giving me the opportunity to work on this interesting project. With his enthusiasm and passion for research, he has been a true source of inspiration for me. He has provided steady support and encouragement even when things looked not so bright. Above all, I am grateful to Maarten for patiently teaching me how to do research. Among other things, he taught me to always take a step back and never lose sight of the big picture.

I would like to thank Spyros Voulgaris, my co-supervisor, for being a good friend and a fantastic collaborator. He greatly helped me to improve my writing skills and shared with me his enthusiasm and knowledge about technical details.

I am especially grateful to my reading committee, Ciprian Dobre, Daniela Gavidia, Konrad Iwanicki and Rob van der Mei, for taking the time to read this dissertation and for their fruitful comments.

I owe a big thank you to a number of people that helped me conducting my research. To Rena Bakhshi, for doing her utmost to help, always with a big smile and great enthusiasm. When I asked her to proofread part of this dissertation, she didn't hesitate and gave me precious feedback. To Matt Dobson, who has greatly contributed by sharing his knowledge. To Konrad Iwanicki, for his extraordinary insight and practical directions with sensor networks. And to Daniela Gavidia, for sharing her expertise on gossiping protocols.

I was blessed to have some great office mates, who provided a fun work environment. Our conversations, supposed to fill little breaks, sometimes lasted hours. Thanks to Vivek Rai, a great tennis player with a cricket twist, for helping me move a couple of times. To Suhail Yousef, from whom I learned a lot about his country, Pakistan, and his culture. I have always admired his ideas and concrete efforts to contribute for his country. To Pieter Hijma for bringing some Dutch spirit in our office and help translating Dutch documents. And to Claudio Martella for the spontaneous, yet effective earworm tunes he dedicated to people around him, me included.

I am grateful to the many fellow researchers and professors of the Computer Systems group. I learned a lot from them in seminars, ASCI conferences, courses, social events and during the monthly borrels. The list is too

long to write it here, but some persons stand out among them. I am particularly thankful to Guillaume Pierre, Paolo Costa, and Jeff Napper for making my first years at the VU easy and fun.

One of the priceless things about my PhD years was the opportunity to meet a lot of fantastic people, some of which turned out to be great friends. I would like to thank, in particular, Guido Urdaneta for his steady support and encouragement, and for revising my dissertation in its early stages. Thank you Ana-Maria Oprescu, Corina Stratan, David van Moolenbroek, Philip Homburg, Daniela Remenska, Ben Gras, Bora Lushaj and Genc Tato for the good time we spend together and for always being there for me. Special thanks go to David for his precious help with the Dutch summary.

I consider myself very fortunate that I obtained a student accommodation in the Django building, especially because there, I found fantastic neighbors, some of which happened to be also colleagues. Thank you Cristiano Giuffrida and Laura Ferranti, Stefano Ortolani, Christian Roth, Kaveh Razavi, Jailan Heidar, Ismail El Helw and Raja Appuswamy for those many movie nights, dinners, and barbecues. And thanks to Stefano for sharing the printer.

I would not have started this adventure if Cecilia Mascolo had not recommended me to Maarten for the open position, and without the great support of my teachers at the University of Bologna. I would like to thank, in particular, Fabio Panzieri, Giorgia Lodi and Vittorio Ghini, who, with their supervision and dedication, instilled in me the passion for research, and taught me academic writing.

My deepest gratitude and love go to my parents for being my life teachers, wise advisors and unconditional supporters. Their love gives me the strength to pursue my dreams and carry on when the going gets tough. Special thanks to my sisters, who are also my best friends, for their love and affection.

I am very grateful to my love, my sunlight, Dirk, for the joy he brings to my life. I must thank him particularly for the continuous support he has provided throughout the long process of writing this dissertation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Advancements in hardware technologies have paved the way for the progressive miniaturization of electronic devices, at very low cost and with high efficiency in terms of battery consumption and computing capacity. The result of the miniaturization process is visible: we are surrounded more and more by wireless unobtrusive devices that can sense, compute and share data. Smart devices, such as wristbands for tracking daily activities, networked badges carried at conferences, shoe tags are already adopted by many. In addition, augmented reality eyewear and smartwatches are already making their way to the consumers.

Provided with various sensors, computing power and a low-power radio, these tiny devices, also known as on-body sensors, can collect a good deal of information about the people carrying them. Such information includes location [46; 47], mobility pattern [78], social interactions [31], daily physical activity, user profiling [73] or even identifying people based on their motion [20; 15].

Despite the significant progress in the field of on-body sensors, the range of operations they can perform is strongly limited due to their size. The embedded low-power antenna restricts the communication range typically to tens, at most hundreds of meters, whereas the computing capacity does not allow complex computations.

However, when considering these devices as a collaborative group, we may see a powerful wireless distributed system emerge. As a group, they may be able to offer real-time information on their surrounding environment, be able to provide insight in their own structure, or act as a common communication channel.

In this dissertation, we look only slightly ahead into the future by considering networks of hundreds or even thousands of wireless devices embedded into credit-card sized tickets to festivals, badges carried at conferences, or special clip-ons attached to shoes during running marathons, to name just a few. We envision a decentralized system where each person carries a wireless device, such as an on-body sensor, collectively forming an ad hoc network. By relaying messages opportunistically through the network people exchange useful information with each other.

While a plethora of applications have been enabled on small-scale ad hoc networks, there are a number of potential applications that could take advantage of large networks of sensors as we will see in the next section.

1.1 The Power of the Crowd

For many years now, the increase in frequency and size of public mass events has become an important problem. Such events are mainly organized across city streets and can attract millions of people. People's mobility can be highly impeded, whereas in more serious cases large-scale events have ended with several casualties, as happened at the Love Parade Duisburg in 2010 and in Mecca for a number of years. Overcrowding, followed by series of chain reactions in the crowd, can make people lose balance and fall to the ground. The very nature of a crowd and the emergent behavior that can take place in vast areas, such as parts of the city, make it very difficult to impose control through conventional means. For example, in the case of Duisburg it was reported that it was not possible to prevent overcrowding since people could access the event from various streets.

To make things worse, any form of communication in the crowd is almost impossible. Traditional means of communication such as cellular networks or other centralized solutions (e.g., WI-FI) have proven to be unable to scale when a great number of users concentrate in one place. Despite efforts to improve the throughput of current cellular networks [50], scaling the communications of such networks remains a major issue. Typical examples of this problem are the huge messaging delays on New Year's Eve, or the drop

in quality of service of cell phones in the neighborhood of a stadium during an important match.

With no possibility to see a few meters ahead and lacking communication means, it is hard for people in the crowd to know whether there is overcrowding or a serious situation hundreds of meters ahead. In contrast, from the perspective of computer networks, a massive presence of mobile entities covering a vast area can be considered an instrument for enabling communication among them in order to gain real-time information. This is the case of ad hoc networks of cars (VANETs[1]), people in a campus or monitoring of animals (Opportunistic Networks). In the same way, people in a large-scale event equipped with wireless devices can enable a mobile ad hoc network in order to share information about critical situations or communicate with each other.

Ad hoc networks typically consist of a collection of similar nodes that communicate with each other over wireless links without central control. Given that the communication range of each node is bound to tens or hundreds of meters, the information in these networks travels over multiple hops until it reaches the destination. Despite their wireless nature, ad hoc networks offer peer communication primitives similar to wired networks. Broadcast, multicast, and point-to-point primitives can be accomplished through multi-hop communications between nodes. Unlike wired networks, the links between nodes in ad hoc network may vary continuously due to their mobility or even unreliability of the connections.

Ad hoc networks are in principle inherently scalable. No centralized infrastructure is necessary to enable communication between nodes. As nodes join or leave the network, their impact is strictly local, in that it affects only their neighborhood. Therefore, such networks, unlike centralized solutions, can expand by simply adding nodes and are resilient to single points of failure. In fact, as nodes leave the network, their duties are normally taken over by other nodes in the neighborhood.

The high concentration of people in a crowd provides good connectivity for the network making it easier for messages to be propagated across the crowd. Moreover, the mobility of people can speed up the propagation of information [82]. To illustrate the qualities of crowd-based communications let us consider the following scenarios:

[1] Vehicular Ad Hoc Networks

Crowd control Researchers [6] observed that moments before a crowd disaster occurs, there is a particular dynamic pattern of the crowd defined as "crowd turbulence", in which uncontrolled shock waves threaten people to lose their balance and fall to the ground. Such dynamics have proven to be detected by networked on-body sensors worn by people, which are equipped with accelerometers [78].

A practical safety application includes people wearing on-body wireless sensors that are able to capture potentially hazardous situations, and propagate warnings across the entire crowd. Such real-time warnings can reach people far away from the place where the critical situation takes place and induce them to slow down or even not proceed any longer. Along these lines, providing general real-time information about the crowd, such as density or people's mobility, would help participants take informed decisions on the route to pick. As a result, participants' mobility may be eased by choosing less frequented routes, finding closest exit from the crowd and so on.

Group monitoring in a crowd A distributed wireless application may enable groups of participants (like family or friends) to keep in touch with each other in a large-scale public event. For example, consider a large number of people attending a festival, each carrying a wearable networked sensor such as an electronic badge. Each person belongs to one group and periodically exchanges messages with his or her group members by relaying them through the network formed by the crowd as a whole. Such a distributed application allows group members to monitor each other's presence over time.

These application examples exploit the co-presence of people in time and space to provide a means of communication between them, thus turning the crowd into a feature rather than an inconvenience.

1.2 The Problem

The problem we address in this dissertation is enabling decentralized communication between people in hot spots, such as amusement parks, festivals, and so on, through ad hoc networks. An important observation for the above mentioned applications is that to keep an updated view of the ever-changing state of the crowd, nodes have to insert fresh messages over time. A message initiated from a source is typically rebroadcast by neighboring nodes and extends outward, hop by hop, until either its destination or the entire network

is reached. Matters get complicated when all nodes in the network act as sources, implying they regularly insert new messages. Considering the large scale of the network, an immense traffic may be generated, which could lead to congestion of the network.

To deal with vast amounts of messages, it is fundamental to make appropriate use of the available resources by getting the maximum outcome out of them. One of the main issues is the utilization of the wireless medium. In a wireless environment this problem is particularly difficult due to the nature of wireless communication, including unreliable links, multi-hop interference and no fixed neighbors. Techniques valid for wired networks do not apply in the wireless case. Consider a carrier-sense technique adopted by nodes for accessing the shared wireless medium. Basically, before transmitting a packet nodes have to contend with each other in the neighborhood, typically by randomly picking a time within a time window for accessing the medium. Establishing the rate of medium access is challenging: when nodes attempt to transmit at a high pace, the chances of packet collisions increase, thus reducing the number of packets that are actually delivered. On the other hand, transmitting packets at a low pace can leave the channel under-utilized. The ideal rate of packet transmission is related to the density of the network and, as the topology may vary over time, deciding the most appropriate rate at each point in time is even more complicated.

Another key issue for large-scale mobile ad hoc networks is that traditional end-to-end routing is not feasible. Information cannot be transferred from one node to another through a predefined route. Due to the huge diameter of the network, the mobility of its nodes, and the inherent unreliability of wireless connections, we can essentially rely only on information dissemination using techniques, like gossiping, flooding, or (directional) random walks. Unfortunately, these flooding-based techniques can easily impose a huge demand on network resources.

Finally, communication among group members in the crowd implies the exchange of messages containing private data, like, for example, location information. Sharing confidential data over an untrusted medium, such as a wireless network, may leak information about the users to third parties. More specifically, adversaries can observe and manipulate the content of packets sent over the network. While encryption may address the confidentiality issues, it cannot protect against traffic analysis. The latter is known as a key issue, as it may expose the communication patterns among nodes (e.g.,who talks to whom), link nodes to messages they transmit, and so on. Such threats expose people to so-called targeted attacks. For example, by selectively drop-

ping packets, a targeted group may not receive any fresh messages from a specific member. Furthermore, complex cryptography is not practical computationally since the size of messages and the computational power of the nodes are extremely limited. As a result, minimalist cryptography should be adopted such that it requires low packet overhead and is simple to compute.

The challenges presented above in the context of peer communication in wireless ad hoc networks have been tackled thoroughly by the research community over the past years. However, the peculiarity of the networks we consider in this dissertation imposes the need for novel solutions. In particular, this work aims at providing a broad study of requirements and possible solutions for a decentralized large-scale network of resource-constraint devices, arranged in ad hoc (ever changing) topology and capable of dealing with large amounts of data.

1.3 Contributions and Outline

Independently of the application running in highly packed areas, the ultimate goal is to send new messages at high frequency, while providing good reachability to the intended destination, be it a collection of nodes or the entire network. To optimize such application-level communication, we take a layered approach by performing optimizations at the level of functional layers. Unlike wired networks, most of the wireless communications do not provide full specification of the OSI model. Such is the case for IEEE 802.11 [59] and IEEE 802.15.4 [3] where only the physical and MAC layer are specified. However, we consider the functional layers that have greater impact on the performance of peer communication applications. More specifically, we focus on the *medium access control* (MAC) layer, the *network* layer responsible for routing messages, and the *application* layer, which has the application logic.

We start off by studying each of these layers individually and propose parameters and solutions that better fit each of them taking into consideration the overall system configuration. At last we conduct experiments that aim at cross-layer evaluation of the system. In particular, we explore the interdependency between the MAC and network layers and look at how this affects the overall performance of the application-level communication.

The rest of the dissertation is organized as follows:

Chapter 2 lays the foundations of this dissertation by introducing a group-monitoring application in a mobile ad hoc network. We study the requirements from the technical perspective and, most importantly, we point out constraints and challenges that such a system might face. Particular focus is placed on the privacy issue of data sharing across ad hoc networks. Overall, this study provides us with the necessary understanding of the system we want to build and gives a good picture of the requirements for the underlying layers (i.e., MAC and network).

Chapter 3 focuses on the network layer, more specifically on message dissemination in ad hoc networks. Based on the system requirements we select a well-known message dissemination protocol, namely, Gossip3. The contributions of this chapter are the following. First, we perform an extensive study of the parameter set of Gossip3 and investigate how different parameters affect the protocol's performance in terms of (i) dissemination coverage, (ii) latency and (iii) the number of forwarding nodes. We also see how different parameters interact with each other. Second, based on the study, we propose a novel self-configuration algorithm for Gossip3. In the algorithm, each node dynamically adjusts its local parameter settings using only local knowledge. We show through simulations that as a result of these autonomic local adjustments, the network as a whole achieves a nearly perfect dissemination coverage with a minimal number of forwarding nodes. In other words, our self-configuration algorithm removes the burden of manually configuring each node, thereby making Gossip3 an attractive all-to-all dissemination protocol for the kind of real-world ad hoc wireless networks we target at.

Chapter 4 presents a study of medium access in ad hoc networks that run data-intensive applications. A thorough study of the existing MAC protocols leads to the choice of a carrier sense medium access protocol (CSMA) for sharing the medium among nodes. The contributions of this chapter include the evaluation of the impact of congestion at the MAC layer on message dissemination. Furthermore, for a number of network densities we evaluate the parameters of a CSMA protocol in terms of number of messages delivered per time unit and fairness in message delivery. We finally conclude, based on a study of the literature, that performing live adaptations to the MAC protocol is very challenging and may require thorough dedicated studies.

Chapter 5 puts together the findings of the previous two chapters and provides an exhaustive study of the interplay between the MAC and the network

layer in the context of a flooding-like message dissemination scenario. In particular, a number of parameters of the CSMA protocol are tested against a number of parameters of Gossip3. Here we observe that both the MAC and the network layer have to be properly set in order to maximally benefit from the limited resources of an ad hoc network.

Finally, **Chapter 6** presents conclusions, lessons learned and future work.

CHAPTER 2

CASE STUDY: GROUP MONITORING IN THE CROWD

The recent developments in sensing, computing and communication have led to a paradigm shift in wireless networking. Started first as small-scale monitoring of scientific and industrial domains, the introduction of the unobtrusive networked devices into popular consumer electronics has opened the door to a whole new range of application possibilities. In particular, a massive adoption of such unobtrusive networked devices by people can be leveraged to make real-time information about events in urban-scale area available to the general public. For instance, information retrieved by a single, or a set of, devices can be propagated through other devices in multi-hop fashion reaching out to cover a far broader area. Large participation of people is a key feature for ensuring connectivity and hence improving network communication, but at the same time it introduces a number of novel issues.

In this chapter we introduce a case study concerning group communication in the crowd using wireless networks. Through the study of system requirements, we aim at fully understanding the challenges and limitations for actually building group communication applications in large ad hoc networks. In particular, to deal with potentially dense and large-scale networks

where people are involved we identify two main problems: efficient group communication and user's privacy. Indeed, big crowds can be challenging for enabling communication, traditional centralized techniques may not be suitable. Moreover, a substantial part of this chapter is dedicated to the study of privacy and security aspects for group communication. Indeed, transferring personal information across an hoc network exposes the users to a number of threats concerning privacy. If not properly managed, the system may be used by adversaries to track down users or even use the system itself to insert misleading information on behalf of target nodes.

By understanding the system requirements, we can determine a number of design principles and propose a protocol for safe group communication in the crowd. More specifically, the contributions of this chapter include the study of the general requirements for our system (Section 2.2), analysis of privacy and security threats (Section 2.3), and a protocol that ensures privacy of the users in wireless group communications (Section 2.4).

2.1 Application Overview

The aim of the application is to enable communication between groups of people in the crowd by means of unobtrusive wireless devices. To illustrate it with a simple example, consider groups of friends, families or school trips that attend a massive public event. Because of people's free mobility, a group may likely fall apart or some may simply take a wrong direction. Ideally, such changes should be mirrored by the group monitoring system as soon as possible and, when necessary, an alarm should be triggered within the group.

To this end, each participant is equipped with a wireless device that periodically sends to his or her group members a new message containing information concerning their current situation. Such information may include location information, estimation of people concentration in the area, a short text message, and so on.

Once an information sharing facility is established, it is easy to foresee a number of additional functionalities. For instance, the users of the system may receive information by the event organizers about services available in the neighborhood or warnings of overcrowding in certain areas. Moreover, sniffers or receptor stations placed in different places can be used by the organizers in order to monitor the mobility of the people and predict hazardous circumstances.

However, this case study leaves out the group monitoring application as such. The problem of monitoring groups of people in an ad hoc network has been addressed by Cattani et al. [17]. The main focus of this study is to analyze the requirements for building a robust communication system for group monitoring applications. The data content of the packets exchanged between users and the way such data is processed to perform group monitoring is outside the scope of our study.

2.2 General Requirements

2.2.1 Independence from any Infrastructure

One of the main issues that our system faces is the need to handle hundreds of thousands of users at the same time and in a limited area. Nowadays many big software companies such as Google, Apple, and Amazon provide a number of applications, which run on mobile devices for a multitude of purposes and scenarios. Through a proprietary cluster of servers, running in what is best known as *the cloud*, such applications are built to scale to million of users. Moreover, a number of phone tracking applications are already available for smartphones based on GPS data.

The problem lies in the fact that in order to access and share location data with a group of people, it is required to have Internet access. This can be very limiting, since Internet connectivity is not always a given in mobile networks. First and foremost, cellular networks have difficulty scaling to large crowds. In dense areas, a huge number of connections to the cellular network may be served by just a few cellular antennas. As a result, connectivity is likely to fail. Moreover, with the increase in popularity of smartphones the problem of congestion has become even more pronounced—not just for the bandwidth they consume, but more for the signaling traffic they generate. In particular, signaling traffic is primarily caused by data-centric, always-on devices like smartphones, which keep connecting and disconnecting from the cellular network to prolong their battery life. Reports specialized in the field of cellular communications [57] show that the signaling traffic can be twice as much as data traffic.

An alternative to the cellular network could be the mesh networks. Basically mesh networks consists of a number of antennas connected via wireless interfaces forming a backbone for routing messages received by mobile devices. Such a system would require the installation of a significant number

of antennas, not only for providing coverage of the interested area, but also handling very large number of users simultaneously. Moreover, financing the installation of infrastructure would be a serious obstacle for the deployment of our system, especially considering ad hoc events such as festivals, crowded beaches or other outdoor activities.

Our system is required to take advantage of the spontaneous distribution of the nodes in powerful ways. In order to enable effective communication between nodes, they should rely solely on the network formed by other nodes present at such events. However, this does not exclude the possibility of having wireless infrastructures for other purposes, like collecting information from the network of nodes, or sending out information from event managers.

A purely ad hoc mobile network has the disadvantage of becoming poorly connected or even disconnected in sparse areas. Although our target scenario concern highly populated areas, we assume that the concentration of people may vary over time. As dense and sparse networks expose quite distinct properties, an important challenge for our group monitoring system is to work efficiently under different densities.

2.2.2 Hardware

Hardware plays a fundamental role in our system. While most networked devices equipped with a radio can provide ad hoc communication, there is a number of requirements that have to be met. First and foremost, the devices should be easily carried by anyone, including children or even pets. Hence, they are required to be unobtrusive and possibly inexpensive, such as for example, in the form of an arm band, wrist watch, badge, or collar. This seems to exclude power-hungry devices that rely on the IEEE 802.11 standard, e.g., smartphones, as they need large batteries to operate.

A valid candidate platform for our system are wireless sensor networks, a relatively new class of computing, which consists of collections of tiny networked devices. Provided with an ultra-low-power microcontroller, a low-power radio and interfacing the surrounding environment with a number of sensors, such devices can be used to monitor vast areas for months. For example, the IRIS motes introduced by Crossbow in 2007 are equipped with an ATMegal1281V chipset that can operate at 7.37 MHz, 8 KB of SRAM and 128 KB of program memory. For networking, it employs a low-power radio with maximal bandwidth of 256 Kbits/s.

Second, the small size of sensor nodes imposes a serious limitation on their battery capacity. Consequently, we require minimal energy consumption for their sustainability. Despite the significant progress of research on WSNs, energy consumption remains a problem. In particular, the radio is one of the most energy-demanding components. The amount of energy consumed by the various components depends heavily on the specific device. However, the battery is typically drained by the radio when it is on, in listening mode, and, upon packet reception or transmission energy consumption may increase up to 50%, as reported by Ye et al. [102] and Stemm and Katz [91]. In static sensor networks, this problem is addressed by introducing duty-cycling, where nodes turn off their radio for a considerable fraction of time, and communicate only when their radio is active. A good synchronization between neighboring nodes is fundamental in order to make sure that their radios are turned on at the same time in order to transmit and receive packets. In general, duty-cycling is broadly used in sensor networks that aim at monitoring events of static nodes and can significantly prolong the lifetime of the network.

Although we target similar low-power devices, our network differs a lot from traditional sensor networks. In particular, due to data intensive scenarios imposed by our group communication application, adopting duty-cycling would not be practical for our system, as that would tremendously reduce the bandwidth. Moreover, since we consider mobile nodes, synchronizing their active/sleep times would add extra complexity, in addition to reducing the bandwidth [22].

Unlike traditional sensor networks, our system should work for a limited amount of time, namely, the time span of a single-day event. Table 2.1 shows the power consumption of the various components of an IRIS Crossbow mote. If we assume a node to work in full operation mode, by maximally using its CPU, continuously receiving/transmitting packets, and performing reads/writes to the flash memory for 50% of the time, the total energy consumed per hour would be approximately 30 mAh. So, running a sensor node at full load will drain an AAA battery (800 mAh) in about 27 hours. This implies that we can safely assume rechargeable or even disposable batteries, which utilize at maximum the available resources in order to accomplish their tasks.

A third requirement concerns the integration of sensing capabilities. As in traditional WSNs, embedded sensors could be used to gain and share more insight about the users. For instance, accelerometers could provide information about the activity of the users. Furthermore, location information could

Operating Mode	Current (mAh)
ATMega1281V, full operation	6 (7.37 MHz)
ATMega1281V, sleep	0.010
Radio, receive	16
Radio, transmit (1 mW power)	17
Radio, sleep	0.001
Serial flash memory, write	15
Serial flash memory, read	4
Serial flash memory, sleep	0.002

Table 2.1: Current consumption of various system components and operations for an IRIS OEM Crossbow node.

be retrieved by using GPS or through beacon-based techniques. While GPS provides a solid way of detecting location and does not require additional infrastructure, it can be relatively expensive and power-hungry. According to a study made in 2010 [16], the GPS installed on a smartphone (e.g., Google Nexus One) consumes 144-160 mW. It would require 53mA - 61mA if GPS were to be installed on a Crossbow node. So, installing a GPS on an IRIS Crossbow mote would consume double of the energy it takes the mote to operate at full load.

Alternatively, beacon-based localization techniques do not require the installation of additional hardware on the nodes. Instead, stationary long-range antennas periodically broadcast beacons containing their location. Receiving nodes can apply triangulation techniques on the location information received by different beacons (at least three) to estimate their relative location [47; 46]. The drawback of this localization technique is that a number of beacon antennas are required to be installed around the area of interest.

2.2.3 Effective Group Communication

The main goal of our system is to keep people informed about the whereabouts or the state of their group members over time in a crowded area. As people are on the move, messages sent by their devices have to reach the relative group members within a reasonable time in order not to lose their validity. This may prove to be challenging for our system. In particular, because

all nodes insert new messages on a regular basis, the network is expected to deal with a high load of messages and to work under utterly congested conditions. As a result, packets may be delayed or even get lost in the case of channel quality deterioration.

To cope with such conditions we identify a number of prerequisites:

- ***Minimize the packet size*** A common way to increase the traffic that a network can handle is adopting small data packets. The size of data packets impacts directly the time that each transmission occupies the channel. Thus, the smaller the data packets the higher the number of transmissions and also the lower the chances of packet collisions. Typically, sensor networks use packet sizes between 32 and 128 bytes.

- ***Reduce overhead of channel access.*** Some low-power Medium Access Control (MAC) protocols stretch the length of the standard preamble (part of the packet header) up to 10 times the size of the data packet [70] in order to alert potential receivers about an upcoming message transfer. This way nodes that sense such a signal keep their radios on until the end of the message. While this technique is very popular among sensor networks where nodes sleep most of the time and the data traffic is relatively low [24; 70], the throughput reduction would be prohibitive for the kind of data-intensive application we are targeting.

- ***Prevent packet collisions*** As our system should deal with a data intensive application, communication between nodes can lead to a great deal of packet collisions. This is particularly the case when nodes access the medium unsynchronized and at a high rate. It is necessary to minimize collisions while allowing the channel to be maximally utilized. Chapter 4 discusses this issue in detail.

- ***Control message insertion rate*** Even if the underlying network layers prevent packet collisions, when the number of messages injected in the network by the application exceeds the delivery capacity of the network, it can lead to buffer overflow at the MAC layer. Packets may be dropped out of the outgoing buffers of nodes. Setting an appropriate pace of inserting new messages is crucial. This is discussed in Chapter 5.

An important problem in the kind of ad hoc networks we are considering is end-to-end latency of messages. Generally speaking, it is difficult to guarantee timely delivery because it depends on a number of factors, most of

which are not possible to control from a system's perspective. Such factors include network density, distance between communicating nodes, traffic load and mobility of the nodes. However, by efficiently utilizing the network resources, we can reduce the traffic generated by the nodes. A lower load at the relay nodes can speed up message delivery. Latency of message propagation is a crucial issue that we discuss thoroughly throughout the next chapters.

2.3 Privacy and Security

Communications over a wireless channel are vulnerable to a number of attacks. Unlike wired networks, data transmitted over a wireless channel can be overheard by anyone in the physical neighborhood. For instance, a *passive attacker* may simply overhear communications in order to gain insight about who is talking to whom or even inspect the data contained in the aired packets. Whereas, so-called *active attackers* may alter the contents of the packets, retransmit them at a different moment possibly at different places, transmit their own bogus packets, or even prevent communication in the neighborhood by jamming the channel.

Mobile ad hoc networks, besides being vulnerable to possible threats related to their wireless nature of communications, are also exposed to a number of perils related to the way nodes interact with each other. More specifically, due to multi-hop routing, messages aired by a source node, have to cross several nodes before reaching the destination. As a result, messages are exposed to a number of attacks to routing, as we shall see in Section 2.3.2. Moreover, the fact that users are mobile has several implications to their privacy. For example, as a user roams with her mobile device, the device itself may become a way to continuously trace her whereabouts, hence jeopardizing her privacy. This is further aggravated by the fact that in our application people are arranged in groups, thus implying that if members of a group can be singularly traced over time, then the safety of the entire group may be compromised.

This main objective of this section is to discuss possible threats to privacy and safety of group communications. The result of this study will then be used further on to propose the design a protocol that ensures privacy in group communications.

2.3.1 Adversary Model

We consider passive and active eavesdroppers equipped with powerful devices that can be ubiquitous in our system. The passive ones may consist of small low-cost sensor nodes that can be planted anywhere in the playground to monitor ongoing activities. Overheard information may be shared among adversaries using long-range wireless communications in order to correlate the data.

On the other hand, active adversaries may send new messages, forward received messages or simply replay them afterwards. These adversaries are capable of computing with high probability the direct (one-hop) sender of a message. To this end, special electronics, such as directional antennas and spectrum analyzers, can be used to compute the angle of arrival and the received signal strength of a message and infer its direct sender. To trace people physically, adversaries may move from one place to another. Finally, we assume that the protocols run by the nodes are known to the adversaries.

2.3.2 Threat Models

Disruptions to Routing

Many routing protocols in ad hoc networks are quite simple, and for this reason susceptible to several forms of attack. A comprehensive study of these attacks is provided by Karlof and Wagner [39]. On-demand routing is among the most common routing techniques in mobile ad hoc networks [63; 38; 68; 67]. Basically, a source node broadcasts at first a route request (RREQ) to the entire network and, once the RREQ reaches the destinations, the latter replies back with a route reply message (RREP). Intermediate nodes in the route between source and destination maintain routing tables for future communications between the two. Communication to multiple nodes results in a rooted tree [81; 99; 60; 34; 35], or even a mesh overlay between a source and its destinations [90; 89; 25; 48; 21; 62; 87].

On-demand routing relies on the collaborative behavior of the relaying nodes, which makes it particularly vulnerable to active attacks. For example, consider an adversary that behaves like normal relay nodes by participating in the routing. It can intervene in the function of forwarding by selectively attracting the traffic from victim nodes and then dropping their data packets. To do so, the attacker can claim to have a particularly good route to the

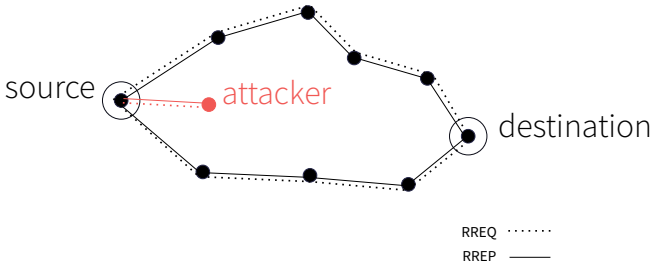


Figure 2.1: *Illustration of a black-hole attack.* The malicious node, in red, claims to have a shorter route to the destination compared to the ones offered by its neighbors. So, the malicious node attracts the traffic from the source to the destination.

destination. This way, he will attract the messages destined to a target node and then drop them. This attack, also known as *black hole* attack, causes the packet delivery ratio between targeted nodes to decrease considerably.

Moreover, an adversary may prevent a route from being discovered between two nodes that are otherwise connected. By answering the route requests with false indications, an adversary may create wrong routes (e.g., loops) that prevent linking two victim nodes with each other, although there exists a route between them. Such attacks may have a major impact on the communication between group members where certain nodes may be isolated from the rest of the group. Applying this scheme to a larger scale would bring the communication between all members of a targeted group to be significantly obstructed. In general, all routing protocols that use a single path to route a packet to its destination are vulnerable to selective forwarding attacks.

Routing through multiple disjoint paths would make black-hole attacks more difficult to succeed, in addition to providing fault-tolerance to broken routes. However, routes believed to be disjoint could be in fact an illusion caused by a single adversary presenting multiple identities. This is the case of the Sybil attack [23], where single nodes present multiple identities to other nodes in the network. An example of a Sybil attack is depicted in Figure 2.2.

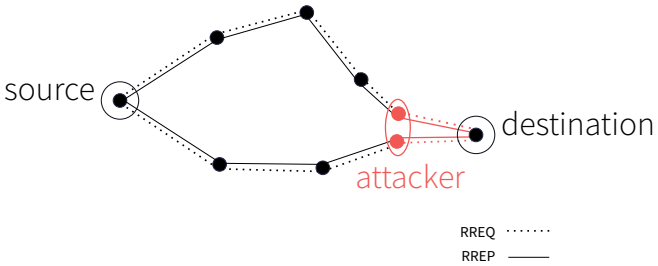


Figure 2.2: *Illustration of a Sybil attack in multi-path routing.* The malicious node, in red, presents two different identities, giving the impression to the source node that there exist two different paths to the destination. So, the malicious node will attract all the traffic from source to destination.

Position-based routing is another form of routing that leverages nodes' location information to efficiently direct their traffic towards the destination [8; 42; 104; 43]. For instance, in GPSR [42] at each hop nodes greedily forward incoming packets to the neighbor closest to their relative destination. Regardless of their position, adversaries may indicate their location to be closest to the destination and subsequently attract the traffic between target nodes and set up a selective forwarding attack as seen above. Furthermore, the adversary may dramatically increase the chances of success by setting up a Sybil attack, where a malicious node advertises multiple nodes claiming to have closest location to the destination.

Injection of Bogus Data

The most common form of *active attacks* includes sending out messages containing misleading information, such as for instance, messages that impersonate targeted nodes. To this end, a victim node may be impersonated by sending a message containing bogus information about its state (e.g., location) or by replaying old messages from that node. When this sort of attack is set up in conjunction with a black-hole attack, then the results could be devastating. This way, legitimate messages sent by a victim node could be dropped deliberately and group members would consistently receive only bogus messages concerning that node. Consider a scenario where one mem-

ber of the group is drifting away, while an adversary's instrumented device (falsely) assures the other members of the group that their friend is nearby. Such form of attacks would turn our application into an instrument to the hands of malevolent people.

Channel jamming is another form of active low-effort attack where the adversary simply transmits in the same channel. This kind of Denial of Service (DoS) attack affects the nodes within the reception range of the "jammer", which is determined by its transmitting power. During these attacks nodes in the interference range of the attacker are prevented from transmitting packets while the channel is jammed.

Another form of DoS attack sees attackers inject at high rate meaningless messages to be disseminated. Nodes that receive such messages, unaware of their content, will propagate them further. As bogus messages are injected at high rate, nodes will be mostly busy disseminating those messages. DoS attacks do not aim at particular nodes, but can have devastating effects on the entire network or just parts of it. In general, they are quite hard to address.

Traffic Analysis

By simply eavesdropping the wireless channel, passive adversaries could have access to a great deal of information about the users such as, for example, the contents of the data packets. Nonetheless, even if the adversaries cannot access the content of a packet due to encryption, they may still be able to observe who communicates with whom, by inspecting the headers of overheard messages. In particular, routing on-demand requires the source node to broadcast route requests when it needs to discover new routes to its destination. Typically a route request contains the identifiers of the source and the destination of the intended communication. Thus, an eavesdropping adversary can easily observe, independently of where she resides in the network, who wants to communicate with whom, and hence observe the group composition.

Another issue is related to the location privacy of the users. The regular transmission of messages from a node, can be exploited by adversaries to trace down a user by moving hop-by-hop backwards up to the source of the messages [61]. More specifically, if an adversary is on the route between a source and a destination, by following backwards the forwarders of the messages, the adversary can reach the source and find the location of a person any time. Such passive attacks can be facilitated when there are multiple adversaries located at various locations of the network.

2.4 Making Group Communication Private

To enable group monitoring in a mobile ad hoc network we adopt a proactive approach. At regular time intervals each node sends a new message containing the current state to its corresponding group members. As the intended recipients receive such periodic messages from their group member, they update the recorded state of the sender accordingly. When messages fail to be reached from a particular node for a time interval higher than a threshold, the application triggers an alarm.

Behind this simple application lie several issues mainly due to the properties of the network we consider. In this section we aim at a communication protocol between group members that prevents non-authorized parties to infer information about the whereabouts of users or group membership.

2.4.1 Design Guidelines

Before delving into the details of private communication techniques, we first examine the main design principles that shape our protocol. The design principles derive from the study of the requirements and security threats as described respectively in Sections 2.2 and 2.3.

Indistinguishable Messages

An effective way to retain control over the access of personal information contained in packets is to hide as much as possible from unauthorized parties. To this end, most wireless communications adopt encryption on the data packets. However, as we have seen earlier in Section 2.3.2 protecting the data is not enough. As it turns out, it is fundamental to prevent unauthorized parties from matching nodes with their messages or even simply identifying the nodes involved in a communication. This privacy principle is referred to as *unlinkability* [14].

A common approach to unlinkability is based on anonymity. Pfitzmann et al. [69] have defined anonymity as "[...] the state of being not identifiable within a set of subjects, the anonymity set". It is clear that a main restriction for adopting anonymity is that the *anonymity set* should be large enough such that individuals are not trivially identified. In our case we apply anonymity on two levels. First, to make a node unlinkable to its actions, the messages the node sends should be made anonymous. It implies that it should not be

possible to relate a node to its messages. As we consider large networks, the requirement on the anonymity set is fully met. Second, to make the communication parties (i.e., group members) unlinkable, information about the destination should be concealed. By doing so, the anonymity set, i.e., the possible destinations, would again include the entire network. It follows that by applying anonymity on the sender and on the destination, messages intercepted by adversaries cannot be bound to any node and, thus, cannot be used for traffic analysis purposes.

There is, however, a special case when the anonymity of a sender can be violated. If an adversary is located within the communication range of a target node, by adopting localization techniques and special equipment, he can potentially identify the location of the target node, and hence, the messages it sends. The trivial case consists of one target node being isolated from others: the messages it sends would be unambiguously identified by any receiving node in the surrounding area. On the other hand, when several transmitters are co-located, techniques such as, Angle-of-Arrival, Time-of-Arrival and RSSI used alone or in combination [53] can potentially localize wireless transmitters. It is important to note, though, that the accuracy of these techniques can be undermined by a number of factors. For instance, reflection and diffraction of the signal, due to obstacles, including human bodies, may seriously impact the effectiveness of estimating the actual distance from a transmitter (RSSI) or the angle of arrival of a signal. In crowded areas, as people are typically densely packed, identifying the sender of a message can be even more challenging. Even if the adversary succeeds in identifying the sender of a message, the information gained by capturing such a message cannot be used for identifying future messages from the same node– subsequent messages are not related to each other (i.e., anonymous).

No Point-to-Point Routing

As we have seen earlier, traditional routing protocols in ad hoc networks do not meet the unlinkability property introduced in the previous section. Indeed, nodes are typically required to include information about the destination in the packets they send. It implies that the communication patterns are exposed to unauthorized parties. However, a number of techniques allow to route without exposing the source or the destination.

Anonymous on-demand routing The problem of maintaining untraceable communications between nodes has been addressed by many studies [44;

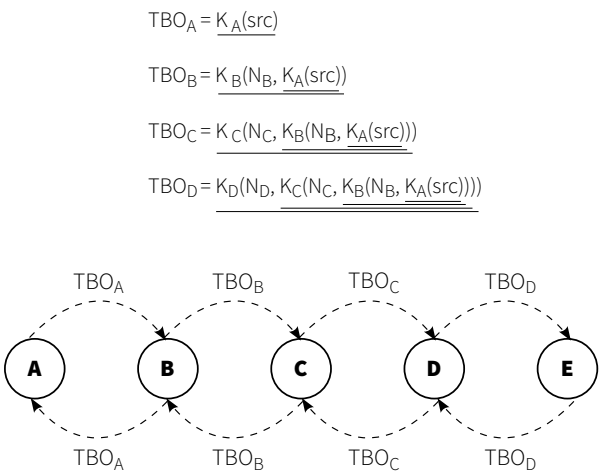


Figure 2.3: ANODR TBO: Anonymous route discovery from node A to node E.

106; 72; 86]. A common technique, also known as Onion Routing [75], allows forwarding nodes to route messages without knowing the source or the destination. More specifically, let us consider as an example the ANODR-TBO scheme [44]. A communication source A initiates the route discovery procedure by assembling an RREQ packet and locally broadcasting it.

$$< RREQ, tr_{dest}, onion > \tag{2.1}$$

The RREQ includes a cryptographic trapdoor tr_{dest} that can be opened only by the destination and a cryptographic $onion$, which is used for establishing an anonymous route from source to destination. As shown in Figure 2.3, each intermediate forwarder X embeds a random nonce N_X to the onion, encrypts the result with a symmetric key K_X, then broadcasts the RREQ locally. The trapdoor information consists of N_X and K_X, and is only known to X.

When the destination D receives an RREQ packet, the embedded onion is used to establish a *route pseudonym* towards the source. So, it broadcasts a RREP, which, in addition to the onion, includes also a locally unique random *route pseudonym* N_D.

$$< RREP, N_D, onion > \tag{2.2}$$

At each local RREP broadcast, only the next hop can correctly open the trapdoor it made in the RREQ phase, by means of the symmetric secret key

and the nonce. During the RREP phase, an intermediate node X in the path between source and destination selects a locally unique nonce N_X, stores the correspondence between the nonce $N_{PreviousHop}$ in the RREP and its N_X in the forwarding table, strips one layer of the onion, and replaces the $N_{PreviousHop}$ with its N_X. Then it locally broadcasts the modified RREP packet. Figure 2.4 illustrates the route pseudonym process.

At the end of the route discovery process a *route pseudonym* is created between the source and the destination. So, for each end-to-end connection, the source wraps its data packets using the outgoing route pseudonym (N) in its forwarding table. A data packet is then broadcast locally without identifying the sender and the local receiver, but only the corresponding outgoing N. All other receiving nodes must look up the route pseudonym in their forwarding tables in order to detect whether they can take part in routing.

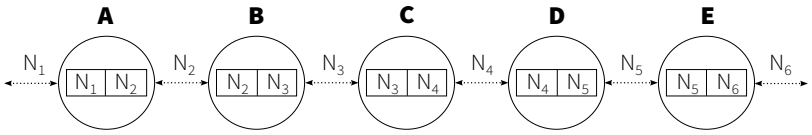


Figure 2.4: ANODR TBO: Using route pseudonyms $N_1..N_6$ to route messages from A to E.

Although this technique can well conceal the communication parties and prevent any attack from eavesdroppers or malicious nodes, it imposes high computational and communication overhead. First, a large amount of the traffic is dedicated to establishing routes between nodes. For each incoming RREQ nodes have to perform an encryption and store an entry in the routing table. In addition, nodes have to attempt to decrypt each RREP they receive. As the topology changes due to the mobility of the users, routes have to be re-established. This means that a significant effort, in terms of computation and bandwidth is devoted to route discovery and maintenance. As a result, routing is not a viable way for data intensive applications. Moreover, in large-scale networks a lot of memory would be required to store entries in the routing tables. That could be prohibitive for memory-constrained devices.

The Gossiping case. An alternative way to routing consists of flooding an anonymous message with trapdoor information for the recipient to the entire network, hence, eventually reaching the intended recipient of the message. Unlike routing, broadcast protocols are very simple and do not require information about the destination. A message is rebroadcast by several, or even

all, nodes in the network, which makes packets redundant and, therefore, resilient to single-route failures and topology changes. While flooding would benefit the property of unlinkability between a message and the communication parties, it generates a lot of traffic, as packets are rebroadcast often unnecessarily by many nodes.

Gossip-style dissemination aims at reducing the traffic of a plain flooding protocol while still providing full network coverage of the sent packets. To achieve this, gossip-style techniques require only a subset of nodes to rebroadcast a message. Several studies [96; 27; 103] suggest that this technique can dramatically reduce the number of rebroadcasts. Such techniques are a good trade-off as they keep the benefits of broadcast protocols while requiring less traffic. However, a careful selection of parameters is required in order to reach optimal performance, or they might either fail in reaching good coverage or generate high packet redundancy.

From the perspective of security, gossiping protocols are quite robust. Because the nodes in charge of rebroadcasting are selected probabilistically, it is hard for adversaries to predict the flow of messages between communicating nodes, and thus, affect the process of message dissemination. Moreover, adversaries capturing gossip messages can gain no information about the source and the destination. This way targeted attacks are intrinsically prevented from taking place. The main disadvantage of gossiping protocols is that, generally speaking, they generate more traffic than routing protocols. Instead of having single routes between source and destination nodes, messages are disseminated through several, unpredictable paths. While this mechanism may have a significant impact on the traffic, it provides a simple and powerful solution to issues such as, traceability of communications and topology changes. In the following chapters, we will discuss more in depth about gossiping protocols and their properties.

2.4.2 Group Communication Protocol

We propose a protocol for untraceable group communication. Based on the design principles discussed earlier, our protocol can be summarized as follows.

Group members communicate by transmitting at regular intervals a new message with their updated state. The communication occurs through a gossip-based protocol, which allows intermediate nodes to decide, in principle, probabilistically, whether to forward a received message. To make messages anonymous, no information about the source or the destination

node is contained in the message. Such a choice is sound, also considering that messages have multiple recipients (group members). The only information that distinguishes these messages is a random-looking number (one-time pseudonym) which can be used to identify the sender of the message. The one-time pseudonym can be accessed by any node that receives it, but it is recognizable only by the intended recipients. To protect the integrity and the confidentiality of the information, the data packet is encrypted with a symmetric secret key known only to the group members and a checksum is used for the data integrity. So, a message sent by node A at the i-th interval would look like this:

$$< K_A(A, data), ID_{A_i}, checksum > \tag{2.3}$$

where $K_A(A, data)$ is the encrypted data and real identifier A using the secret key of the sender K_A, ID_{A_i} is the pseudonym of node A at time i, and $checksum$ applies to the encrypted block and to the pseudonym.

Each node is provided at bootstrap with a secret sequence of one-time pseudonyms and a symmetric secret key, which are both known to the group members. At regular time intervals each node picks a new pseudonym from its own sequence and includes it in a new packet. Furthermore, it encrypts the data with its secret key and uses a gossiping protocol for flooding the packet.

Moreover, to be able to recognize anonymous messages sent by their group members, nodes maintain a hash table of expected pseudonyms from each of them. The hash table uses the pseudonym of a group member as key and its actual identifier as value. So, upon reception of an anonymous message, the receiver looks up–in constant time–the hash table for the pseudonym of the incoming packet. If the pseudonym is not found, then the receiver will know that it is not the intended recipient of the message. Otherwise, to make sure that the pseudonym is correctly identified, the receiver will try to decrypt the data packet with the secret key of the recognized group member. If the decryption succeeds, then the receiver confirms the identity of the group member that sent the message and reads the data contained in the packet. The activity of a node, both as a sender and a receiver is described in Algorithm 1.

Concretely, let us consider a group with only two members, A and B. At time interval t, node A wishes to send a new message m to node B. To this end, it adds to the message its successive pseudonym, id_{A_t}, from the preloaded sequence of pseudonyms. The data packet is then encrypted with the secret key of A, K_A, and broadcast. The intended recipient B, at the

same time t adds the same pseudonym of A, id_{A_t}, to the hash table of expected pseudonyms. This hash table has pseudonyms as key and the corresponding group member identifier as value. B keeps adding expected pseudonyms from A to the hash table at each time interval. Due to dissemination latency or packet collisions, the message m may reach B some time after or even get lost. As B receives new packets, it looks up the hash table: when the packet with identifier id_{A_t} reaches node B, the lookup will return the identity of the sender, A. If the decryption of m with the secret key of A succeeds, B confirms that the packet was sent by A. An important assumption in our approach to secure group communication is that devices are protected, preventing, thus, stealing of pseudonyms and secret keys.

An important prerequisite of this algorithm is clock synchronization between the communicating parties–pseudonyms should be generated at the same time both at the sender and at the intended recipient. If the clock of the sender drifts ahead, it may lead to false negatives at the intended recipients. Basically, an anonymous message may reach the intended recipient before its corresponding pseudonym is actually generated and added to the hash table of the recipient. As a result, the message is dropped immediately. To contrast this effect of nodes getting out of sync, we estimate the maximum clock drift of the nodes (δ) over the duration of the running time[1]. Thus, intended recipients add new items to the hash table of expected pseudonyms at time $t - \delta$, where t is their local time.

Properties of Pseudonyms

Each sequence of one-time pseudonyms is long enough to allow for a new pseudonym at every periodic message sent out to the group members. If a new message is inserted every minute, then, for running over 12 hours, 720 pseudonyms would be required for each node.

Earlier we mentioned the importance of minimizing the packet size and its impact on network throughput. In this context, the length of the pseudonyms directly influences the packet size, so has to be minimized as well. But if the pseudonyms' space is too small, the risk is that there may be many duplicates in the pseudonym sequences of the nodes. So, nodes will incur into many false positives and, hence, they will try to decrypt packets that they falsely identify as belonging to a group member. In contrast, if pseudonyms are unnecessarily long, then space will be wasted in the packets, negatively impact-

[1] We assume that nodes operate for a limited time, that is, for instance, the duration of a single-day event.

Algorithm 1: Activity of nodes involved in group communication

```
1  Variables
2      HashTable <nodeID,Pseudonym []> pseudonyms
3      HashTable <nodeID,Long> symKey
4      HashTable <Pseudonym,nodeID> expectedPseudonyms
   // Called by the nodes at regular intervals t
5  function PeriodicThread()
       // Send new message
6      newPacket.ID ← pseudonyms [myID].Successive()
7      newPacket.DATA ← Encrypt( GetCurrentState(), symKey [myID])
8      Send(newPacket)
       // For each group member add a new expected pseudonym in hash table
9      for node ∈ groupMembers do
10         K ← pseudonyms [node].Successive()
11         V ← node
12         expectedPseudonyms [K] ← V
13     end
   // Called on new packet arrival
14 function Receive(rxPacket)
15     if rxPacket.ID ∈ expectedPseudonyms then
16         nodeID ← expectedPseudonyms [rxPacket.ID]
17         if Decrypt (rxPacket.DATA, symKey [nodeID]) then
18             UpdateState(nodeID, rxPacket.DATA)
19         end
20     end
```

ing the network performance (higher chances for packet collisions and lower throughput). In order to optimally choose the length of the pseudonyms, it is necessary to have a rough estimate of the network size.

Data Protection

To keep the information secret from unauthorized parties a plethora of encryption techniques have been devised and tested for resource-constrained devices. Symmetric key cryptography is the simplest technique that adds minimal overhead in terms of computation and packet size. Karlof et al. [40] propose TinySec, an algorithm that uses chain block cypher (CBC) [9] and Skipjack cipher for data encryption. This kind of encryption provides also *semantic security* which prevents adversaries from learning even partial infor-

mation from the past messages (for static information in the data packet). To this end an 8-bytes block of non-repeating string is necessary to be included in clear. The one-time pseudonym seems to be the perfect candidate for this.

Moreover, to ensure authenticity of the data packets, TinySec generates a 4 bytes Message Authentication Code (MAC), which is a by-product of the encryption process with the CBC technique. As a result, minimal overhead is required for authenticating encrypted packets. Results show that encryption of 30 bytes data packets with TinySec requires only 6% bandwidth overhead and 10% increase of energy cost for packet transmission.

2.5 Concluding Discussion

In this chapter we introduced a system for group monitoring in the crowd. We examined a number of challenges that are derived from the requirements of such a system. Moreover, we analyzed some aspects of privacy and security of the information exchanged by the users. We then proposed a protocol that allows users to share private data within their group in a safe way. This protocol is mainly based on anonymity of the sender and receiver. Keeping communication parties anonymous not only protects the privacy of the users (e.g., location, traceability), but also prevents adversaries from setting up selective attacks on the messages of the users. Indeed, since messages cannot be linked to the communication parties, it is hard for an adversary to target users by disrupting the routing of their messages. Similarly, impersonation of users is infeasible since users adopt a different pseudonym for every message sent out. Moreover, the integrity of the packets is protected by a MAC, which prevents an adversary from changing the contents of packet unnoticed. Finally, our pseudonyms protocol inhibits replay attacks by design: replaying an old message would not have any effect as no node would take a replayed message into account.

Our protocol does not address Denial of Service attacks. For instance, one example consists of malicious internal nodes that pollute the network by inserting meaningless, yet regular messages to be propagated. This form of DoS attack aims at preventing normal operation of the network. Moreover, channel jamming is another DoS attack that aims at disrupting the functionality of the network in certain areas. Similarly, a malicious node can continuously send messages to nodes in a certain area in order to prevent them from receiving other messages. DoS attacks are known to be hard problems in ad hoc networks.

Anonymity is a technique that is widely adopted for privacy purposes in many communication protocols ranging from Internet protocols to vehicular ad hoc networks and so on. However, it comes at the cost of significant communication overhead. Since point-to-point routing is not viable, flooding-based protocols are necessary to propagate messages between nodes in an unconcealed way. This way messages sent by a node have to cross multiple hops in order to reach its group members. With pure flooding, each message is replicated as many times as the number of nodes encountered. So, in a network composed of n nodes, if a node sends m messages per second, there will be $(n-1)*m$ new messages in the network every second. Since all nodes in our system act as sources, at every second there will be $n * (n-1) * m$ messages in the network.

To give an idea about the amount of messages that each node has to process, let us imagine that in a network of $1,000$ nodes, each node sends a new message every minute, hence, 0.016 messages per second per node. Assuming perfect communications, therefore, each message rebroadcast by all nodes, every second there would be $16,000$ messages in the network. Theoretically, by excluding the effect of congestion, each node would receive and transmit 16 messages every second. Under realistic conditions, the interference of simultaneous transmissions would generate many collisions, leading to poor dissemination.

To deal with large-scale networks, it is necessary to investigate rigorously message dissemination, aiming not only at providing reliability of communications but also, and especially, at minimizing the utilization of network resources. In Chapter 3 we address the problem of message dissemination in networks with properties similar to the ones we aim at.

CHAPTER 3

MESSAGE DISSEMINATION IN AD HOC NETWORKS

This chapter focuses on the network layer, which is concerned with getting packets from the source all the way to the destination. As the networks we consider consist of thousands of nodes, getting to the destination requires transferring messages through many hops.

A key observation of the previous chapter was that group communication in our system should occur through broadcasting rather than transferring information from one node to another through predefined routes. Indeed, broadcasting is a good match for our system not only as a technique for keeping the communication parties undisclosed. In large-scale mobile networks we aim at, flooding messages to the whole network constitutes a rather common and fundamental operation. Unlike point-to-point routing, flooding is resilient to topology changes and packet loss.

Since the advent of wireless networks, a lot of attention has been devoted to broadcast techniques as a building block primitive for communication between nodes. Whether for advertising route requests (e.g., on-demand routing) or for simply sharing information with the entire network [26], broadcasting is widely used in wireless multi-hop networks. In this chapter we

examine message dissemination from the perspective of our system scenario. The main question this chapter aims to address is under what conditions broadcast can be applied to our system. To this end, we conduct a broad survey of current broadcasting protocols (Section 3.2). Among them we select Gossip3, a representative broadcast protocol that fits well our requirements and that is highly parameterized. In Section 3.3 we conduct an extensive study of the parameter set of Gossip3 that aims at finding out how different parameters affect the message dissemination performance according to some metrics. We also investigate to what extent different parameters interplay with each other. Finally, based on this study, in Section 3.4 we propose a novel self-configuration algorithm for Gossip3 where each node dynamically adjusts its local parameter settings using only local knowledge.

3.1 Network Layer Requirements

Before starting with the survey of existing broadcast protocols, it is worth restating the context in which the network layer protocol has to operate. The application layer, responsible for sending new updates to the group members, hands out at regular time intervals a new message to the network layer to propagate. In addition, the network layer has to decide whether to propagate messages received from the neighboring nodes.

In this context, it is important to stress that we are looking into networks where nodes can be situated in quite diverse regions with respect to node density. In parts of the network where nodes have hundreds of neighbors, due to the high connectivity, one rebroadcast can cover many nodes. As a result, a low number of rebroadcasts is sufficient to ensure message propagation. In contrast, sparse networks have unreliable links among nodes. The unreliability of the internode connectivity has to be compensated for by the message dissemination protocol through message rebroadcasts.

Furthermore, as users move around the playground, the connectivity between the nodes changes. As a result, nodes will likely have a different view of the network at different moments in time, which implies that the density of the network may also vary quite dramatically. For this reason, the mobility of the nodes requires message dissemination to account for the various network conditions.

3.2 Broadcasting in Ad Hoc Networks

Broadcast protocols have been studied extensively due to their important role in many application scenarios. Plain flooding is the mechanism by which each node, receiving a flooded message for the first time, rebroadcasts it only once. This simple mechanism is extremely effective in covering all the nodes in the network with broadcast messages. But plain message flooding can incur a very high number of transmissions, a large part of them being redundant [58]. On the other hand, sophisticated protocols that reduce the number of transmissions dramatically by organizing nodes in a tree-like structure are impractical in dynamic scenarios. The key issue in broadcast protocols is the trade-off between the following contrasting goals: minimizing the message overhead and providing reliable message dissemination.

We group broadcast protocols for wireless ad hoc networks as follows.

3.2.1 Local-Knowledge Based Broadcasting

Local-knowledge based approaches generally decide on a per-node basis whether to rebroadcast an incoming message. One of the simplest ways to reduce message redundancy is by letting nodes decide probabilistically whether to forward a new incoming message. This technique, also known as *purely probabilistic*, is based on the fact that one transmission covers many nodes at once. Thus, randomly having some nodes *not* rebroadcast saves network resources without harming dissemination effectiveness. The properties of such a probabilistic scheme have been thoroughly studied in order to understand the impact of the rebroadcast probability p on message dissemination. Haas et al. [27] explore analytically and experimentally a vast range of probabilities in order to understand the core properties of probabilistic broadcast (also referred to as gossiping) in large wireless networks. They observe a bimodal behavior in gossiping, meaning that, in general, a broadcast message is either received by all the nodes or by none. Sasson et al. [83] have theoretically explored the same phenomenon based on percolation theory. They conclude that there exists a threshold probability $p_c < 1$ such that for any $p > p_c$ the coverage is close to 100% while for $p < p_c$ the coverage is very low. Moreover, they found that for any given uniform network configuration there exists an optimal probability p_c .

It is important to note that in the studies previously mentioned the optimal p applies on networks with uniform node distribution. In case of non-

uniform distribution of nodes, the optimal rebroadcast probability differs according to the local density. Several techniques have been proposed to approximate the optimal probability independently of the local density of the nodes. For example, Zhang et al. [105] have proposed a dynamic probability scheme where the rebroadcast probability of a node is adjusted depending on the number of times a message is received from the one-hop neighbors. While this approach makes the probability p adaptable, it introduces another static parameter (e.g., timeout) that is density-dependent. Furthermore, in Dynamic Gossip [84] the author introduces a so-called density-driven gossip where the rebroadcast probability p is adjusted based on local density awareness. Assuming a uniform distribution of nodes, dynamic gossiping uses a mapping between network density and optimal rebroadcast probability. Nodes estimate the density through relay pinging and apply the probability accordingly.

Another basic broadcast technique is based on the redundancy of a broadcast message for deciding on whether to rebroadcast. Ni et al. [58] observed the inverse relationship between the number of times a message is received by a node and the efficacy of a possible rebroadcast from that node. The efficacy here is intended as the number of additional nodes a rebroadcast can cover. Hence, they introduced a counter-based technique. Upon the reception of a new message, the node initializes a counter C to one and sets a timer chosen at random between 0 and T_{max} seconds. During the time interval the counter is incremented by one for each redundant packet received and if, at the end of the interval, the counter is less than a threshold the packet is rebroadcast. Otherwise the packet is simply dropped.

The main problem with the counter-based scheme is the fixed redundancy threshold, which does not account for various network densities. Originally, the value of the threshold counter was proposed to be six. But that would not be suitable for dense or sparse networks. To overcome this limitation, adaptive approaches of the original counter-based scheme have been proposed [92; 101; 56].

3.2.2 Area-Based Broadcasting

The idea behind the area-based approach is that the effectiveness of a rebroadcast depends on the additional area it covers. So, if a node receiving a rebroadcast message is only one meter away from the sender, then rebroadcasting would not be relevant since the additional covered area would be quite low. On the other extreme, if a node is located at the boundary of the

sender node's transmission range then a rebroadcast would cover a significant additional area.

Ni et al. [58] proposed a location-based scheme where each node must be able to determine its own location (e.g., GPS). When a node receives a rebroadcast message, it reads the location of the sender and computes the additional covered area were it to rebroadcast. If the additional area is less than a certain threshold, the node decides not to rebroadcast and the message is dropped. Otherwise, the receiver waits for a randomly picked time interval, commonly referred to as *random assessment delay* (RAD). After every redundant message received during this interval, the node will recompute the additional area covered by a possible rebroadcast. At the end of the RAD, the message will be rebroadcast only if that would cover an area larger than the threshold.

Quite similarly, a distance-based approach proposed by the same authors [58], takes into account the distance between a sender and a potential rebroadcaster to decide whether a rebroadcast is required. So, upon the reception of a broadcast message, a node cancels the rebroadcast if the distance d is smaller than a certain threshold. Otherwise, it initializes a variable $d_{min} = d$ and waits for a random interval. During this interval, the d_{min} is updated with the minimum distance. If such value gets smaller than a threshold, the rebroadcast is cancelled. The distance is assumed to be estimated by the received signal strength indicator (RSSI).

Miranda et al. [55] observe that such a distance-based scheme [58] is suboptimal as it does not take into account the message redundancy. For example, think of a node which acts as a bridge between two parts of the network. Upon the reception of a broadcast message such a node may decide not to rebroadcast it simply because it is too close to the sender. This would leave a part of the network uncovered. To address this, Pampa [55] takes a hybrid approach between a counter- and a distance-based scheme. Just like in the counter-based approach a counter is initialized upon the reception of an unseen message and incremented after every redundant message received during a certain time interval. But, unlike the counter-based approach, the RAD is replaced with a time value inversely proportional to the distance from the sender of the broadcast message. This means that a node farther from the sender sets a shorter waiting time, so it will rebroadcast earlier than those closer to the sender. This way, nodes close enough to the sender may not rebroadcast a message if it turns out to be redundant.

Using insight about the location of nodes can be effective for message dissemination. Area-based protocols rely on location information obtained by

GPS equipment. As such, they may be prohibitive to adopt in low-power networks given their high demand of energy, as discussed in the previous chapter. On the other hand, distance-based approaches that rely only on RSSI information do not require additional hardware. In general, retrieving RSSI information from the received packets can be an integrated feature of the radios. However, just like counter-based protocols, they rely on some constant thresholds which do not take into account the density of the area where nodes are located.

3.2.3 Overlay-Based Broadcasting

Overlay-based broadcasting techniques are known as the most efficient in minimizing the number of rebroadcasts [96]. As such, they have been extensively studied. A good part of research in this area is mainly inspired by a theoretical optimal case for choosing which nodes to rebroadcast, also known as the minimum connected dominating set (MCDS). An MCDS is the smallest set of rebroadcasting nodes such that the set of nodes is connected and all nodes outside the set are within one hop from at least one member of the MCDS.

While it has been proven that building an MCDS in a network is an NP-complete problem [49], a plethora of protocols have been proposed [37; 65; 66; 71; 51]. The main idea is that the rebroadcast should be performed by a set of nodes connected with each other which have high connectivity (degree). To this end, nodes share information about their neighbors such that each node has a view of the two-hop neighborhood. For example, in [66; 71; 51] a node receiving a new broadcast message makes use of the two-hops neighborhood in order to select appropriately the next forwarders among its one-hop neighbors. So, the rebroadcasting nodes are explicitly chosen by the upstream nodes. In particular, in multipoint relaying [71], the forwarders are chosen by first computing those two-hop neighbors reachable only by single one-hop neighbors. Those one-hop neighbors are then added to the list of forwarders (i.e., multipoint relays). After that, out of the remaining one-hop neighbors the one that covers the highest number of two-hop neighbors are added to the list of forwarders. The latter step is executed iteratively, until all the two-hop neighbors are covered. So, a node receiving a new message will forward it by specifying also the list of forwarders that are supposed to propagate it further.

The scalable broadcast algorithm (SBA) [65], similarly to the multipoint relaying technique assumes that nodes build and maintain a view of their

two-hop neighborhood. But it takes a different approach as nodes decide locally whether to rebroadcast a message. More specifically, upon reception of a new message the receiver waits for a time interval (RAD), which is randomly selected between 0 and T_{max}. T_{max} is inversely related to the degree of the node and is computed in such a way that nodes with higher degree result in having smaller delays. This ensures that nodes with higher degree rebroadcast before nodes with fewer neighbors. For each redundant message received during the RAD, the node will compute the one-hop neighbors that would be covered were it to rebroadcast. If by the end of the RAD all one-hop neighbors are covered by the redundant rebroadcast of other nodes, the rebroadcast of that message is cancelled.

The efficiency of this category of broadcast protocols depends on two-hop topology information. Typically nodes advertise their list of neighbors either implicitly by piggybacking it in the packets, or by specific *hello* messages transmitted at regular intervals. This has several implications to the applicability of such protocols to our system. First, as nodes move around, the network topology changes, and, as a result, the local view of the nodes concerning their two-hop neighborhood can be often outdated. This may affect significantly the performance of message dissemination both in terms of coverage and number of forwarders, as has been observed by Williams et al. [96]. Increasing the frequency of *hello* messages could mitigate this effect, but at the expense of increased traffic. Second, in dense networks, where nodes have hundreds of neighbors, sending *hello* messages, containing lists of neighbors, would simply be prohibitive. Consider, for example, a network where the degree of the nodes is 100 and nodes are identified by integer numbers 4 bytes long. This would require each node sending periodically *hello* messages 400 bytes long in addition to the regular data packets. If the rate of transmission of *hello* messages is high—due to node mobility—the overhead could easily outweigh the regular data traffic by which increasing congestion even further. Finally, overlay-based protocols aim at minimizing the communication overhead by selecting a minimal number of forwarders. While a reduction in the number of forwarders is very important in broadcast protocols, overlay-based protocols do not tolerate packet loss that derives from collisions or simply due to signal attenuation between communicating nodes. To cope with channel unreliability link layer signalling (e.g., RTS/CTS) could be used, but that would reduce the available bandwidth for data packets.

3.2.4 Discussion

We seek a simple, yet effective dissemination protocol that can provide good coverage in various network densities, able to cope with topology changes and that requires relatively low message redundancy. For the reasons mentioned earlier an overlay-based approach would not be a suitable solution. Moreover, an area-based technique [58] based on node location is prohibitive as it relies on energy-demanding equipments such as GPS. Local-knowledge protocols are by far the most suitable as they simply rely on local observations and can be robust to packet loss and topology changes. In particular, Gossip3 [27] is a well-known broadcast protocol that combines the benefits of probabilistic and counter-based approaches. It is designed as a result of an extensive study on the gossiping properties in ad hoc networks. A number of parameters allow Gossip3 to be easily fine-tuned for performing optimally on a variety of network configurations. However, the original work on Gossip3 lacks a broad evaluation of the parameters in various densities and the effect of radio communication (packet collisions and signal attenuation).

3.3 Exploring the Parameter Space in Gossip3

In this section we present the original Gossip3 protocol, and we explore the relation between its configuration parameters and network density. We start by describing in detail Gossip3.

3.3.1 Overview of Gossip3

Gossip3 constitutes an extension of probabilistic forwarding with a counter-based feature. The probabilistic part contributes to a quick propagation, whereas the counter-based part compensates for those packets that are not enough redundant during the probabilistic phase and that may die out prematurely. This way Gossip3 ensures a robust and simple message propagation protocol that can achieve high coverage with reduced traffic volume. More specifically, the operation of Gossip3 is determined by the following three configuration parameters:

Parameter p Upon reception of a packet, a node decides to forward it with a pre-configured probability, p. Consequently, with a small value of p, only a small fraction of nodes forward the packet.

Parameter m If a node has decided not to forward a packet in the first place, it waits for a short time interval snooping for the traffic on air. If within that interval the node does not hear at least m neighbors forwarding the packet, at the end of the interval, it will forward the packet despite its initial decision. This parameter is to compensate for packets that do not reach enough nodes due to probabilistic forwarding.

Parameter k Finally, to further minimize the chance of a packet dying out early, nodes within the first k hops from the source node that generated the original packet always forward the packet (i.e., for them $p = 1$).

In the original paper, $m = 1$ and $k = 1$ were claimed to be sufficient for most scenarios, while $p = 0.65$ was argued to provide the best performance. In the following section we revisit these claims by studying the impact of m, p, and k on coverage, traffic, and dissemination latency, for diverse network densities.

3.3.2 Simulation Settings

Our work is focused on a broad evaluation of the parameters of Gossip3 under realistic scenarios. In particular, we aim to see how the various parameters of Gossip3 impact dissemination performance when nodes experience traffic conditions. For this reason the choice of the simulation environment is crucial for this study.

The MiXiM Framework

The MiXiM simulation framework [45; 95] consists of a number of models and protocols for wireless simulations in OMNeT++ [1]. MiXiM provides detailed physical and channel modelling including multidimensional signals (time, frequency and space). This allows to obtain quasi-realistic simulations of state-of-the-art radio technologies, as in low-power radios. In particular, the attenuation as well as the actual receiving power of a signal are represented as functions of time, frequency and space. The interference between packets sent simultaneously is computed by considering the interference range which outbounds the communication range. So, to decide whether a packet is received correctly at a given node, a module named Decider calculates bit errors of a received signal based on information about its attenuation, interference, and actual receiving power. In our experiments we use the physical and MAC implementation of the standard IEEE 802.15.4.

We set the transmission power of the nodes to 1 mW, which is rather representative of low-power nodes. This results in a maximal reception range of 50m from the sender. Due to signal attenuation the chances of receiving a packet drop progressively as nodes approach the border of the reception range. The interference range is about twice as much as the reception range. The interference range refers to the distance a transmission may disturb other simultaneous communications.

MiXiM has separate modules for each layer of the protocol stack. This facilitates the implementation of our protocols in terms of functional layers. In particular, in this chapter our focus falls mainly on the network layer.

Experimental Setup

We consider networks of 529 nodes randomly distributed. The average internode distance Δ is: $5m, 10m, 15m, 20m$. The latter determines also the playground area where nodes are placed. A typical scenario would be having all nodes broadcasting new messages at a certain rate (we denote it *message generation rate*). Setting the message generation rate for a given network is not easy: a low rate is neither realistic for the scenario we aim at, nor a challenging problem. On the other hand, a high rate would give rise to a number of issues at the MAC layer (e.g., packet collisions and buffer overflow), which would not allow a proper evaluation of Gossip3.

For this reason, we evaluate Gossip3 in isolation from the other layers. In particular, for each network density, we configure the MAC to transmit at a rate that has been prior selected to generate optimal goodput[1]. Next, we emulate traffic at the MAC layer by imposing continuous transmissions of *dummy* packets that have the same size as a regular packet and whose role is only to reach nodes within the reception range. So, nodes generate a new dummy packet as soon as their MAC buffer is empty. As a result, nodes have at any time at least one packet to transmit, which ensures congested traffic, yet controlled by the optimal configuration of the MAC.

To evaluate Gossip3, three nodes, placed in various parts of the playground (center and borders), in addition to the artificial traffic at the MAC layer, inject 100 broadcast messages to be propagated with a frequency of one message per second. Whereas the rest of the nodes run Gossip3 for propagating broadcast messages they receive.

Table 3.1 shows the characteristics of each network configuration. The number of neighbors for a given node is computed as the sum of reception

[1] Goodput is explained in detail in Chapter 4.

ratios from all the nodes in the network (see Section 3.4.1). This estimation relies on the assumption that the network is static. The network diameter is the maximum, across all pairs of nodes in the network, of the length of the optimal route between that pair of nodes [29]. Finally, the MAC transmission success ratio for a given node represents the mean ratio of successfully received packets from all neighboring nodes.

Avg. Internode Distance	Avg. Neighbors	Network Diameter	MAC TX Success Ratio
$\Delta = 5m$	76	3	0.39
$\Delta = 10m$	29	8	0.52
$\Delta = 15m$	14	15	0.53
$\Delta = 20m$	7	23	0.53

Table 3.1: Characteristics of the considered network configurations.

3.3.3 Impact of Parameters m and p

We conducted an extensive series of experiments to explore the parameter space of Gossip3. More specifically, we tested all combinations for initial forwarding probability $p \in \{0\%, 10\%, 20\%, \ldots, 100\%\}$ and compensation parameter $m \in \{0, 1, 2, 3\}$. We tested each combination of p and m on four topologies of different densities, namely $\Delta \in \{5m, 10m, 15m, 20m\}$. In all these experiments, parameter k was fixed to the value 1, as suggested in the original Gossip3 paper.

The impact of p and m was tested with respect to three metrics: the coverage of the network, the induced traffic, and the dissemination latency.

Impact on Coverage

The top row of Figure 3.1 illustrates the effect of parameters p and m on network coverage. When $m = 0$, we see that the forwarding probability, p, has a clear impact on the dissemination coverage. It is not hard to see why: when a node decides ***not*** to forward a received packet, this decision is final, as the compensation mechanism is disabled. Consequently, for low values of p, packets die out prematurely, effectively reducing the dissemination coverage. In denser networks this effect is limited, as a single node deciding to

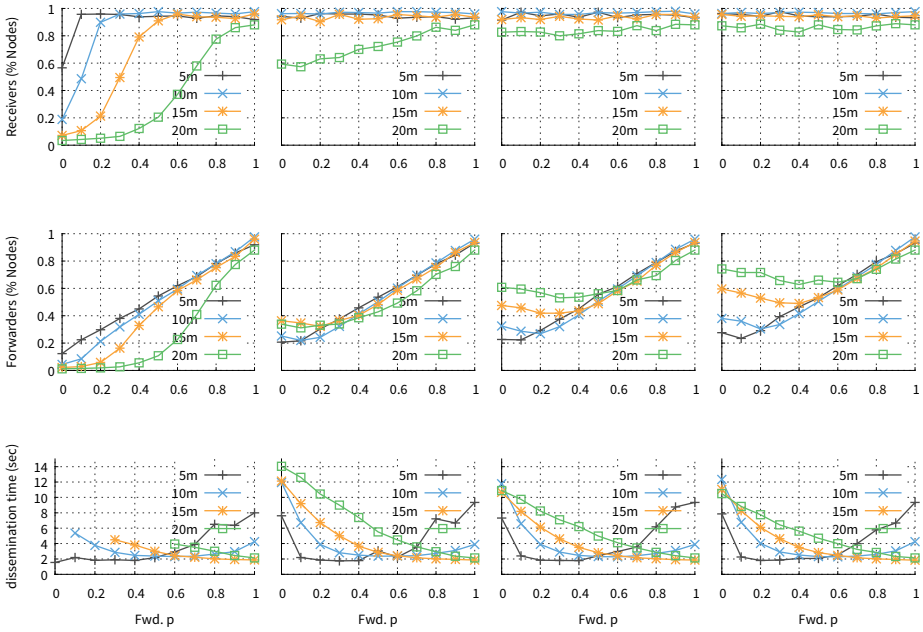


Figure 3.1: *Performance of Gossip3.* We vary the parameters p and m in networks with inter-node distance: $5m, 10m, 15m, 20m$. Latency represents the time, in seconds, to cover 90% of the nodes.

forward a packet is enough to cover a significant number of nodes around it. Topologies of lower density suffer significantly more.

In contrast, for $m \geq 1$ (still in the top row), the initial forwarding probability, p, has negligible effect on the dissemination coverage. This is a direct consequence of Gossip3's compensation mechanism, which compensates for insufficiently forwarded packets. The only exception is for the sparsest topology ($\Delta = 20m$) for $m = 1$. This is a special case, as the network is marginally connected, so a single node's decision *not* to forward a packet may result in a number of further nodes not receiving the packet at all.

In sparse networks we notice that in order to improve coverage, it is necessary to either start with a high probability, such as $p = 1$, which implies no compensation, or any probability (surprisingly even $p = 0$) in combination with compensation parameter $m = 3$.

Impact on Traffic

The middle row of Figure 3.1 shows the effect of parameters p and m on the amount of traffic induced by the dissemination of a single message.

As each node forwards a given message at most once, we measure traffic as the number of nodes that forwarded it, which essentially denotes the number of times a given message was transmitted. We express that number as a fraction over the whole number of nodes in the network.

Here we see that p strongly affects the number of forwarding nodes. Indeed, high values of p imply that nodes receiving a packet forward it with high probability, probably resulting in redundant transmissions, especially in dense topologies. On the other hand, for lower values of p, nodes receiving a packet are most likely ***not*** to forward it in the first place. As observed earlier, though, Gossip3's compensation mechanism forces certain nodes to deterministically forward insufficiently forwarded messages, resulting in no loss of coverage. Clearly, by means of the compensation mechanism, for low values of p nodes do not forward received messages unless "necessary".

Surprisingly, also setting the probability too low (e.g., as low as $p = 0$) may lead to higher traffic. When hardly any node decides to forward a packet, several nodes in the neighborhood jump in to compensate for it, leading to more transmissions than necessary.

The minimum number of forwarders is reached for $p = 0.1$, $p = 0.2$, or $p = 0.3$. Interestingly, due to the compensation phase, the same set of parameters $p = 0.1$ and $m = 1$ works best for most of the network densities, providing high coverage and a minimal number of forwarders. Although it appears as a win-win situation, the price to pay is higher latency, as we will see below.

Impact on Dissemination Latency

To represent the dissemination latency we average the time it takes packets to reach 90% coverage. The bottom row of Figure 3.1 depicts the effect of parameters p and m on the time needed for a message to reach 90% of the nodes.

Here we make two observations. First, in general low density topologies experience higher dissemination latencies. This is expected due to their larger diameter, which demands more propagation hops to reach out to the most distant nodes. Second, the forwarding probability p highly impacts the latency and its impact differs depending on the network density. For example, a general trend shows that the value of p has an inverse effect on the

dissemination latency. This is not surprising, as for large values of p nodes forward messages instantly upon reception and little is left to compensation. On the contrary, for low values of p, significant part of the dissemination is done through compensation. However, a different behavior can be observed for dense networks, i.e., $\Delta = 5m$ and $\Delta = 10m$. As p increases latency decreases, but for p greater than, respectively 0.5 and 0.7, latency increases again. This is an interesting observation as it shows the impact of high message redundancy. As nodes rebroadcast messages at high probability it is easy to see that messages will be buffered at the MAC layer waiting to be transmitted. Overall, the rate at which nodes forward the received broadcast messages is quite low compared to the incoming rate of new messages. Buffering time seems to increase significantly the dissemination latency, despite the fact that the network has small diameter.

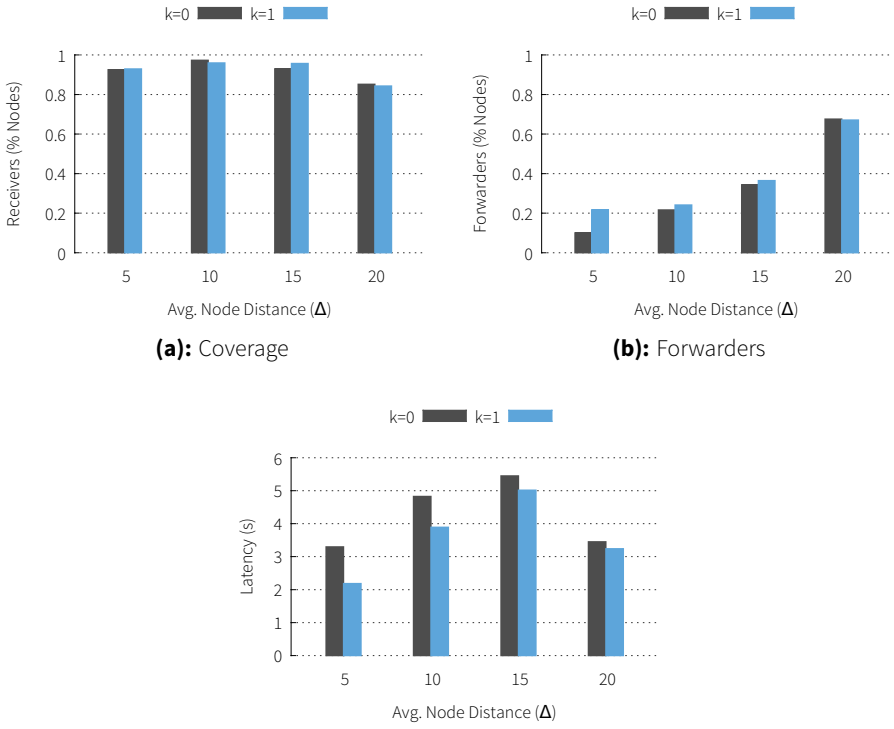
3.3.4 Impact of Parameter k

Parameter k dictates the number of (initial) hops for which nodes should forward a message deterministically, preventing a message from dying out before having reached a critical initial mass of nodes. As mentioned earlier, in the original paper, Haas et al. claim that $k = 1$ works best for many networks. We evaluate the impact of k, by running Gossip3 on the same topologies seen in previous section for $k = 0$ and $k = 1$. For each network topology nodes run Gossip3 with those parameters that show to perform best in Figure 3.1. In particular, for $\Delta = 5m : p = 0.1, m = 1$, for $\Delta = 10m : p = 0.2, m = 1$, for $\Delta = 15m : p = 0.3, m = 1$ and for $\Delta = 20m : p = 0.6, m = 3$.

Figure 3.2a show that there is barely any improvement in terms of coverage for $k = 1$. Moreover, in dense networks ($\Delta = 5m$), due to the large number of (one-hop) neighbors, $k = 1$ results in up to twice as much message forwards compared to $k = 0$, as Figure 3.2b shows. The parameter $k = 1$ seems to benefit the latency though, as depicted in Figure 3.2c. So, when all first-hop neighbors forward a message, it propagates faster as it can spread out in all possible directions.

3.3.5 Setting the Random Assessment Delay

Another important, yet overlooked parameter in Gossip3 is the time interval nodes have to wait before deciding whether to compensate for a packet. We have seen this parameter in several broadcast protocols under the name of



(a): Coverage

(b): Forwarders

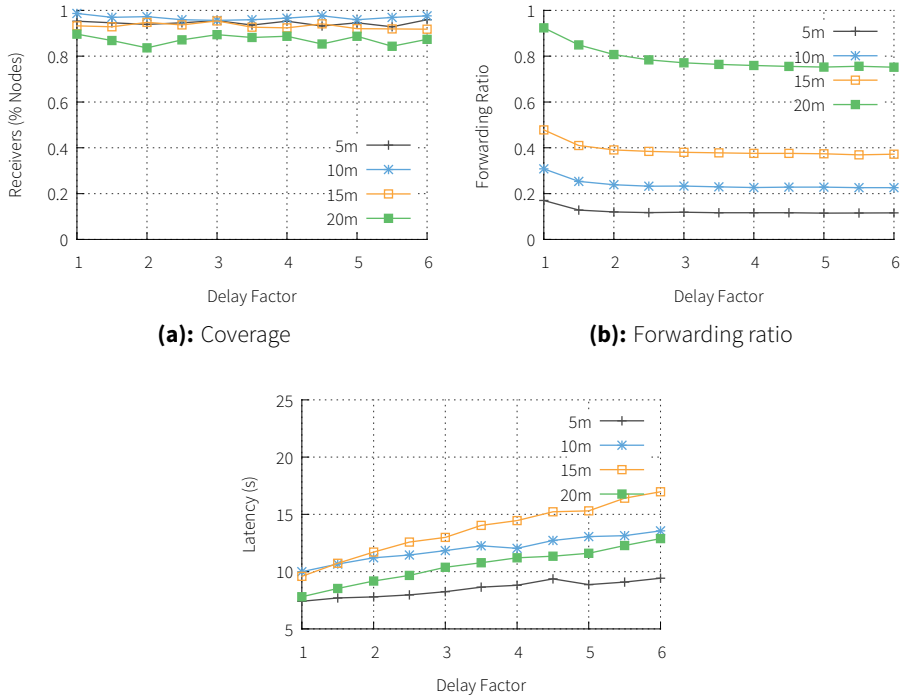
(c): Latency to reach 90% coverage

Figure 3.2: Evaluation of parameter k for uniform networks of various densities.

Random Assessment Delay (RAD). The choice of RAD is crucial to both latency and traffic generated, as pointed out by Peng and Lu [66]. Indeed, if the RAD is too short, nodes will decide too rapidly to rebroadcast, without waiting long enough to hear neighbors' probabilistic rebroadcast. So, nodes may decide prematurely that a message is not redundant, which may eventually lead to a high number of "unnecessary" rebroadcasts. On the other hand, if nodes wait too long to compensate, they may have a good estimate of the redundancy of the messages. But that could lead to a high dissemination latency.

The authors of Gossip3 suggest that the RAD should be five times the one-hop delay[2]. The one-hop delay includes the time that occurs from the

[2]Experiments in Figure 3.1 have the RAD set according to the original Gossip3.



(a): Coverage

(b): Forwarding ratio

(c): Latency

Figure 3.3: Performance of Gossip3 for various values of *delay factor*.

moment a packet is sent out from the network layer until it reaches the network layer of the receiver. The most time consuming component here is the buffering time at the MAC layer. Given that buffering time may vary significantly depending on the traffic, MAC configuration and so on, it is important to take that into account. However, it is not clear how the factor of one-hop delay (from now on we will refer to it as *delay factor*) affects the performance of message dissemination with Gossip3.

We evaluate Gossip3 with various *delay factors* ranging from 1 to 5. To this end, similarly to the previous section, we consider the same topologies running Gossip3 with parameters that have shown to perform best in our study (Figure 3.1). Figure 3.3a shows that coverage is not affected by the *delay factor*. This is not surprising, since the RAD determines the time when to compensate. so, it does not affect the (minimal) redundancy of a message, but rather its upper bound redundancy. Indeed, as depicted in Figure 3.3b,

the ratio of forwarders reaches its lowest value for *delay factors* greater than 2.5, pretty much independently on the network density. The *delay factor* impacts to a larger extent the latency of dissemination as can be observed in Figure 3.3c. Given that latency increases almost linearly with the *delay factor*, whereas the ratio of forwarders stabilizes past the *delay factor* of 2.5, it is wise not to adopt values of delay factor higher than 2.5.

3.3.6 Discussion

After evaluating the performance of Gossip3 on a somewhat realistic scenario we observe that the default parameters suggested by Haas et al. in the original paper ($p = 0.65, m = 1, k = 1$) do not perform optimally in different network densities. For example, in a network configuration with average internode distance $\Delta = 5m$, it is possible to reduce up to three times the number of forwarders while achieving the same coverage, namely when $p = 0.1$. In contrast, on a sparse network, $\Delta = 20m$, we observe that $m = 3$ can achieve a higher coverage than the default $m = 1$. So, from our study it is clear that there are no universal parameters that provide optimal performance to any network configuration. In a practical setting, the network density may not be known a priori, or may be difficult to estimate. Our goal is to come up with a single mechanism that allows nodes to adaptively adjust their behavior to function optimally in diverse scenarios."Is it possible to get optimal performance on any network density without preconfiguration?". The following section addresses this question.

3.4 Self-Configuration of Gossip3

We have shown that different network configurations exhibit different problems when it comes to information dissemination by Gossip3. In short, in dense networks, there are enough potential candidate nodes to forward data broadcast by a node, and these nodes are also well connected with each other. Therefore, high dissemination coverage comes virtually for free, and the key challenge is to avoid wasting bandwidth on redundant transmissions, which can be achieved by limiting the number of nodes that ultimately forward particular data. In contrast, in sparse networks, a node broadcasting data has few candidate nodes that can forward it further, the wireless links to those nodes are of poor quality, and the nodes are rarely connected with each other. Consequently, even if many nodes forward data, there are virtually no redundant

transmissions, and instead, guaranteeing reasonable dissemination coverage becomes the main issue.

These conflicting requirements cannot be satisfied by a single parameter configuration for Gossip3. Worse yet, a single configuration may not be optimal even for a single network, especially if the density of the network is not homogeneous. We address this problem by introducing algorithms in which each node self-configures its parameters depending on its current environment to maximize the performance of Gossip3. Our algorithm consists of two components—self-configured probabilistic forwarding and a self-configured compensation mechanism—which we discuss next.

3.4.1 Self-Configured Probabilistic Forwarding

Our first idea is to let each node dynamically adjust its probability p of forwarding data based on the number of other candidate nodes that can forward the data if necessary. One of the benefits of this mechanism is that a network can dynamically self-configure after a deployment with a probability value that is most suitable for that particular deployment. Moreover, each node can choose a different, custom forwarding probability, which may be important especially in heterogeneous networks with sparser and denser regions. Finally, any mobile node can automatically reconfigure itself when moving between sparser and denser regions of the network.

Choosing the Forwarding Probability

We have devised the self-configuration algorithm for p based on our empirical results. More specifically, using the aforementioned experimental data, we have identified those values of p for which Gossip3 achieves the best performance, that is, a maximal coverage with as few forwarders as possible. Assuming that the compensation mechanism is active ($m > 0$) and that first-hop nodes are not forced to always forward data ($k = 0$), the best performing values of p are as follows: for $\Delta < 10m$, $p = 0.1$, for $10m \leq \Delta \leq 15m$, $p = 0.2$, and for $\Delta > 15m$, $p = 0.4$. The compensation parameter is $m = 1$ for $\Delta \leq 15m$ and $m = 3$ for $\Delta > 15m$. In Figure 3.4, we plot the results from the experiments corresponding to these best performing configurations. Each point in the figure corresponds to a single node in a single experiment, with points belonging to the same experiment being represented with the same color and shape (dots, squares, crosses, etc.). Each point represents the final forwarding probability of a node as a function of the number

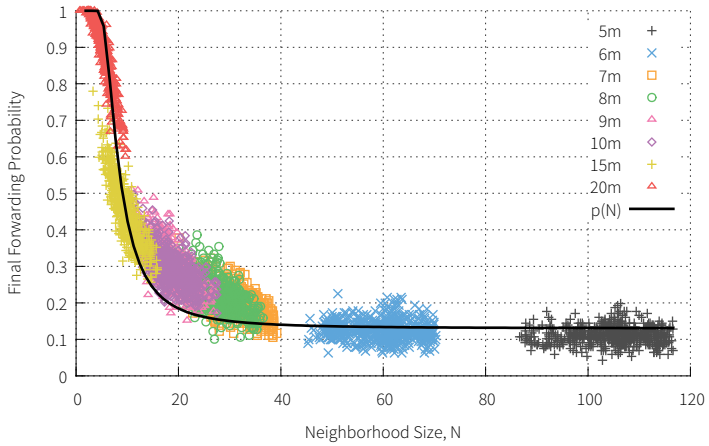


Figure 3.4: *The resulting forwarding ratio per node of Gossip3 with optimal parameters in function of neighborhood size.* Dots represent nodes and colors represent the network density.

of the node's neighbors (the local density of the network). The final forwarding probability is computed for a node as a ratio of the number of unique data items forwarded by the node to the number of unique data items received by the node during an entire experiment.

The probability thus encompasses the forwarding probability p and the probability of compensating for data dying out. Oversimplifying things, the plotted final forwarding probability for a node may be thought of as the value of p the node needs to deliver an optimal dissemination performance without triggering the compensation mechanism.

Using the empirical values we have devised a heuristic for assigning p to a node depending on the local density of the network. More specifically, we have fed the empirical values to data analysis software[3] to obtain a best-fit function. The resulting function, representing a Weibull Model, is as follows:

$$p(N) = 1 - 0.87 \cdot e^{\frac{-50}{N^{2.3}}} \tag{3.1}$$

where N denotes the number of neighbors of a node. In Figure 3.4, this function corresponds to the solid black curve.

Given the above function which denotes an optimal forwarding probability $p(N)$ we modify Gossip3 such that each node autonomously assigns its forwarding probability p according to the function. To be precise, upon re-

[3] CurveExpert Professional

ception of a new data item, a node computes its current forwarding probability for the data item as $p_{curr} = p(N_{curr})$, where N_{curr} denotes the current number of neighbors of the node. Then, with the newly computed probability p_{curr}, the node decides whether to forward the data item, just like in Gossip3.

Estimating Network Density

For the above mechanism to work, each node is required to know not only how to compute $p(N)$, but also how many neighbors, N, it has, that is, how dense the network is in its vicinity. Due to the properties of wireless communication, notably signal attenuation and transmission collisions, accurately computing network density is a nontrivial problem. For this reason, we borrow from [30] the following heuristic solution.

We augment each packet broadcast by Gossip3 with an 8-bit sequence number. By analyzing sequence numbers in packets received from node j, node i can compute the packet reception rate $PRR(i, j)$. It can then use the computed packet reception rates for all nodes k in its radio range to obtain the estimate of the network density as follows:

$$N(i) = \sum_{k \in \{nodes\ in\ i's\ radio\ range\}} PRR(i, k)$$

This computation is redone periodically to account for changes in the node's vicinity since the last computation, for example, due to node mobility or fluctuations in the quality of wireless links. While not perfect, this simple heuristic for computing network density turns out sufficient for our algorithm.

3.4.2 Self-Configured Compensation Mechanism

The second component of our self-configured algorithm is a novel mechanism for compensating for data that seem to be dying out. In the original Gossip3 protocol [27], if a node that decided not to forward a data item does not hear at least m neighbors forwarding this data item, it will forward the data item irrespective of its initial decision.

Although the authors of Gossip3 argue for using $m = 1$, our experiments provide evidence that this is not always the best configuration. In particular, in sparse networks $m = 1$ is simply insufficient, as only $m \geq 3$ provides the maximal coverage (cf. $\Delta = 20m$ in Figure 3.1, top row). In denser networks,

in contrast, $m = 3$ is an overkill as it increases the fraction of forwarders without any gain in coverage (cf. $\Delta = 10m$ in Figure 3.1, top and middle row).

Furthermore, hearing m neighbors forward data does not necessarily mean that the data is safe, because not all neighbors are equal. More specifically, a neighbor that itself has few neighbors is less likely to create enough replicas of a data item to ensure that this data item does not die out than a neighbor that itself has many neighbors. Intuitively, in sparse regions of the network, the compensation mechanism should be triggered more aggressively than in denser regions. In other words, using just the number of neighbor retransmissions as a trigger for the compensation mechanism is too crude.

For these reasons, we propose a finer-grained heuristic for triggering the compensation mechanism. To this end, we make use of the fact that nodes already compute their neighborhood size, N. In addition, we require a node to embed its value of N (up to 8 bits) in every packet it broadcasts.

The heuristic works as follows. A node i that decided not to forward a data item waits for a predefined interval. At the end of the interval:

- If i heard no neighbor forward the data item, it triggers compensation: forwards the data item.
- If i heard at least 3 neighbors forward the data item, no compensation is necessary.
- Otherwise, i looks at its own neighborhood size, N_i, and the smallest neighborhood size, N_j, among the neighbors it heard forward the data item. If N_i or N_j is smaller than a connectivity threshold, N_{min}, node i decides to compensate and forward the data item; otherwise, it does not compensate.

Among others, the above heuristic ensures two important properties. A well connected node (one with $N \geq N_{min}$) does *not* trigger the compensation mechanism if the neighbors that took over forwarding are well connected or there are many of them. A poorly connected node, in turn, triggers the compensation mechanism unless many of its (few) neighbors take over forwarding. This heuristic minimizes the number of forwarders in dense network regions, while, at the same time, guarantees sufficient forwarding redundancy in sparse regions and on the borders between dense and sparse regions. The best performing value of $N_{min} = 8$ has been identified empirically. As N_{min} is used to determine whether a node is poorly connected, it is

	$\Delta = 5m$	$\Delta = 10m$	$\Delta = 15m$	$\Delta = 20m$
p	0.1	0.2	0.2	0.4
m	1	1	1	3
k	0	0	0	0

Table 3.2: Parameters of optimal Gossip3 configuration for the considered uniform networks.

not bound to any network density. As such, the empirically identified value of N_{min} can be invariably adopted across many different network topologies.

3.4.3 Evaluation

Uniform Node Distribution

In this section we compare our self-configured algorithm with (i) the best performing parameters of Gossip3, denoted as *optimal static* Gossip3, and (ii) the default parameters of Gossip3 as Haas et al. have suggested in the original paper [27], that are $p = 0.65$, $m = 1$ and $k = 1$ (denoted as *default* Gossip3).

We start by looking at uniform topologies of various densities. More specifically, like in Section 3.3.2, we consider networks of 529 nodes placed randomly with four inter-node distances, namely, $\Delta = 5m, 10m, 15m, 20m$. The latter determines the size of the playground. The parameters of the *optimal static* Gossip3 configuration are summarized in Table 3.2. In the case of *adaptive* Gossip3 we pre-configured $k = 0$, as that is not part of the adaptive algorithm.

Before looking at the results, it is worth mentioning that we use the same traffic pattern as described in Section 3.3.2. More specifically, all nodes send continuously *dummy* packets to their direct neighbors to emulate a saturated network. For each network density the MAC layer is configured accordingly, so as to maximize the number of packets delivered per time unit. Moreover, only three nodes placed in different locations, send 100 broadcast messages each at regular intervals. All the nodes, in addition to dealing with synthetic traffic, contribute in disseminating the broadcast messages they receive.

Figure 3.5 shows the performance of the various configuration of Gossip3 in terms of coverage, message redundancy and latency. With respect to coverage our approach performs the same or better than the two static configurations, as shown in Figure 3.5a. As expected, the default parameters of Gos-

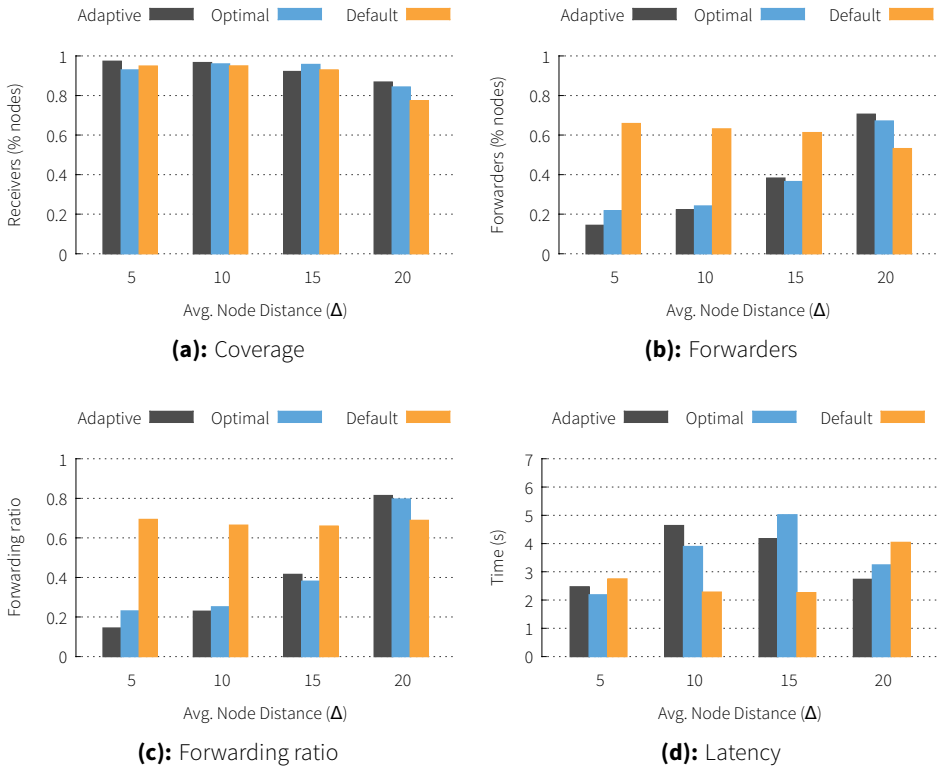


Figure 3.5: Performance of various Gossip3 configurations in uniform networks with inter-node distance $\Delta = 5m, 10m, 15m, 20m$.

sip3 perform poorly in sparse networks ($\Delta = 20m$), where coverage does not even reach 80% due to insufficient compensation ($m = 1$). However, for denser configurations, the coverage of *default* Gossip3 is comparable to *optimal* and *self-configuration*.

The major difference between adaptive and default Gossip3 concerns message redundancy. Figure 3.5b shows that adaptive Gossip3 involves the lowest number of forwarders, while the same metric remains invariable in *default* Gossip3 for most network configurations (i.e., 0.65). The network configuration with $\Delta = 20m$ exhibits a different behavior though. In this case, adaptive Gossip3 results in a higher forwarding ratio (Figure 3.5c) due to the poor node connectivity. This leads to more forwarders and, ultimately, higher coverage as compared to default Gossip3. In contrast, the pre-configured

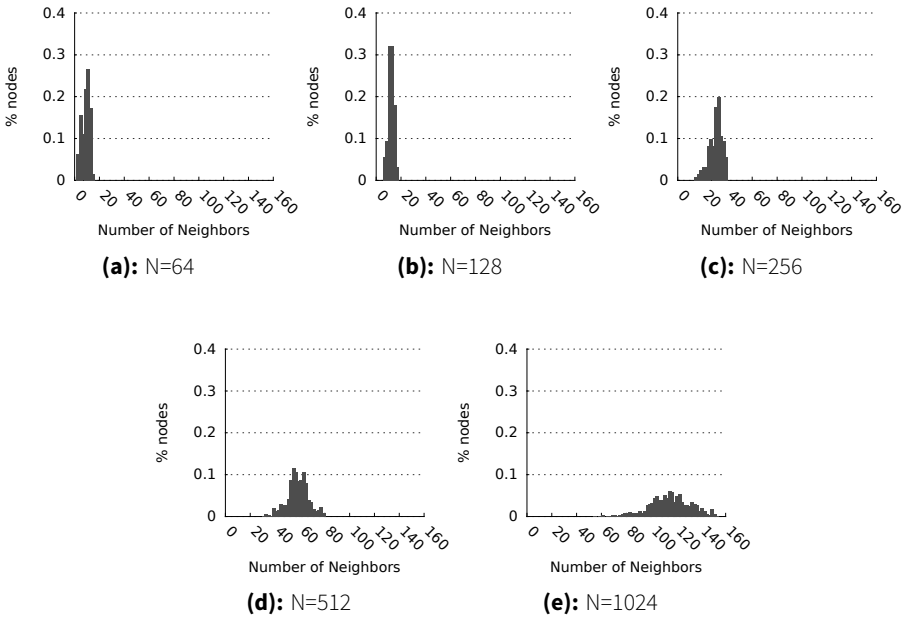


Figure 3.6: Histogram of degree distribution for various network sizes.

forwarding probability of default Gossip3, namely $p = 0.65$, leads to low coverage in the sparse configuration with $\Delta = 20m$. Due to such a low coverage, the fraction of forwarders is low too (Figure 3.5b), which is not be confused with better performance.

Finally, Figure 3.5d shows that the *self-configured* and the *optimal* Gossip3 are outperformed by the *default* configuration in terms of latency for $\Delta = 10m$ and $\Delta = 15m$. In contrast, when $\Delta = 20m$ latency is higher for the default Gossip3. This shows that dissemination latency is inversely proportional to packet redundancy.

An important observation from the experiments seen above is that our adaptive Gossip3 results in the same forwarding behavior, hence, performance, as the *static optimal* configuration of Gossip3. In the following experiments we evaluate the performance of the *self-configured* Gossip3 for various network sizes, yet with uniform nodes distribution. In particular, we fix the playground size and vary the number of nodes in it. The rest of the settings remains unchanged. The playground is fixed to 140x140m, whereas the net-

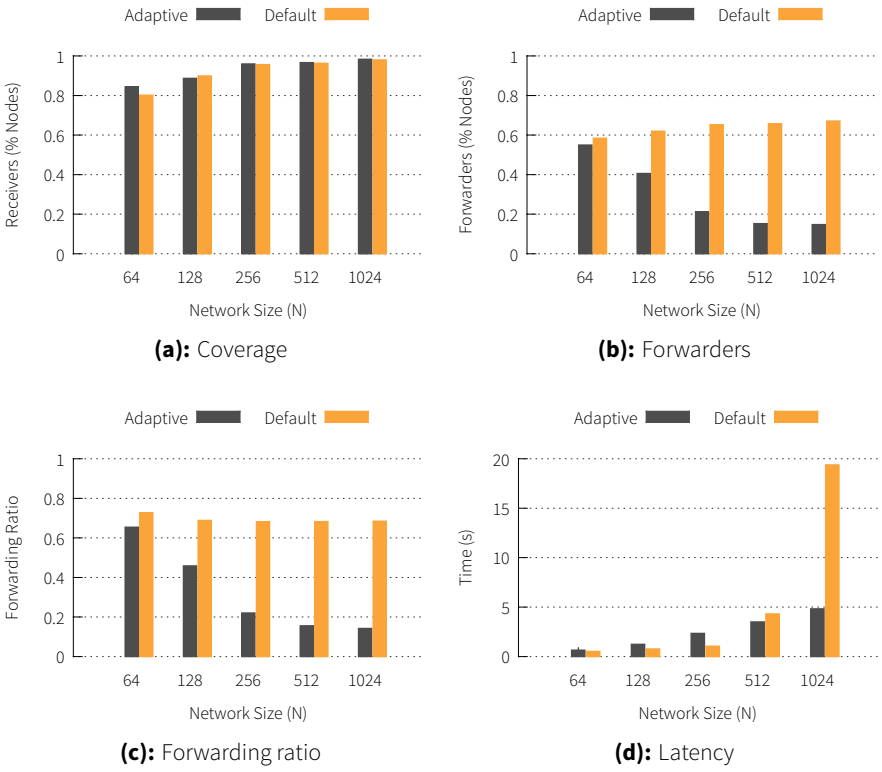


Figure 3.7: Performance of default and adaptive Gossip3 for various network sizes.

work size varies from 64 to 1024 nodes. For all network sizes the diameter is about 5 hops. Figure 3.6, depicting the distribution of nodes' degrees, substantiates the huge difference in nodes' connectivity for these networks.

Figure 3.7a shows that our self-configured approach of Gossip3 is not superior to the *default* Gossip3 in terms of coverage. Indeed, they perform roughly the same. But we notice a significant difference with respect to message redundancy (Figures 3.7b and 3.7c) and dissemination latency (Figure 3.7d). In terms of forwarders, we see the same pattern as in the previous experiments: by adapting the forwarding probability to the local connectivity of the nodes, our approach results in a much lower number of forwarders.

Moreover, in Figure 3.7d we observe a reoccurring trade-off between latency and message redundancy: low message redundancy (i.e., fewer forwarders) leads to higher latency. This observation is notably visible for $N =$

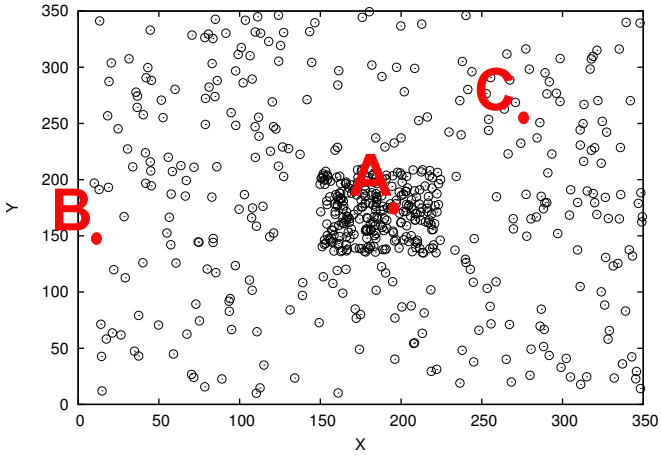
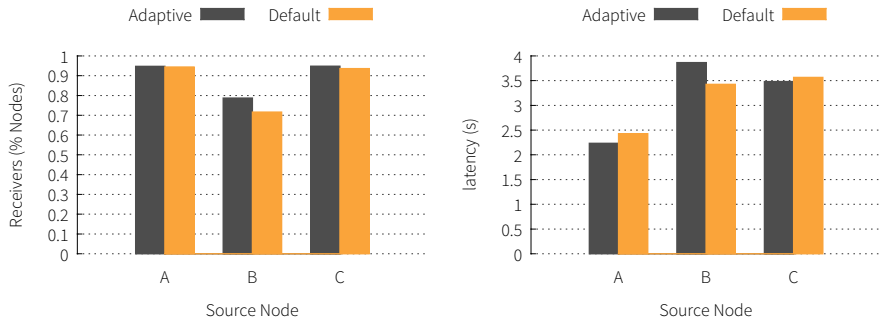


Figure 3.8: *Non-uniform network topology.* Nodes A, B and C act as sources while the rest of the nodes generate continuously synthetic packets to emulate high traffic at the MAC layer.

128 and $N = 256$–up to three times fewer forwarders results in almost twice slower dissemination than the *default* Gossip3. However, this is not always true. Too much redundancy can result in huge dissemination latency, as can be best observed for $N = 1024$ where the latency is almost four times higher than in Adaptive Gossip3. To explain this, we have to recall that a high connectivity between nodes implies that nodes receive a lot of new messages from their neighbors (~ 100). As the forwarding probability is relatively high ($p = 0.65$), a large part of the incoming messages are scheduled to be forwarded. This, directly impacts the buffering time of the outgoing packets and, hence, the overall dissemination latency. In contrast, a low forwarding probability, such as the one provided by our self-configured approach ($\sim p = 0.1$), is sufficient to provide good coverage without generating much redundancy.

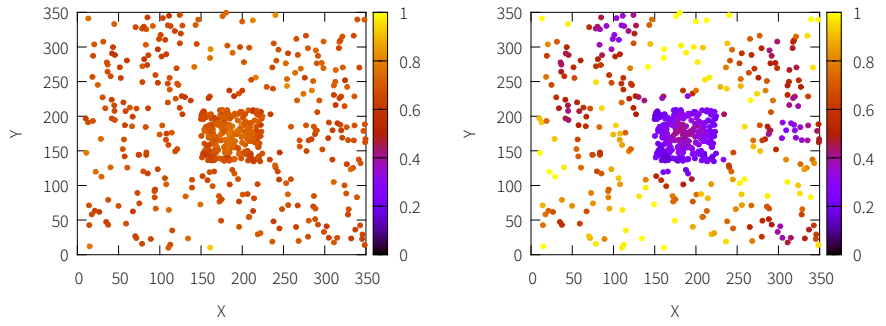
Non-Uniform Node Distribution

To assess the performance of our self-configured Gossip3 in a more realistic scenario, we consider a non-uniform topology, where nodes are highly concentrated in the center and sparser at the borders. We adopt the same traffic pattern as in the previous experiments. Three source nodes are placed in different areas, as depicted in Figure 3.8. The network consists of 564 nodes.



(a): Coverage **(b):** Latency

Figure 3.9: *Performance of default and adaptive Gossip3 in a non-uniform network.* Latency represents the average time it takes to reach 80% coverage.



(a): Gossip3 **(b):** Adaptive Gossip3

Figure 3.10: Resulting forwarding ratio of nodes in a non-uniform network represented as color-coded circles.

Figure 3.9 depicts the average coverage and latency for each node. Both default and self-configured Gossip3 perform equally with respect to coverage. The central node (A) and the node in a sparse, yet well connected area (C) reach almost full coverage, in contrast to the node in very sparse area (B). The dissemination latency of messages sent by node A is the lowest given its central location.

It would seem there is no difference in the performance of default and self-configured Gossip3 but looking at the extent of forwarding, Figure 3.10, we see a very different behavior of the two. The gradient color of each node rep-

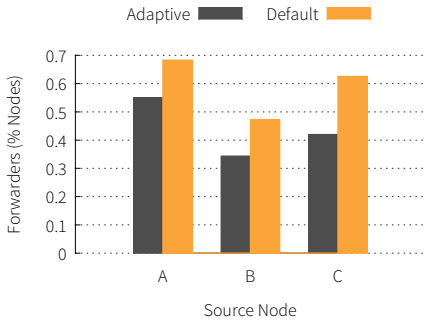


Figure 3.11: Fraction of forwarders, grouped by source node, for default and adaptive Gossip3 in a non-uniform network.

resents its ultimate forwarding probability. We observe that unlike default Gossip3, the *self-configured* approach provides a much better distribution of rebroadcasts among the nodes: sparse nodes forward at a higher probability compared to those in dense areas. Moreover, Figure 3.11 shows that overall the forwarding probability is smaller in the self-configured Gossip3 than in the default one. This implies that for achieving the same coverage, default Gossip3 would have to pump up to 50% more messages than our self-configured Gossip3. Under realistic traffic conditions, that is, all nodes sending new broadcast messages at regular intervals, forwarding messages unnecessarily can visibly affect the dissemination coverage.

3.5 Conclusions

In this chapter we revisited the well-known Gossip3 protocol [27] for message dissemination in wireless ad hoc networks. We performed extensive experimental analysis in diverse network densities, and under high network utilization, a setting not investigated for Gossip3 before. We observed that the protocol's input parameters are highly sensitive to the density of the network, and we explored this relationship, with respect to three metrics: dissemination coverage, traffic generated, and latency observed. Such a dependency between the configuration parameters and the properties of the underlying network may discourage the use of Gossip3 when the value of the parameter cannot be unambiguously determined, for example, in scenarios with mobility or unpredictable node density.

One of the major findings of this chapter is a novel algorithm that alleviates this shortcoming of Gossip3. Our algorithm determines a node's rebroadcast probability, p, in such a way that the dissemination protocol can retain its optimal performance irrespectively of the density of the network it operates in. This method results in a significant increase of the application range of Gossip3: the same configuration can operate optimally in sparse, dense, and heterogeneous networks, making the protocol more attractive for real-world deployments.

CHAPTER 4

MEDIUM ACCESS CONTROL

Thus far, we have argued that an all-to-all broadcasting scheme is necessary to ensure communication between group members in a large-scale ad hoc network. Despite the minimized message redundancy discussed in Chapter 3, broadcast protocols demand high network resources, especially considering that all nodes send new messages to be broadcast on a regular basis. A massive presence of nodes that attempt to transmit at the same time can become a problem in broadcast mediums where nodes can only transmit one at a time. Regulating the transmissions of tens, or even hundreds of competing nodes without a central entity, in a fair and collision-free way can be particularly hard. The protocol that determines who goes next in a multi-access channel belongs to a sublayer of the data link layer called the Medium Access Control (MAC layer of simply MAC).

MAC plays a key role in the overall system performance as it directly impacts several aspects of the network. In particular, it is not only responsible for preventing collisions, but also for saving energy, minimizing end-to-end latency and providing fairness in channel usage. Our system is no exception: consider a dense multi-hop network where nodes contend for transmitting packets. When the MAC allows for high channel access rate, chances are that simultaneous transmissions will occur, which lead to packet collisions. It is often followed by retransmissions of the lost packets. There are two im-

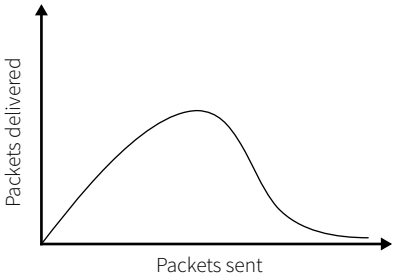


Figure 4.1: Representation of the congestion problem at the MAC layer.

mediate consequences to this. First, the time required to reach a multihop destination grows significantly, and second, energy is wasted, as nodes may have to retransmit several times before a packet actually gets delivered. In contrast, a conservative MAC where nodes access the medium at a low pace may incur much higher latencies despite a good packet delivery ratio.

The problem of MAC congestion is illustrated in Figure 4.1. When the rate of transmitted packets is low, all sent packets are delivered and the number delivered is proportional to the number sent. As the rate of transmissions increases, so do the delivered packets up to a maximal point. The rate of transmissions corresponding to such a point represents an optimal configuration. But increasing the transmissions too far beyond the channel capacity, severely degrades performance up to a complete collapse where almost no packets get through. By and large, whether the rate of packet transmission is too low or too high with respect to the channel capacity, the resultant number of delivered packets will not be optimal.

This chapter presents a study of medium access control in ad hoc networks. The goal is to devise a mechanism that reaches optimal packet delivery for a variety of network configurations. Section 4.1 discusses an overview of existing MAC protocols and points out the various trade-offs of random and scheduled access schemes. In Section 4.2 we consider a simple random access scheme and evaluate the performance of Gossip3 under various traffic loads. We show the impact that congestion at the MAC layer has on the upper network layer and the interdependency of the two layers. In Section 4.3 we optimize the performance of the MAC protocol by properly configuring the parameters for a variety of network configurations. Section 4.4 discusses possible techniques for self-configuration of the MAC parameters and finally we conclude in Section 4.5.

4.1 Overview of MAC Protocols for Ad Hoc Networks

The design of a MAC protocol depends to a great extent on the target application. As ad hoc networks serve many diverse applications, a plethora of MAC protocols have been proposed to match their specific requirements. For example, a typical monitoring application in sensor networks may put higher priority on saving energy because it is required to work unattended for a long time. On the other hand, wireless LANs or real-time applications in vehicular ad hoc networks are focused on achieving maximal bandwidth allocation and low latencies. To provide for such contrasting goals the MAC has to pursue very different strategies. In the following section we study two main techniques of medium access and discuss how they fit the requirements of our system.

4.1.1 Random Medium Access

Random access protocols, also known as contention-based protocols, are based on Carrier Sense Multiple Access (CSMA) mechanisms. Nodes sense the channel to determine whether another transmission is in progress before trying to send. If the channel is idle, nodes transmit their own packet, otherwise they back off for a time interval that is randomly picked within a contention window. The CSMA mechanism is simple, robust and offers a lot of flexibility. In fact, no synchronization or topology information is required and nodes may join or leave the network effortlessly. Due to these properties, the contention-based paradigm is widely adopted in wireless networks.

The advantages provided by the carrier sensing mechanism come at a cost. Nodes cannot sense the channel beyond their communication range. This, gives rise to the well-known *hidden terminal problem*. With reference to Figure 4.2 we explain it as follows. Node B is within the range of nodes A and C, but A and C cannot hear each other. Node A starts transmitting a packet to B. Node C also has to transmit. It cannot detect the ongoing transmission of node A and starts a transmission as well. This will cause a collision at B. A common technique for alleviating the hidden terminal problem is adopting a so-called Request-To-Send/Clear-To-Send signalling before any transmission. More specifically, when node A has a packet to send to node B, it starts by sending an RTS to B. All nodes that receive such RTS suspend

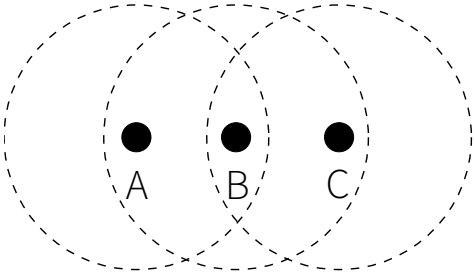


Figure 4.2: Hidden terminal problem.

their transmissions, whereas B responds with a CTS if no other activity is scheduled. The CTS message sent by B blocks its immediate neighbors from transmitting, allowing, thus, node A to transmit virtually without collisions. However, this approach does not prevent all collisions. For example, when a node fails to receive an RTS or a CTS, it will ignore the fact that a communication is taking place. Consequently, it will respond to other requests to transmit or even send its own RTS and, thus, risk incurring collisions. Moreover, collisions can occur between control packets when two or more nodes transmit an RTS at the same time after detecting the channel to be idle.

The standard IEEE 802.11 is widely used in a number of environments such as wireless LANs, ad hoc networking and vehicular networks. It can be adopted for unicast as well as broadcast communications. The first relies on random medium access with RTS-CTS-ACK handshake for collision avoidance and reliable transmissions. This proves effective in minimizing collisions, but on the other hand imposes a large overhead of control packets. Whereas in broadcast mode none of the reliability mechanisms is used and, as such, nodes can start a transmission any time they have packets to send. The performance of this mechanism depends on a number of factors, such as, the density of contenders, the size of the packets and the configuration of the MAC parameters.

CSMA-Adaptive Rate Control [98] takes a different approach on collision avoidance. Assuming that information flows from sensors to the gateway in a tree-like fashion, Woo et al. [98] introduce an implicit notification scheme that allows nodes to prevent the hidden terminal problem without extra control messages. In particular, after a node hears a message from its parent, it estimates the time it will take to the grandparent to retransmit the same message, and refrain any transmission during this time interval. Similarly, acknowledgments are also implicitly inferred: a node concludes that a packet

was successfully delivered, when it is retransmitted by its parent node. The major drawback of such a notification scheme is that it requires a specific information flow, from the leaves up to the root, which restricts its applicability.

In Sift [33] nodes competing for the channel use a non-uniform probability function to pick a slot within a fixed-size contention window. Such skewed distribution gives nodes higher preference towards the end of the window leading to one lucky node winning the competition among the many contenders. Nodes do not need prior knowledge about the neighborhood size. Instead, they have a shared belief of the current number of neighbors, which starts off at some large value and decreases after every slot in which no node transmits. If no node starts to transmit in the first time slot in the window, then each node increases its transmission probability exponentially for the next time slot. Since it makes no assumption on network size, Sift easily adapts to various densities. In addition, it greatly reduces collisions among a number of contenders and achieves low latency given that a winner is decided in the early slots. RTS-CTS signaling is used for preventing hidden terminal collisions. Moreover, Sift requires a network-wide time synchronization to provide the same view of the time-slotted contention window to the contending nodes.

Low-power MAC protocols are prevalent among traditional sensor networks. Such protocols allow nodes to save energy by turning off their radio at regular intervals. For example, in S-MAC [102] nodes synchronize their active times, during which they contend for channel access through a CSMA protocol. By alternating between sleep and active mode, nodes can save up to 90% of energy, which allows sensor networks to operate unattended for months or even years. The network throughput, though, is significantly reduced.

B-MAC [70] is also an energy-aware protocol widely used in sensor networks and currently adopted as the default MAC in TinyOS [2]. Unlike many energy-aware protocols, it provides higher flexibility as it does not require nodes to synchronize with each other. B-MAC relies on Low-Power Listening to save energy and it introduces a fine granularity channel sensing. The idea behind low-power listening is that packets are preceded by a long preamble, as a kind of busy tone to alert potential receivers about the upcoming packet. Nodes wake up at regular intervals just to sense the channel and, when a busy tone is intercepted, they keep their radios on to listen to the packet which comes after the preamble. The preamble is longer than a sleep interval, which guarantees that receivers will wake up regardless of

their phase of sensing the channel. RTS-CTS handshakes are employed to alleviate the hidden terminal problem.

4.1.2 Schedule-Based Medium Access

In schedule-based MAC protocols, nodes coordinate with each other to establish transmission schedules that allow them to transmit without collisions. Such a mechanism does not require extra control messages to prevent collisions. A common implementation of this scheme is Time Division Multiple-Access (TDMA), which divides time into frames and frames into slots, which in turn can be allocated by nodes to transmit their packets collision-free. Several protocols have been proposed for assigning slots to nodes.

Lightweight MAC [93] uses a distributed slot selection mechanism based on two-hop neighborhood information. Each node owns a time slot within a frame, in which it always transmits a header, optionally followed by a payload. Nodes include in the header a bitset detailing which slots are occupied by them and by their one-hop neighbors. By OR-ing the occupancy bitsets of all headers received within a frame, a node can easily determine which slots are still available in its two-hop neighborhood before selecting its own. The number of nodes in any two-hop neighborhood should not exceed the number of slots in a frame, which is predetermined before deployment. Since nodes may allocate only one slot in each frame, LMAC is not flexible to traffic conditions and to various network densities. Moreover, maintaining transmission schedules between nodes in a dynamic network can be challenging: by the time nodes establish their schedules, the neighborhood may have changed and the process has to start over running the risk of never converging.

AI-LMAC [19] gives more flexibility to the original LMAC, as it allows nodes to allocate more slots within one frame based on traffic conditions. But AI-LMAC assumes that the flow of messages follows a tree-like pattern, where parents are responsible for fairly allocating slots to their children according to the load they report.

To deal with node mobility Mank et al. propose MLMAC [52], a topology adaptive version of LMAC. First, the synchronization process can be started by any node that does not belong to a synchronization cluster. As topology changes, the process of slot allocation remains adaptive. Since any node can start the synchronization process, MLMAC prevents coexisting synchronizations by adopting the fields *identity of synchronization* and the *age of synchronization* contained in the packet header. When two synchronizations with

different *identity of synchronization* are detected, *age of synchronization* field is used to merge them into the younger or the older one, depending on the synchronization strategy.

GMAC [22] supports mobility and various network densities. In a similar way to MLMAC [52], age and identity of synchronization clusters are used to decide how to merge coexisting clusters. Unlike many TDMA protocols that assign specific slots to nodes within a frame, at each frame nodes pick randomly their slot within an active period. As this approach exposes nodes to collisions, the length of the active period is adjusted based on an estimate of the neighborhood size. GMAC provides a way in-between schedule-based and random access. The schedule-based scheme keeps the active periods synchronized, so nodes can sleep and wake up synchronously. Whereas, the random access within the active period with slotted aloha, along with the adaptive length of the active period, gives GMAC robustness and adaptivity to mobility and various network densities. However, since no physical carrier sense is performed before picking a slot randomly, collisions can occur when two or more nodes aim for the same slot.

Z-MAC [77] addresses some limitations of schedule-based schemes, such as time-varying traffic loads and slot assignment failures. It adopts a hybrid approach by relying both on TDMA and CSMA schemes. In particular, by means of an efficient channel scheduling algorithm, DRAND [76], a slot is assigned to each node within a frame in a conflict-free way. Any node assigned to a time slot becomes the owner of that slot. But unlike TDMA, a node may transmit during any time slot. Before a node transmits during a slot, it performs carrier sensing and transmits a packet when the channel is clear. However, at any slot, its owner has a higher priority to transmit over the other nodes. Such priority is implemented by adjusting the initial contention window size, ensuring that the owner of a slot gets earlier chances to transmit. The schedule-based slot allocation has the advantage that no RTS-CTS are required for preventing collisions. Instead, in each slot collisions are avoided by having owners winning contention earlier than non-owners. However, Z-MAC relies on the synchronization between nodes, which can be costly, especially when the topology changes frequently.

4.1.3 Discussion

Table 4.1 shows a comparative summary of the MAC protocols in ad hoc networks as discussed above. Energy efficient protocols are hardly adaptive to changes in topology and density. Some of them, like B-MAC, GMAC and

	Type	Collision Avoidance	Energy-aware	Data rate	Adaptive to mobility	Adaptive to density
IEEE 802.11	CSMA	RTS-CTS/ None	No	High	Yes	Yes, limited
Sift	CSMA	RTS-CTS	No	High	No	Yes
CSMA-ARC	CSMA	Implicit	No	Low	No	Yes
B-MAC	CSMA	RTS-CTS/ Implicit	Yes	Low	Yes	Yes
S-MAC	CSMA	RTS-CTS	Yes	Low	No	No
LMAC	TDMA	Scheduling	Yes	Fixed	No	No
MLMAC	TDMA	Scheduling	Yes	Fixed	Yes	No
GMAC	TDMA	None	Yes	Low	Yes	Yes
Z-MAC	TDMA/ CSMA	Scheduling	Yes	Low	No	No

Table 4.1: Comparative summary of MAC protocols.

MLMAC are well adaptive to topology changes, but that feature comes at the expense of much lower bandwidth. On the other hand, CSMA protocols can, in general, cope better with topology and density variations, but that comes at the expense of little or no concern for energy. Moreover, to avoid collisions, additional control messages are necessary.

For our system we require a MAC protocol that is robust, able to adapt to various densities and that can withstand continuous topology changes. Clearly, schedule-based schemes are ruled out, while a CSMA mechanism appears to be a better fit for our system given its resilience to topology changes and ease of deployment. In the rest of the chapter we show to what extent a simple CSMA protocol fits these requirements and what the trade-offs are.

It is important to point out that although we aim at low-power devices, we do not target energy-aware MAC protocols. While energy is an important

aspect in wireless networks, our study focuses on scenarios where devices have to run for a limited time, e.g., 12 hours per day. Throughout their lifetime nodes keep the radio on trying to maximize the number of messages delivered.

4.2 Impact of MAC on Message Dissemination

In carrier sense MAC protocols, typically the frequency of accessing the medium has a direct impact on the quality of connection between nodes. This may in turn affect the behavior of the routing protocol. In this section we explore the way the MAC and the network layer influence each other. In particular, we look at the role of the MAC layer in message dissemination. To this end, we consider a simple scenario where all nodes inject a number of messages to be broadcast through Gossip3. By varying the message insertion rate (i.e., the offered load) and the CSMA configuration at the MAC layer, we observe how the dissemination performance is affected and how the MAC and network layer impact each other.

4.2.1 Simulation Settings

We adopt the same simulation settings as in Chapter 3, which includes a MiXiM simulation framework with the physical and MAC implementation of the standard IEEE 802.15.4. A detailed configuration of the application, network and MAC layer is summarized in Table 4.2, whereas below follows a short description of the layers considered.

Application Layer

The goal of the application layer is to send new broadcast messages at various rates. Each node sends 100 messages throughout the whole experiment. The rate at which these messages are sent is determined by ρ which refers to the total number of new broadcast messages inserted by all nodes every second. So, each node sends a new message every $s = N/\rho$ seconds, where N is the total number of nodes in the network. The application layer passes the new data packet to the underlying network layer.

	Parameter	Value
Topology	Network Size (N)	100
	Nodes Distribution	random
	Network Diameter	4
APP Layer	Number of Messages	100
	Message Generation Rate (ρ)	1-100 mps
NET Layer	Protocol	Gossip3
	Fwd. Probability (p)	0.3
	Compensation factor (m)	1
	Flooding Hops (k)	0
MAC Layer	Protocol	CSMA exp. back-off
	MinBE	3, 8

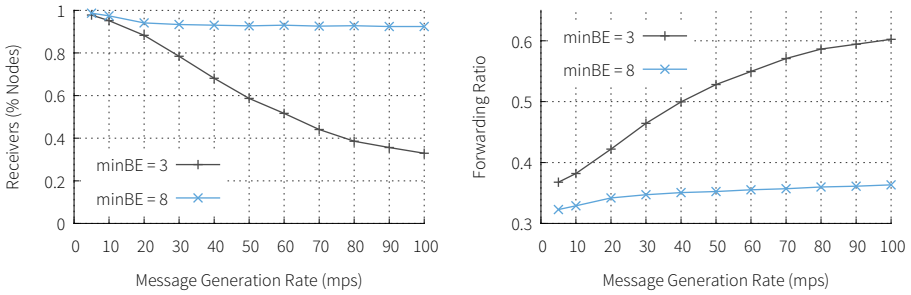
Table 4.2: Experimental settings divided by protocol stacks

Network Layer

The network layer, which is responsible for propagating messages, runs Gossip3 for messages received by the application layer as well as for those received by other nodes. Gossip3 is configured with parameters that resulted to be optimal for the chosen network density, i.e., $\Delta = 15m$. The list of Gossip3 parameters is specified in Table 4.2. Messages that are decided to be propagated by Gossip3 are passed to the data link layer for being actually transmitted.

MAC Sublayer

As part of the data link layer, the MAC is responsible for scheduling when to air packets. More specifically, messages received from the network layer are stored in a buffer while waiting to be transmitted. We have set the maximum buffer capacity to 400 packets. The MAC layer runs a simple CSMA protocol with *binary exponential back-off*. The later refers to the way the CSMA protocol computes the interval when to access the medium. In particular, a node senses the channel before airing a packet. If the channel is idle the packet is transmitted, otherwise, the node *hello*s off for a certain interval. Such interval is computed by picking a random number of time slots within a *contention window* with length $2^{BE} - 1$, where BE is the back-off exponent and a time



(a): Coverage **(b):** Forwarding Ratio

Figure 4.3: *Performance of Gossip3 configured with optimal parameters* Evaluation of two different MAC configurations in function of message generation rate.

slot consists of $320\mu s$ (for IEEE 802.15.4 [3]). The back-off exponent, BE, is initialized to a certain value, $minBE$, for each packet under transmission. After every back-off, the BE is incremented by one. This means that the contention window grows exponentially after every back-off. For each packet there is an upper bound of five back-offs, after which the packet is dropped.

When the MAC deals with a continuous flow of messages to be aired, the starting back-off interval, determined by $minBE$, becomes crucial. It determines the degree of contention between nodes. To keep things simple we evaluate only two values of $minBE$, 3 and 8. The first is the default value for the standard IEEE 802.15.4, whereas the second is an arbitrarily large value which provides a very different contention and good dissemination coverage.

4.2.2 Network Layer Perspective

We start off by looking at the coverage as a function of the message generation rate, which directly impacts the offered load in the network. Figure 4.3a shows that for $minBE = 3$ the coverage is strongly affected by the offered load, whereas for $minBE = 8$ the influence of traffic on the performance of Gossip3 is almost negligible.

Figure 4.3b shows the effect of the compensation mechanism in Gossip3. For $minBE = 3$ it is much more pronounced especially as the offered load increases, whereas for $minBE = 8$, Gossip3 compensation stays low. To understand this figure, it is probably worth recalling that Gossip3 rebroadcasts new incoming messages with an initial probability $p = 0.3$, but it allows to

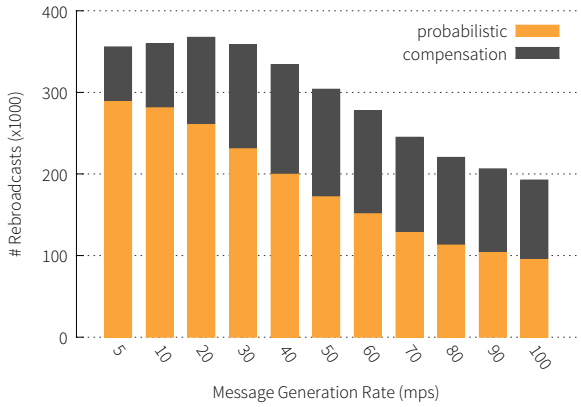
rebroadcast a message that was not rebroadcast at first if that is not received back by at least one neighbor. So, the effective rebroadcast ratio of the nodes is, in general, higher than the initial rebroadcast probability $p = 0.3$, but for $minBE = 3$, such an increase is much higher than for $minBE = 8$.

For $minBE = 3$, Gossip3 compensation works well for offered loads up to 20 messages per second. In fact, the coverage reached for this configuration is comparable to $minBE = 8$. On larger loads, the performance of Gossip3 drops sharply, while compensation increases. It is not hard to see that as packets are being lost, Gossip3 compensates for them by rebroadcasting more eagerly.

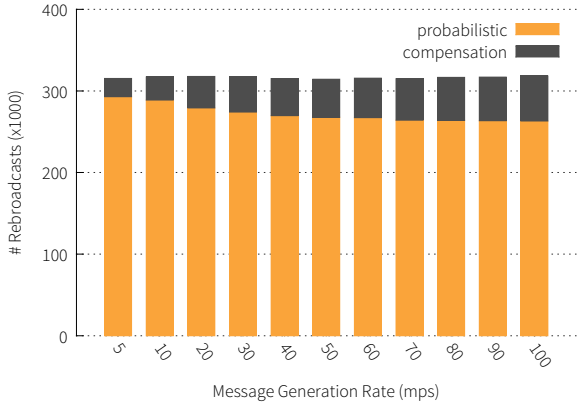
Figure 4.4 gives a closer view to what extent congestion at the MAC layer and compensation at Gossip3 are interrelated. Each bar depicts the total number of rebroadcasts from all nodes. We illustrate the behavior of Gossip3 by looking at the rebroadcasts in terms of probabilistic and compensation ones.

A first look at the two figures shows a distinct behavior between the two configurations of CSMA in the way Gossip3 reacts to the offered load. For $minBE = 3$ (Figure 4.4a), as the offered load increases, so does compensation traffic, to the point where it makes up the same amount as the probabilistic one. Moreover, it is interesting to see that an offered load greater than 30 messages per second results in reduced traffic handled by the nodes. This means that in congested networks, the flooded messages fail to reach many nodes, which implies that the overall amount of messages nodes have to rebroadcast goes down. In contrast, Figure 4.4b shows that for $minBE = 8$ compensation increases gracefully, whereas the total number of messages nodes rebroadcast does not vary.

It would seem that $minBE = 8$ is a better choice than $minBE = 3$ for this network configuration especially for offered loads greater than 20 mps. But a quick look at Figure 4.5 shows the downside of $minBE = 8$: the message propagation latency increases tremendously. In this figure we plot the average time it takes messages to reach 90% coverage. Although very few messages manage to reach 90% coverage when $minBE = 3$, propagation latency would be insignificant for such a configuration. In contrast, for larger values of $minBE$, the messages are buffered for longer time intervals, affecting this way the overall propagation latency.



(a): minBE = 3



(b): minBE = 8

Figure 4.4: *Distribution of rebroadcasts in function of message generation rate.* In Gossip3 newly received packets are either rebroadcast probabilistically, at first, or, shortly after, if found to be not redundant enough. We have adopted Gossip3 with optimal parameters, p=0.3,m=1,k=0 and network size $N = 100$.

4.2.3 MAC Layer Perspective

We observe now the behavior of the MAC layer for the same experiments. In particular, we look at the total number of messages transmitted, collided and overflown for various offered loads. A high number of collisions is a good indication of channel deterioration, but to determine whether a transmitted

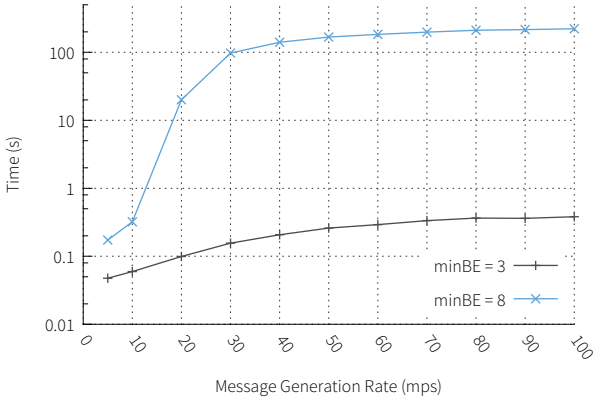
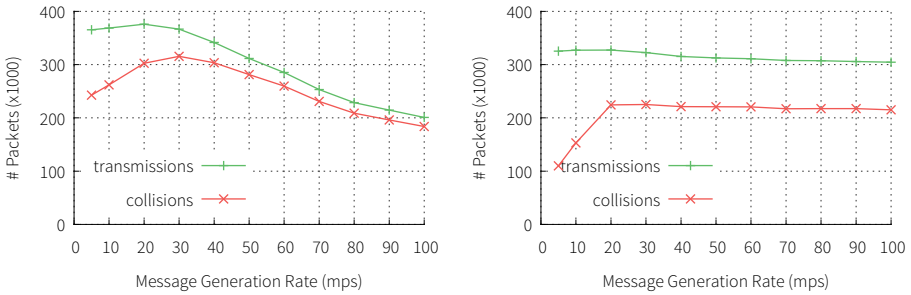


Figure 4.5: Latency of Gossip3 to disseminate messages to 90% of the nodes.

packet collides with others, it is not straightforward. Indeed, collisions occur at the receiver, so, it is common that the same packet is correctly received at some nodes while it collides at others. Although we cannot directly relate the number of collisions with the application-level performance, it provides an interesting perspective when considered as a relative measurement for various traffic loads.

Figures 4.6a and 4.6b illustrate collisions and transmissions respectively for $minBE = 3$ and $minBE = 8$. We notice that $minBE = 3$ induces more transmissions than $minBE = 8$ for low offered loads (i.e., up to 40 mps). For offered loads higher than 40 mps, the number of transmissions drop sharply for $minBE = 3$, while almost all transmitted packets appear to collide at least once. This suggests that due to collisions, messages do not reach far from the originator nodes, which brings down the overall number of transmissions.

It is interesting to see the behavior of $minBE = 8$. As the offered load increases so do collisions. This trend occurs up to 20 mps, after which both transmissions and collisions stay around the same values. This suggests that at 20 mps the network becomes saturated. Figure 4.7 backs up this observation. It shows the total number of dropped packets as result of buffer overflow at the MAC layer. Clearly, for $minBE = 3$ there are no dropped packets because the packet queueing time is very short, but for $minBE = 8$ the number of dropped packets starts to increase as the offered load reaches 30 mps. So, at 20 mps the network is saturated, whereas for higher offered loads packets begin to be dropped as result of buffer overflowing. However, it is



(a): minBE=3 **(b):** minBE=8

Figure 4.6: *Number of transmissions and collisions at the MAC layer* Evaluation of two different MAC configurations in function of message generation rate.

worth mentioning that the number of dropped packets is low enough such as not to affect the application-level performance (coverage).

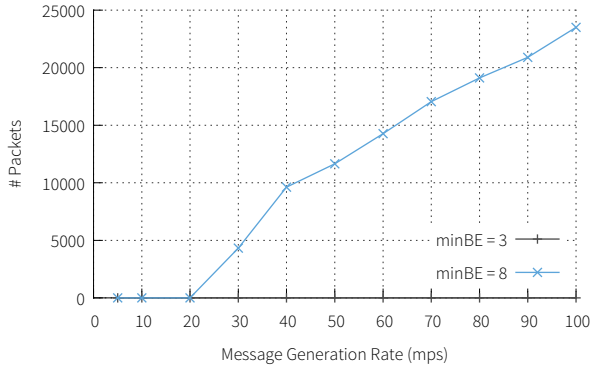


Figure 4.7: Number of packet overflow at the MAC layer in function of message generation rate.

4.2.4 Discussion

The MAC layer and the network layer mutually impact each other. We have seen that for high offered loads Gossip3 can rebroadcast up to twice as much when CSMA is configured with a small $minBE$. This, in turn, increases even more contention at the MAC layer. So, setting an appropriate channel access rate at the MAC layer is essential for providing an optimal performance at the

network layer. However, a "slow" MAC can increase significantly the latency and even cause message overflow. It is thus crucial to tune the MAC properly so as to maximize the number of packets handled by the nodes.

4.3 Optimizing MAC in Static Ad Hoc Networks

Keeping collisions under control is crucial to preserve the performance of the MAC layer in presence of continuous traffic. Although carrier sensing techniques can detect if the channel is busy before any transmission, they cannot prevent collisions that derive from concurrent transmissions outside the carrier range, i.e., caused by the hidden terminal problem. Things get even more critical in dense networks. Typically, data intensive wireless networks adopt RTS-CTS signalling to prevent hidden terminal collisions. But that would be unsustainable for broadcast communications.

As an alternative, a simple way to keep collisions under control relies on assigning nodes a suitable medium access rate that reduces the probability of collisions while delivering a good deal of packets per time unit. In Section 4.2.1 we have seen that $minBE$ determines the medium access rate. Another important parameter is the medium access algorithm. The most common ones are the binary exponential back-off (BEB) and the constant back-off. In the first algorithm the contention window increases exponentially after every back-off, whereas in the second, as the name suggests, the contention window does not vary.

In this section we explore through experimental simulations the impact that the aforementioned CSMA parameters have on the performance of the MAC layer. To this end, we consider a number of network configurations. We start off by introducing some performance metrics of the MAC layer.

4.3.1 Metrics

Based on the observations above, we propose a number of measures to evaluate the performance of the MAC layer, which are *network goodput*, *packet delivery fairness* and *packet latency*. We define *network goodput* as the total number of packets that are delivered by all the nodes, in a unit of time. Since we target networks that rely on broadcast communications, it is not straightforward to determine whether a packet was successfully delivered. In particular, due to signal attenuation, nodes located nearby a sender have much higher

chances of receiving a packet as compared to others located at the boundaries of the transmission range. Also, as the load of the network increases, the connectivity between nodes varies dramatically. Considering this, in order to assess the packet delivery success of the nodes, we first evaluate the optimal link quality between nodes for a given network topology. So, in an unloaded network, we characterize the connectivity strength between each pair of nodes by computing the transmission success ratio on a number of packets transmitted by each node. More specifically, we define the optimal link quality between any two nodes i and j:

$$Q_{ij}^{opt} = \frac{RX_{ij}}{TX_i} \tag{4.1}$$

where RX_{ij} is the number of packets node j receives from i, whereas TX_i is the number of packets transmitted by node i. To measure the performance of a MAC configuration, we compute the packet delivery ratio between any pair of nodes i and j relative to the optimal link quality between them, Q_{ij}^{opt}. So, the relative link quality between any two nodes can be expressed as follows:

$$q_{ij} = \frac{Q_{ij}}{Q_{ij}^{opt}} \tag{4.2}$$

where $Q_{ij} = \frac{RX_{ij}}{TX_i}$ is the absolute packet delivery ratio of node i to j, whereas Q_{ij}^{opt} is the optimal link quality between i and j measured under low network load. The *network goodput*, G, comprises the goodput of each node in the network.

$$G = \sum_{i \in N} G_i \tag{4.3}$$

To compute the goodput of a node, G_i, we estimate the number of its delivered packets, that is, the product of transmitted packets and the average delivery ratio of that node. So, the goodput of node i can be expressed as follows:

$$G_i = TX_i * \overline{q_i} \tag{4.4}$$

where $\overline{q_i}$ is the average delivery ratio of node i measured over the number of neighbors $n(i)$.

$$\overline{q_i} = \frac{1}{|n(i)|} \sum_{j \in n(i)} q_{ij} \tag{4.5}$$

To measure *packet delivery fairness* among nodes, Φ, we adopt the Jain fairness index [32], which is a widely used fairness indicator in computer networks. In particular, the Jain fairness index, is bound between 0 and 1, where 1 indicates that the system is fair to all nodes and 0 refers to a totally unfair system. In our experiments we measure the fairness relatively to the average rate of packets delivered (i.e., goodput) by each node:

$$\Phi = \frac{(\sum_{i=1}^{N} r_i)^2}{N * \sum_{i=1}^{N} {r_i}^2} \tag{4.6}$$

where r_i is the average rate of goodput of the i-th node:

$$r_i = \frac{G_i}{simulation \quad time} \tag{4.7}$$

Finally, to measure the *packet latency* we compute the average time it takes for packets to reach immediate neighbors. Such latency includes the buffering time at the MAC layer of the sender and, in our scenario, it represents the minimum buffering time. In the experiments that follow we emulate continuous traffic at the MAC layer of each node, by generating a new packet after every transmission. This way nodes have always packets to transmit, similarly to the scenario of high traffic we considered in Section 4.2.3.

4.3.2 Uniform Grid Distribution

To understand the impact of CSMA parameters we start off by considering very regular topologies of various densities, where nodes are arranged in a square grid fashion equidistant from each other, and the area is modelled as a torus to eliminate the border effect. In particular, we consider networks of 529 nodes arranged on a 2D plane in 23 x 23 rows, where the inter-distance between nodes includes $\Delta = 5, 10, 15, 20m$. The simulation environment is the same as described in Section 4.2.1.

Each node runs CSMA with various configurations of $minBE$ and back-off techniques. More specifically, the values of $minBE$ range from 7 to 12.5, whereas the back-off techniques include constant and exponential back-off. These values of $minBE$ provide a comprehensive set of contention windows size for studying the impact of CSMA on the network densities we consider. Larger or smaller values would not provide additional useful insight for the purpose of our study, since we are looking for values that maximize the goodput.

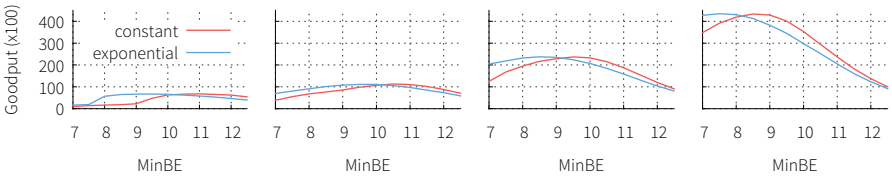


Figure 4.8: Network goodput for grid topologies.

The simulation duration is set to 20 seconds for all the experiments. Such a time span was chosen as a good tradeoff that provides results within reasonable times for values of $minBE$ as low as 7 and as large as 12.5[1]. Throughout the experiment, each node contends for the channel continuously, since a new packet is generated after every transmission, making sure that the network is fully loaded.

We start by showing the impact of CSMA parameters on *network goodput*, Figure 4.8, for the different network densities. Although the maximal achieved goodput varies significantly for different network densities, we can clearly see a bell-shaped curve, which is more or less pronounced in each case. This suggests that network goodput is maximal around certain values of $minBE$ and such values differ for various network densities.

Both exponential and constant back-off techniques provide the same maximal goodput that is obtained for different values of $minBE$. However, the exponential back-off approach can deal better with congestion, i.e., small values of $minBE$, because the contention window is doubled after every back-off. This explains the reason the bell shape is slightly skewed to the left for the exponential back-off.

Figure 4.9 shows that *packet latency* is increases with the $minBE$. Indeed, as the contention window size increases, so does the latency. The latency curve does not vary much for different densities, except when the inter-node distance is $\Delta = 5m$. This suggests that in presence of congestion packets queue up for longer time intervals given that the likelihood of back-off is higher. Given that the contention window doubles after every back-off, the exponential back-off approach results in higher latencies in general, and most notably for dense topologies, such as $\Delta = 5m$.

[1]The running time of an experiment with $\Delta = 5m$, $minBE = 7$ and simulation time 20 seconds is approximately 94 hours.

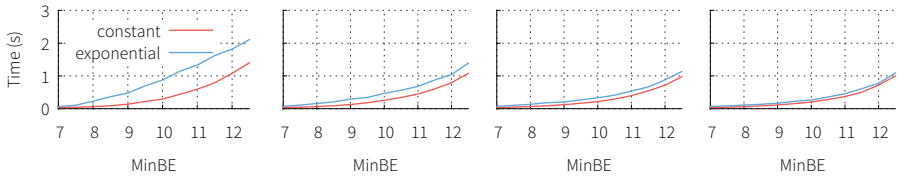


Figure 4.9: One-hop packet latency for grid topology.

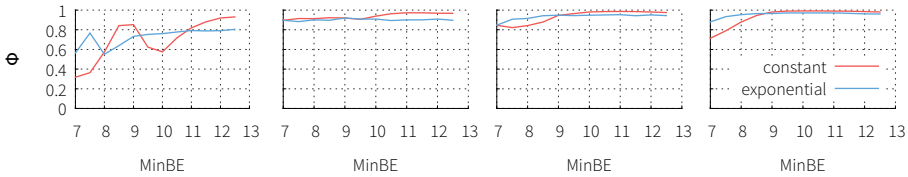
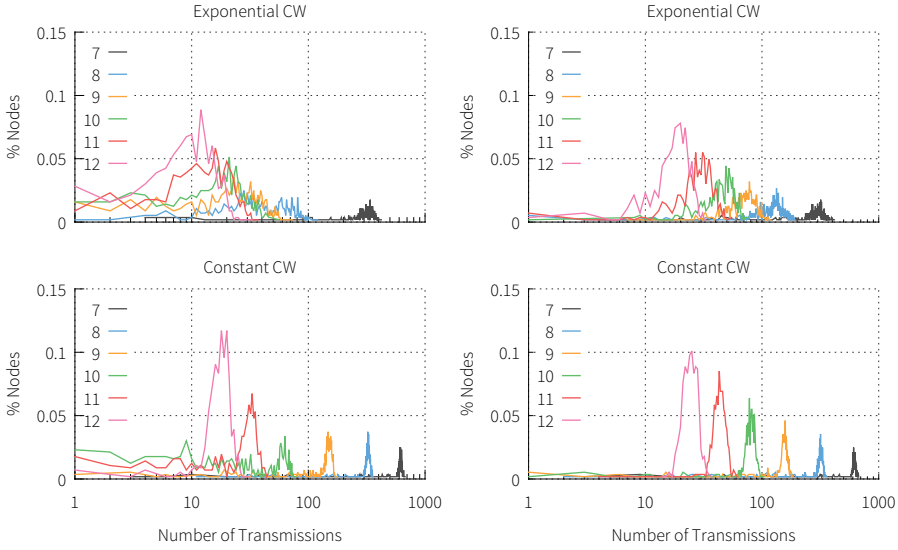


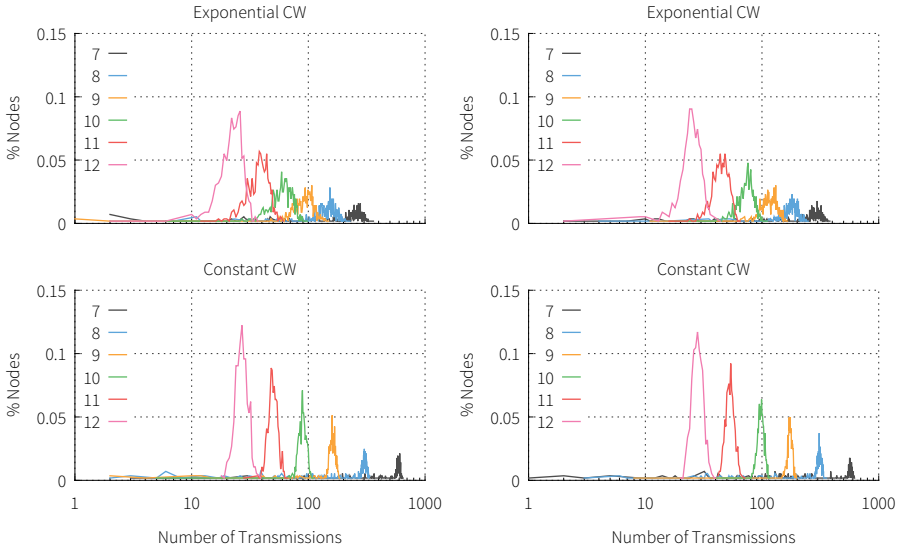
Figure 4.10: Fairness Index Φ for grid topologies.

Finally, the fairness index Φ in Figure 4.10 shows that for all the densities, those values of $minBE$ that yield high congestion, make the system less fair or even unpredictable, such as the case of $\Delta = 5m$ (Figure 4.10a). Indeed, excessive congestion, as results from $minBE = 7, 8, 9$, with constant back-off and inter-node distance $\Delta = 5m$, not only leads to very low goodput (as seen in Figure 4.8) but also unpredictable distribution of channel access and delivery success among nodes.

While comparing constant and exponential back-off approaches, we notice that although they provide the same maximal goodput and, in general, the same latency and fairness, under congestion they show some differences. For example, when the likelihood of backing off is high, e.g., $\Delta = 5m$, a constant back-off is more fair and also provides lower latencies. These results are in line with a vast body of literature [5; 28]. Figure 4.11 shows the distribution of transmitted packets among nodes for the two back-off configurations. The exponential case shows a wider curve, which indicates that nodes have a more differentiated number of transmissions compared to the constant back-off. This is even more pronounced in the denser configurations.



(a): $\Delta = 5m$ **(b):** $\Delta = 10m$



(c): $\Delta = 15m$ **(d):** $\Delta = 20m$

Figure 4.11: Distribution of transmitted packets in grid topologies.

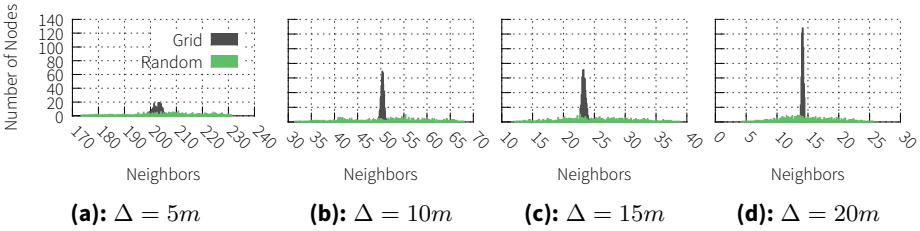


Figure 4.12: Degree distribution of nodes for grid and random topologies of various densities.

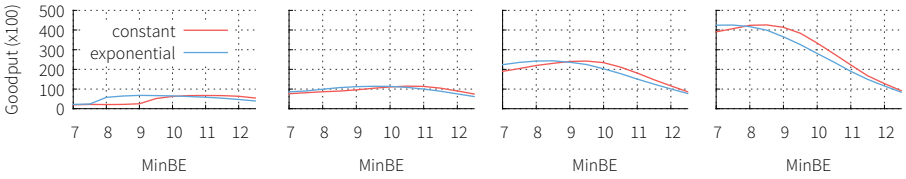
4.3.3 Random Node Distribution

While grid networks provide a good basis to understand the behavior of the MAC for a specific density, they are far from representing a real-world ad hoc network. In this section we evaluate the performance of the same set of CSMA parameters in scenarios where the area is the same for each density but nodes are randomly placed in it. The purpose is to study how the different connectivities between nodes within a network affects the performance of CSMA. Figure 4.12 shows the degree distribution[2] for the random and grid configurations of each density. On average the node degree is approximately the same for both configurations, but the distribution of degrees varies significantly. Similarly to the previous study, the surface of random networks is modelled as a torus to give uniformity to the connectivities between nodes.

A first look at Figure 4.13 and Figure 4.14 shows a strong resemblance of, respectively, network goodput and packet latency with the grid scenarios seen in Section 4.3.2. The slightly asymmetric bell shaped curves show that the maximal network goodput is reached around the same values of $minBE$ for all the densities. More importantly, despite the big difference among the degrees of the nodes, the random scenario reaches the same maximal network goodput as the grid scenario. However, a closer look at network goodput shows an important difference between the two.

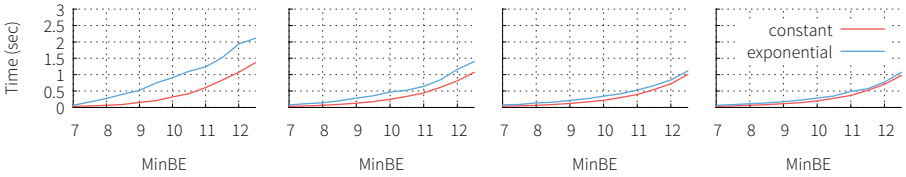
Let us take a look at Figure 4.13b and, in particular, at the constant backoff. Here, the maximal goodput, 11400, is reached for $minBE = 10.5$, which is 150% higher than the lowest goodput achieved for $minBE = 7$, i.e., 7590. In comparison, in the grid scenario the maximal goodput, i.e.,

[2]The degree of the nodes is measured according to the same technique employed in Section 3.4.1.



(a): $\Delta = 5m$ **(b):** $\Delta = 10m$ **(c):** $\Delta = 15m$ **(d):** $\Delta = 20m$

Figure 4.13: Network goodput for random topologies.



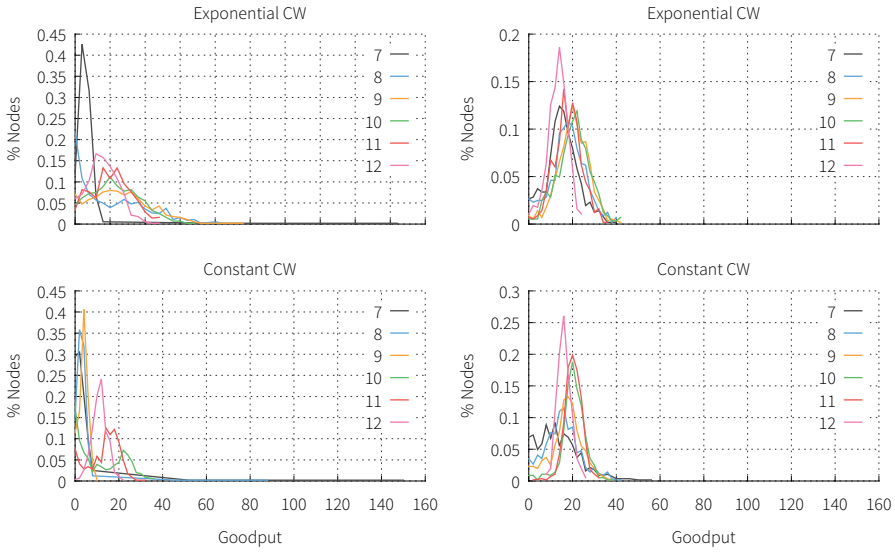
(a): $\Delta = 5$ **(b):** $\Delta = 10$ **(c):** $\Delta = 15$ **(d):** $\Delta = 20$

Figure 4.14: One-hop packet latency for random topologies.

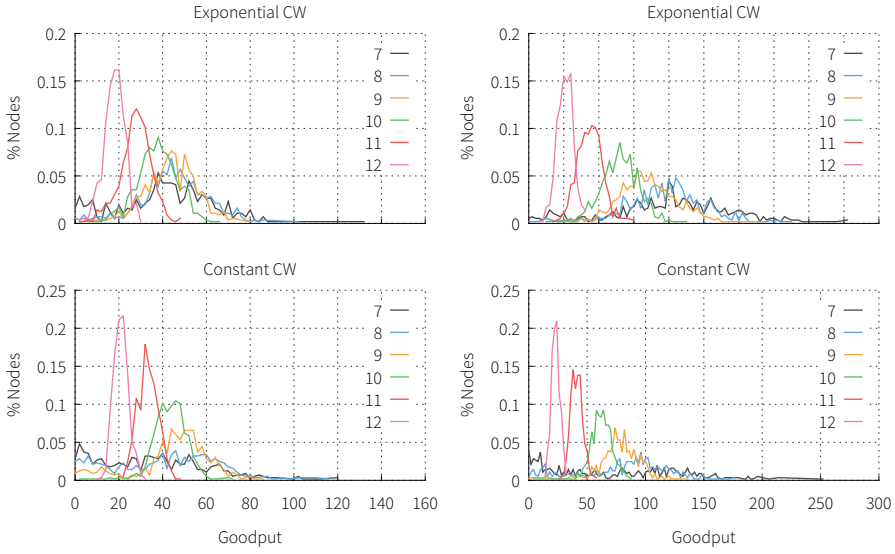
11281, achieved for $minBE = 10.5$ is 290% higher than the lowest goodput resulting from $minBE = 7$, i.e., 3816. The same observation is valid also for sparser networks. So, it would seem that the $minBE$ does not have the same impact on network goodput for the random network as it does on grid networks. In other words, a random network seems to perform much better than the grid in presence of congestion (i.e., low value of $minBE$).

The key to understanding this behavior lies in Figure 4.15, which shows the distribution of goodput among nodes. Looking at Figure 4.15b and in particular at the constant back-off approach, we observe that for low values of $minBE$ such as 7, 8 and 9, the goodput curve is very widely spread over a broad range of values. This suggests that nodes achieve a far-from-equal distribution of the goodput. However, as $minBE$ increases we see that the curve becomes more pronounced indicating that a large number of nodes reaches similar goodput. Figures 4.15a, 4.15c and 4.15d show a similar trend.

The link between congestion and unfairness does not come as a surprise. In fact, we have seen this behavior even in very regular topologies like the grid. Due to congestion nodes back-off more frequently, but, while doing so, some nodes can be more lucky than others in finding the idle channel to transmit while other have to back off. As the connectivity between nodes



(a): $\Delta = 5m$ **(b):** $\Delta = 10m$



(c): $\Delta = 15m$ **(d):** $\Delta = 20m$

Figure 4.15: Goodput distribution for random topologies.

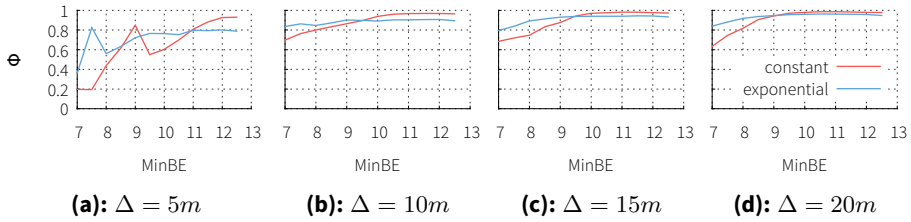


Figure 4.16: Fairness index Φ for random topologies.

varies, unfairness becomes more pronounced. Nodes located in very dense or very sparse areas may strive to deliver packets.

Such an observation is backed by the fairness index shown in Figure 4.16. For all the densities we see that low values of $minBE$ induce low fairness among nodes. In comparison to the grid topologies, for the same densities we notice lower values of the fairness index. However, for optimal values of $minBE$ the system appears to be fair to more than 95% of the nodes.

The study of CSMA parameters in uniformly distributed networks leads to the following observations:

1. The $minBE$ plays a crucial role in the performance of a CSMA protocol. So, depending on the network density and on the back-off approach, one has to choose the right value in order to reach optimal MAC performance.

2. While *network goodput*, as in the total number of delivered packets in a unit of time, is a reasonable metric for evaluating performance, it is not always accurate since it does not capture the distribution of the goodput among nodes. So, fairness has to be considered, in addition to goodput. Measuring goodput alone can be misleading.

3. While choosing between constant and exponential, we have seen that exponential back-off can deal better with congestion. However, for dense topologies, it raises the problem of fairness and latency.

4. Congestion leads to unfairness, even in very regular networks as the grid, and gets more pronounced as nodes' degrees vary.

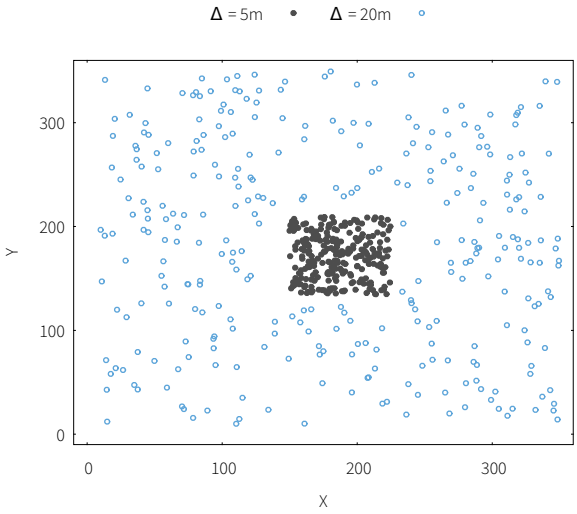


Figure 4.17: Non-uniform random topology.

4.3.4 Heterogeneous Node Distribution

After studying the behavior of CSMA in uniform networks, we consider now a more realistic scenario, where nodes gather around a point of interest. We crafted a topology with 564 nodes with the highest concentration occurring at the center of the area. To simplify matters we adopt two densities that we have already studied thoroughly in the previous sections. In particular, the nodes at the center have average inter-node distance $\Delta = 5m$, whereas those in the sparse area are randomly placed at $\Delta = 20m$ inter-node distance. Such a heterogeneous topology is depicted in Figure 4.17. Similarly to the previous studies regarding the MAC, we eliminate the border effect by modeling a torus surface.

The aim of this study is to investigate the benefits of differentiating the configuration of CSMA for different densities. To this end, we assess the MAC performance by adopting two CSMA configurations. In the first configuration we assign to all nodes the same CSMA parameters, irrespectively of their density. While in the second, we assign different parameters to the nodes depending on their density. So, in Figure 4.17 the nodes are color-coded to show the two groups.

In particular, we consider the following CSMA configurations:

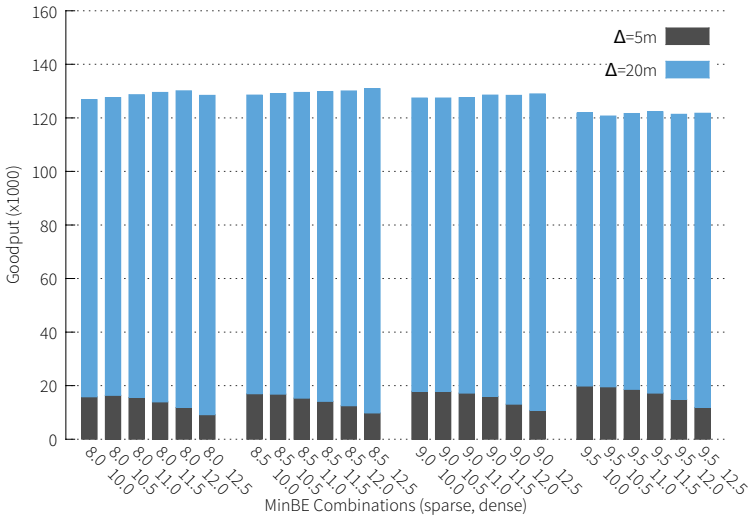


Figure 4.18: Network goodput for dual CSMA configuration.

- **Single configuration:** All nodes are configured with constant back-off and the values of $minBE$ range from 8 to 12.5.
- **Dual configuration:** Nodes in the sparse area have $minBE$ values 8, 8.5, 9 and 9.5, whereas the $minBE$ values of nodes in the dense area range from 10 to 12.5. Such values of $minBE$ are selected among the ones that have shown in our previous study (Section 4.3.3) to provide highest goodput for each respective density. Also here we employ constant back-off.

Figure 4.18 shows the network goodput of the dual configuration. Each bar represents the network goodput divided for each group of nodes. A first look at this graph suggests that network goodput does not vary much for various combinations of $minBE$. The major difference can be observed when $minBE = 9.5$ is assigned to the nodes in the sparse area. It is worth recalling that, according to the study in Section 4.3.3, the $minBE$ that provides the highest goodput for such a network density is $minBE = 8.5$.

It is interesting to see how various configurations of the two densities affect the performance of each other. So, when the nodes in the dense area transmit at the lowest rate ($minBE = 12.5$), we observe that the goodput of this group of nodes reaches the lowest value, while the nodes in the sparsest area reach their maximum goodput. And vice versa, when the nodes in the

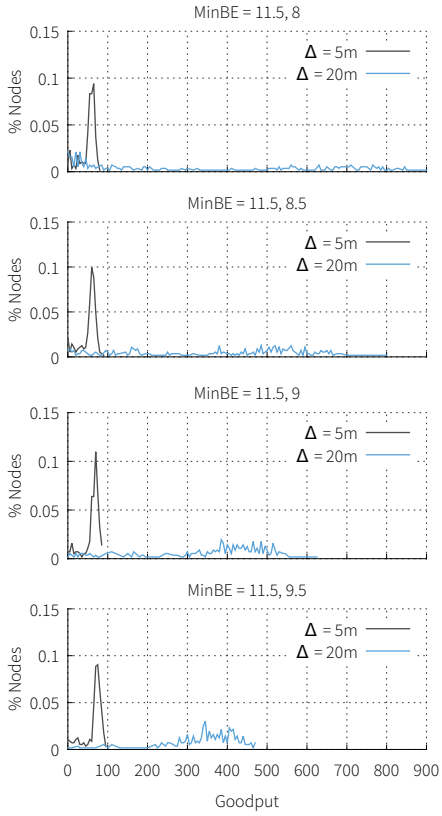


Figure 4.19: *Goodput distribution for dual CSMA configuration.* Varying the $minBE$ of the nodes in the *sparse* area.

sparsest area reach their minimum transmission rate ($minBE = 9.5$), hence, minimum goodput, the nodes in the dense area reach their highest goodput (the last block of bars).

Although goodput varies little for various combinations of $minBE$, from the perspective of fairness–goodput distribution–things look significantly different. In particular, Figure 4.19 shows how the distribution of goodput is affected by the MAC configuration of the nodes located in the sparse area. So, we fix the $minBE$ of the nodes in the center to 11.5, while we vary the $minBE$ of the nodes in the sparse, from 8 to 9.5. When $minBE = 8$, we see that nodes in the dense area have a relatively fair goodput distribution around the value of goodput 50. Yet, some of the nodes in the dense area

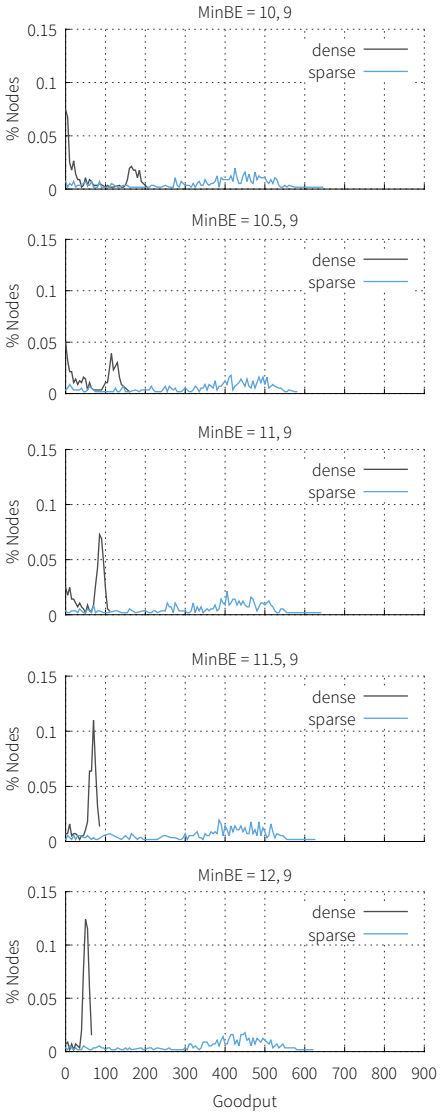


Figure 4.20: *Goodput distribution for dual CSMA configuration.* Varying the $minBE$ of the nodes in the *dense* area.

have goodput close to 0. On the other hand, we observe a highly unfair distribution of goodput for nodes in the sparse area, ranging from 100 to 900. As nodes in the sparse area slow down the rate of transmission–the $minBE$

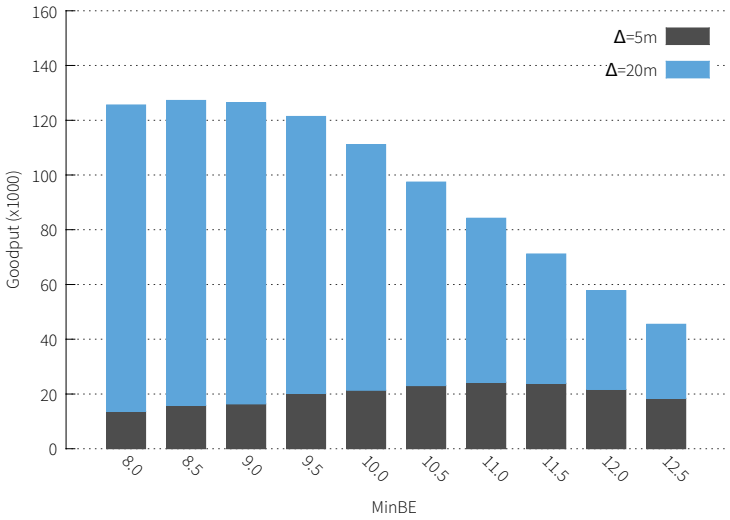
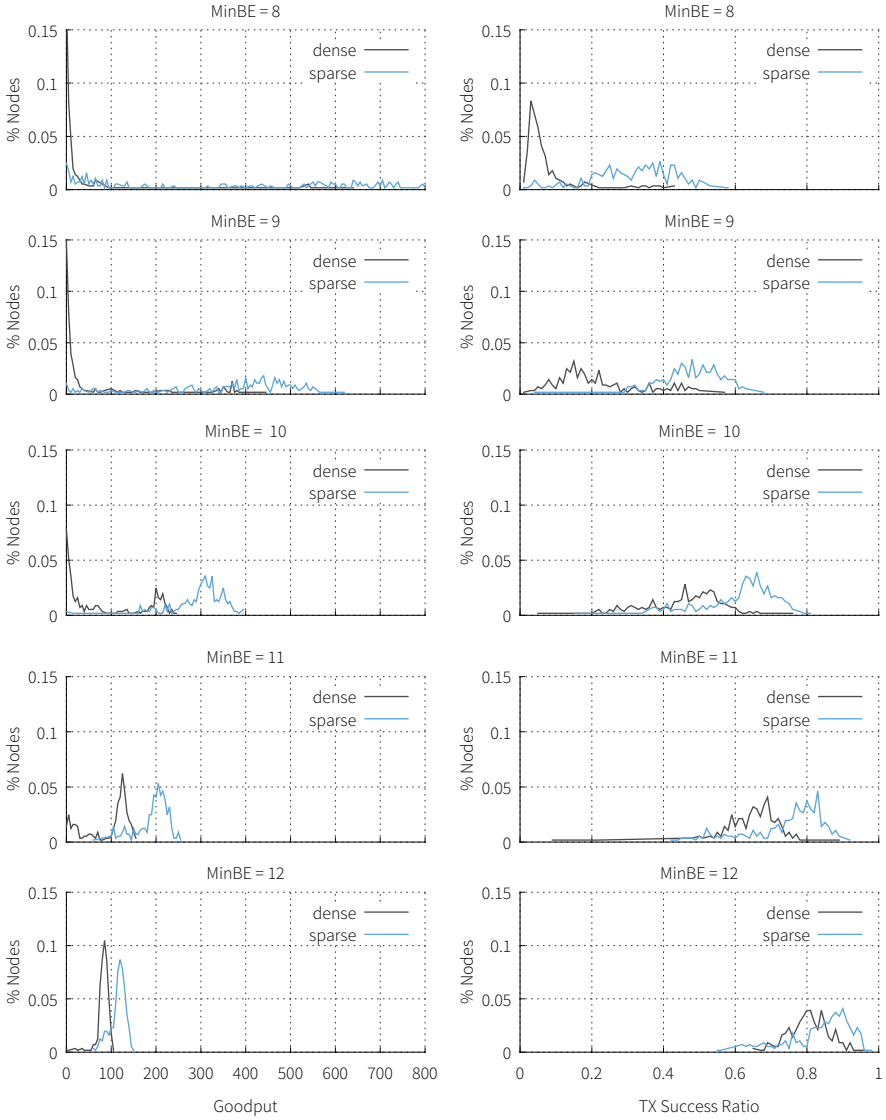


Figure 4.21: Network goodput for single CSMA configuration.

of sparse nodes increases–we observe that the flat curve takes a bell shape around goodput 400 and fewer nodes with goodput close to 0. Another positive effect of $minBE$ increase for sparse nodes is that the curve representing the goodput of nodes in the dense area skews slightly towards the right. It means that nodes in the dense area also increase their goodput. Finally, we see that overall fewer nodes have very low goodput.

Vice versa, we now look at goodput distribution as we vary the $minBE$ of the nodes in the dense area. So, similarly, we fix the $minBE$ of the nodes in the sparse area to 9, while we vary the $minBE$ of nodes in the dense area. Figure 4.20 shows that for $minBE = 10$ some nodes in the dense area have almost no goodput, while others have relatively high goodput. As $minBE$ increases, the two curves merge into one single spike–the goodput of dense nodes concentrates around 50. This shows that low transmission rates provide a much fairer environment, but at the cost of lower goodput.

Now, let us look at a *single CSMA configuration* for all nodes independently of their density. Figure 4.21 shows that for values of $minBE$ ranging between 8 and 9.5, a single MAC configuration can provide comparable goodput to the dual configuration (see Figure 4.18). In fact, for CSMA configurations with low $minBE$ we see that goodput reaches highest values and they are rather comparable to each other. But this view is incomplete. Figure 4.22 provides a closer look at the distribution of goodput and transmission success



(a): Goodput distribution **(b):** TX Success ratio

Figure 4.22: Distribution of goodput and transmission success ratio for single CSMA configuration.

ratio. Here, we see that low values of $minBE$ provide an extremely unfair performance among nodes. In particular, when $minBE = 8$ a large number of nodes have little or no goodput at all (top of Figure 4.22a) and their transmission success ratio is close to zero (Figure 4.22b). Yet, some nodes in the sparse area reach pretty high goodput. As $minBE$ increases, we see two bell-shaped distributions for both goodput and transmission success ratio, suggesting a better performance and fairer allocation of the channel.

Through these experiments we showed that properly assigning different MAC configurations to nodes in non-uniform networks (here, $minBE = 11.5$ in the dense area and $minBE = 9$ in the sparse area) provides high goodput while keeping fair performance among nodes. Moreover, we saw that only looking at the goodput may not be enough to judge the MAC layer performance.

4.4 Towards Self-Configuration of MAC Parameters

In the previous sections we have seen that an appropriate tuning of the CSMA parameters would provide a great benefit in terms of both goodput and fairness. In ad hoc networks, usually, the density is not known a priori and nodes may be mobile. So, to operate optimally, nodes would have to adapt their CSMA configuration accordingly.

When nodes contending for channel access are within each other's communication range, they can fairly share the channel bandwidth by increasing/decreasing their transmission rate based on information collected by the neighborhood [28]. But matters get complicated in multi-hop networks where contenders can be several hops away. Indeed, nodes located way beyond the transmission range of a sender, in what is known as interference range, can disturb local communications. And since the interference range can be much larger than the transmission range [100], the number of possible interferer can be significant. This property of wireless networks makes it challenging to estimate the effective number of contenders. In addition, changes to the MAC configuration of individual nodes may affect others way beyond their reach. In the rest of this section we look at existing approaches for self-configuration of MAC parameters in multi-hop networks and expose the challenges for applying them to our system.

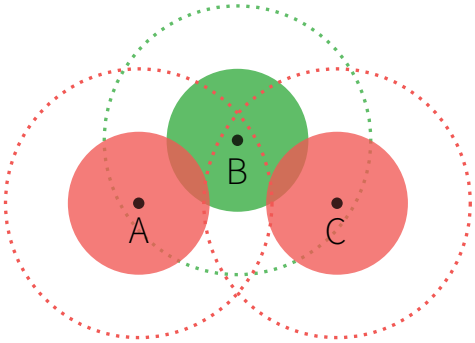


Figure 4.23: *Illustration of the problem of interference.* The interference range is depicted with dashed lines whereas the transmission range with filled circle.

A common approach to deal with high traffic in wireless networks is adopting a congestion avoidance mechanism similar to the one used by TCP in wired networks [98; 97; 28; 36]. TCP relies on an *additive increase multiplicative decrease* approach to optimally share the bandwidth between various contenders. Essentially, the sender keeps a congestion window to limit the number of packets in transit. The congestion window is initialized to one unit and, after every acknowledgement, it increments exponentially up to a low watermark, after which it continues to grow linearly. When a packet loss occurs (i.e., timeout) the congestion window is set back to one unit and the same process is repeated. While this is a viable and efficient solution in wired networks, adopting it in a wireless environment is not as easy given the inherent unreliability of the latter.

In fact, in wireless networks, packet loss cannot be uniquely attributed to collisions. Weak signal between nodes, typically caused by signal attenuation, is just one example of how packets may get lost. Although some solutions have been proposed for detecting packet collisions in wireless LANs [94; 85], it remains a challenging problem for low-power radio antennas. Moreover, acknowledgements cannot be used in broadcast protocols.

Another common technique to detect congestion consists of periodically sampling the noise level of the channel. When high contention occurs, it tends to increase the noise level. For example, in Z-MAC [77] the noise level is estimated by measuring the average number of back-offs that a sender takes before transmitting a packet. This measurement comes without extra overhead, as part of the carrier sensing mechanism. Basically, before transmitting a packet, a sender performs a clear channel assessment (CCA). When

the noise level during a CCA is higher than the CCA threshold, the node backs off. As the average number of back-offs reaches a certain threshold (it was found empirically that 0.3 was a suitable threshold), a node determines that the network is congested.

However, locally measured channel noise cannot be used directly by the nodes for tuning their MAC parameters. Unlike wired networks, where congestion is detected by all connections passing a router, in multi-hop wireless networks congestion is not uniformly perceived by nodes. As a result, decreasing the transmission rate as a response to locally detected congestion would make matters even worse. Consider the scenario depicted in Figure 4.23. The solid-colored areas represent the transmission range, whereas the areas delimited with dashed lines represent the interference range. Node B is in the interference range of nodes A and C that are transmitting at a higher pace compared to B. If B decides to decrease its own pace as a result of the perceived contention, it will not help to mitigate congestion. On the contrary, nodes A and C will experience lower contention, and have higher chances to find the channel idle. This would result in even less fairness.

Such an observation implies that to decrease contention, local changes to CSMA parameters would not help. Rather, nodes perceiving high contention levels would have to notify other nodes in the neighborhood and hope they will adjust their transmission rate accordingly. But since contention may be caused also by nodes outside the transmission range (i.e., interference range), using notifications would not solve the problem.

One approach could be to send congestion notifications to all nodes within a certain number of hops. As observed by Rao and Stoica [74], hop count is not a reliable way to identify a contending relationship between nodes. In fact, the interference range is impossible to foresee and it may change frequently.

4.5 Concluding Discussion

In this chapter we studied the behavior of a simple CSMA MAC protocol in relatively large ad hoc networks of different densities. Our main goal was to find the parameters that maximize the MAC performance of ad hoc networks for in a data-intensive scenario.

In the first part of this chapter we showed that the MAC configuration has a great impact on the performance of gossip-based dissemination protocols (e.g., Gossip3). Furthermore, we showed that MAC and network layer are

tightly interdependent on each other: a poorly performing MAC leads to a high number of collisions which, in turn, induces the gossip protocol to forward messages more aggressively. This, ultimately, results in higher traffic at the MAC layer. Such a vicious cycle caused by congestion can collapse the entire message dissemination process.

In the second part, we performed a broad study of the CSMA parameters for a number of network densities and node distributions. To evaluate the behavior of a highly loaded MAC in isolation from the upper layers, we performed experimental simulations where nodes handled continuously synthetic traffic from the network layer. Moreover, we considered two metrics for the evaluation of each configuration, namely, goodput and fairness.

The study showed that, in order to reach maximal goodput, it is important to tune the configuration of CSMA according to the network density. An inappropriate configuration of CSMA can lead to either high levels of contention or to underutilized channel. In both cases the resultant goodput is suboptimal.

Moreover, we observed that high contention can also cause poor fairness in channel allocation among various nodes. This proved to be true even for very regular network topologies such as grids. In general, a constant back-off can provide more fairness compared to an exponential back-off mechanism. The benefits of a proper CSMA configuration turned out to be even more visible in non-uniform networks. Differentiating the CSMA configuration of nodes located in areas with different densities provided a much better performance compared to a single CSMA configuration.

While it would be desirable to have a self-configuring mechanism of CSMA parameters at the node level, we argue that it can be a challenging task to pursue in our system. This is mainly due to the specific properties of multi-hop wireless networks. In particular, changing the MAC parameters of a transmitting node, may affect a great part of the network, beyond its transmission range, and, as such, difficult to control at a large scale.

CHAPTER 5

PUTTING THE PIECES TOGETHER

In the previous chapters we focused on two important protocol layers, namely network and MAC, aiming at optimizing them in isolation from each other. For the network layer we optimized a well-known broadcast protocol while assuming an optimally tuned MAC layer for the specific network. Whereas, for the MAC layer, we optimized the network goodput and fairness while assuming constant incoming traffic by the upper, network layer.

Still, our ultimate goal is to maximize the message insertion rate from all nodes, while achieving good coverage. Moreover, we want this to be reached for any network configuration, as in density and size. We believe that this can be achieved by optimizing both layers simultaneously: the MAC layer tuned to give best goodput; the network layer making sure that the messages reach many nodes while keeping the traffic to a minimum. In this chapter we explore the interdependency of the MAC and the network layer for message dissemination in ad hoc networks.

5.1 Background

A large body of research has exploited the interdependency between various layers of the protocol stack, giving thus rise to the so-called cross-layer design of communication protocols. Such a paradigm design allows layers to exchange state information in order to improve the efficiency of certain aspects of the network. A common technique consists of adapting the routing of messages accordingly (network layer) in order to (i) prevent congestion in certain areas of the network, and (ii) increase the lifetime of single nodes and, hence, of the network by efficiently distributing the load on the nodes [54; 88; 12; 18]. Bononi and Di Felice [11] propose a solution for creating and maintaining a backbone in VANETs by relying on the cross-layer interactions among MAC and the clustering scheme. In [10] the authors propose to adapt the contention window of the MAC protocol to the type of message under transmission, allowing, thus, emergency messages to win contention with higher probability. A more radical cross-layer approach is proposed by Akyilidiz et al. [4] which consists of merging the functionalities of the network, MAC, and link layer into one. In this way, nodes in a WSN may take the initiative to participate in communication based on a number of conditions, including buffer overflow, link quality and congestion indication.

Other research works have looked into the interplay between MAC and routing protocols [7; 80; 79; 64]. For example, Barrett et al. [7] have investigated the impact of the interaction between routing and MAC layer protocols in multi-hop MANET topologies. At the network layer the authors consider two well-known point-to-point routing protocols, namely, AODV [68] and DSR [13], and for the MAC layer the standard protocol IEEE 802.11 [59] and MACA [41]. The authors show that the paths selected by the routing protocol directly affect the contention of the nodes at the MAC level. While at the same time, contention at the MAC layer makes routing inefficient, as messages struggle to reach their destination. Consequently, a higher number of route queries may take place. Pazurkiewicz et al. [64] propose NarrowCast, a link-layer primitive that improves the energy efficiency of gossiping in sensor networks. In particular, NarrowCast allows to broadcast to just a fraction of the neighbors, those that have not seen a broadcast before.

Our study, like the previously mentioned ones, also looks into the interdependency of the MAC and network layer, but from the perspective of a gossip-based broadcast protocol, namely Gossip3, and a simple CSMA MAC protocol in ad hoc networks.

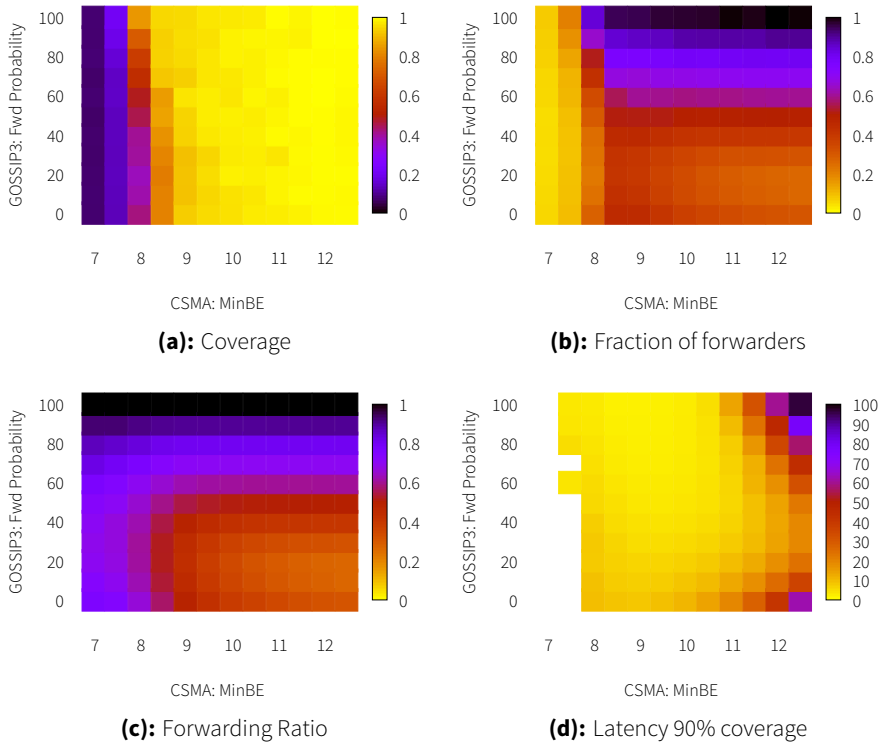
	Parameter	Value
Uniform Topology	Network Size (N)	529
	Nodes Distribution	random
	Inter-node Distance (Δ)	$15m$
APP Layer	Number of Messages	100
NET Layer Config 1	Protocol	Gossip3
	Fwd. Probability (p)	0 .. 1
	Compensation factor (m)	1
	Flooding Hops (k)	0
NET Layer Config 2	Protocol	Adaptive Gossip3
MAC Layer	Protocol	CSMA
	Backoff	constant
	MinBE	7 .. 12.5

Table 5.1: Experimental settings divided by protocol stacks

5.2 Cross-layer Performance with Artificial Traffic

In the previous chapters we introduced artificial traffic in order to evaluate various aspects of congested networks without the problem of buffer overflow. This technique consists of all nodes continuously generating and transmitting "dummy" packets to their immediate neighbors, while only a few of them additionally broadcast useful packets. The benefit of this approach is that simulations execute significantly faster, allowing thus, a broad evaluation of parameters. In this section we consider the same setting to evaluate the interplay between the various configurations of (i) Gossip3 at the network layer and (ii) CSMA at the MAC layer.

We start by considering a random uniformly distributed topology. Table 5.1 shows a summary of the parameters we adopt for this set of experiments grouped into functional layers. All nodes generate synthetic traffic, while only three nodes placed in different locations act as sources by inserting 100 messages at a rate of 1 message/second each. At the network layer we consider two configurations: default Gossip3 and adaptive Gossip3 as introduced in Chapter 3. At the MAC layer, we adopt a simple CSMA with



(a): Coverage

(b): Fraction of forwarders

(c): Forwarding Ratio

(d): Latency 90% coverage

Figure 5.1: *Impact of CSMA and default Gossip3 parameters on message dissemination with synthetic traffic.* Network size is 529 and nodes are distributed randomly with average inter-node distance $\Delta = 15m$.

constant back-off where we only vary the $minBE$. We chose a topology with $\Delta = 15m$ as a representative example. Results for $\Delta = 5m, 10m, 20m$ (not plotted here) show the same trend although with different absolute values.

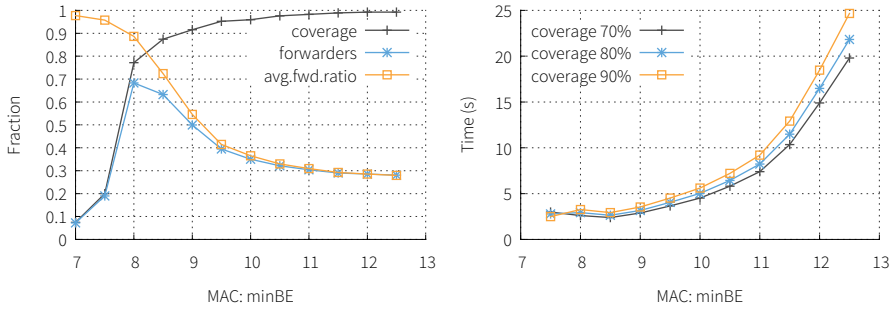
Gossip3 evaluation. Figure 5.1 shows some aspects of the performance of message dissemination in color-coded graphs, where yellow indicates good performance, as opposed to black that indicates poor performance. As expected, Figure 5.1a shows that best coverage is achieved for large values of $minBE$, that is, when contention is minimized. In this case, the forwarding probability p has almost no impact on coverage ($minBE \geq 10$). However, large values of p (e.g., $p \geq 0.7$) can make up for a poorly performing MAC (e.g., $minBE = 8.5$) at the cost of much higher traffic, as in fraction of forwarders and forwarding ratio (see Figure 5.1b and 5.1c). A good perform-

ing MAC results in fewer rebroadcasts. The lowest traffic can be achieved for high values of $minBE$, namely, $minBE \geq 10$, and for low values of p, namely $p = 0.1$, $p = 0.2$ and $p = 0.3$. But these configuration parameters are not optimal in terms of latency of message propagation, as shown in Figure 5.1d. It is important to note that in Figure 5.1b a low number of forwarders (yellow cells) in correspondence to low values of $minBE$ is not to be considered as an indication of good performance. Rather, the fraction of forwarders is low because, due to high contention, the coverage is low too.

Although the impact of the network and MAC configurations is not fully reflected due to the artificial traffic, Figure 5.1d shows that as $minBE$ increases so does the latency. But interestingly, $minBE$ is not the parameter that affects latency the most. Rather, low and high values of p increase the latency significantly . There are different reasons for this. A low p implies low initial redundancy and, as such, nodes have to compensate more in order to make up for non-redundant packets. Considering that compensation occurs after a certain time interval after a new packet reception, per-hop latency affects the overall latency at the network level. In contrast, with a high p, nodes forward a good part of the received packets, which implies that compensation is often unnecessary. In this case, the high latency is due to the long buffering time at transmission as the gossiping protocol decided for a large number of packets to be aired. Clearly, as shown in Figure 5.1d, large values of $minBE$ can magnify even more this effect.

To sum up, the choice of $minBE$ and p is very important for an efficient performance of message dissemination. A high $minBE$ can provide high coverage and generate minimal traffic, but it can also lead to high message dissemination latency. While the forwarding probability p seems not to have an impact on the coverage, it should be chosen properly in order to avoid high latency and unnecessary rebroadcasts. The configurations with $minBE = 10, 10.5$ and $p = 0.3$ seem to provide a good trade-off for the considered network topology.

Adaptive Gossip3 evaluation. Figure 5.2 shows the impact of $minBE$ to Adaptive Gossip3. It contains fewer data points since the forwarding probability is determined by the number of estimated neighbors. The resultant forwarding ratio is plotted in Figure 5.2a in an orange, squared line. A poorly performing MAC, namely $minBE < 9$, causes the gossiping protocol to respond with high forwarding probability and yet the achieved coverage is very low. But as $minBE$ increases, the forwarding ratio is reduced whereas coverage reaches its maximum. There is an inverse relation between the $minBE$ and the forwarding probability. As expected, the traffic generated by the net-



(a): Dissemination Performance **(b):** Dissemination latency

Figure 5.2: *Impact of CSMA parameter, $minBE$, on message dissemination through adaptive Gossip3.* Network size is 529 and nodes are distributed randomly with average inter-node distance $\Delta = 15m$.

work layer goes down as $minBE$ increases. However, considering the huge impact that $minBE$ has on latency (Figure 5.2b), large values of $minBE$ are to be avoided. As seen in the previous experiments, $minBE = 10$ or $minBE = 10.5$ seem to provide the best configuration for Adaptive Gossip3 when $\Delta = 15m$.

5.3 Cross-layer Performance with Realistic Traffic

The results of the last section gave us a glimpse on how the network and the MAC layer can affect each other under congested conditions. The artificial traffic used in these experiments made it possible to evaluate a broad number of parameters across the two layers in a reasonable time. While the considered traffic scenarios represent a good projection of a congested network, they do not capture all the factors affected by congestion.

In the following experiments we adopt a realistic scenario where the traffic stems from packets injected by all the nodes at regular intervals. As all nodes will be sending messages to all other nodes by means of gossiping, messages will become redundant as they are passed along hop by hop. This implies that messages will be buffered at the MAC layer possibly for long intervals, which will have an impact on dissemination latency and more.

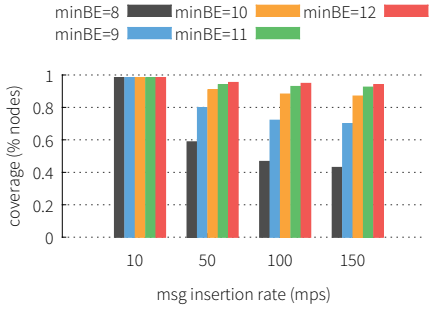
	Parameter	Value
Topology	Network Size (N)	529
	Nodes Distribution	random
	Inter-node Distance (Δ)	$15m$
APP Layer	Source nodes	529
	Messages sent by each source	50
	Message Insertion Rate (network)	10, 50, 100, 150 msg/s
	Period of inserting new message	52.9, 10.6, 5.3, 3.5 s
NET Layer	Protocol	Adaptive Gossip3
MAC Layer	Protocol	CSMA
	Backoff	constant
	MinBE	8 .. 12
	Buffer Size	200

Table 5.2: Experimental settings divided by protocol stacks.

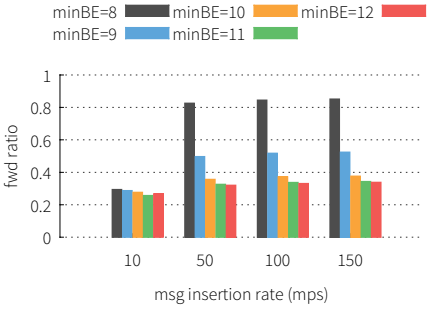
We use the same network topology as in the previous study, i.e., 529 nodes, arranged randomly with average inter-node distance $\Delta = 15m$. In addition, we introduce a couple of parameters that come into play with a realistic traffic pattern. First, the message insertion rate is the number of messages inserted by all nodes every second (mps). It has a direct influence on the offered load. We vary this parameter from 10 to 150 mps, which implies that each node will send a new message at a period ranging from 52.9s to 3.5s. Second, the MAC buffer size determines the maximal number of messages that a node can buffer before transmission. We set the buffer size to 200, as it has shown to perform reasonably well for the topology configuration we consider. Table 5.2 summarizes the details of the following experiments.

Figure 5.3a shows that at the rate of 10 mps maximal coverage is reached, independently of the MAC configuration. In contrast, for offered loads that are greater than 10 mps, the MAC configuration plays an important role: low values of $minBE$ (i.e., 8 and 9) show a significant drop in coverage, whereas for $minBE >= 10$ coverage is still high and somewhat comparable between these MAC configurations. The sudden drop in performance at 50 mps for all MAC configurations suggests that at this message insertion rate network saturation is reached.

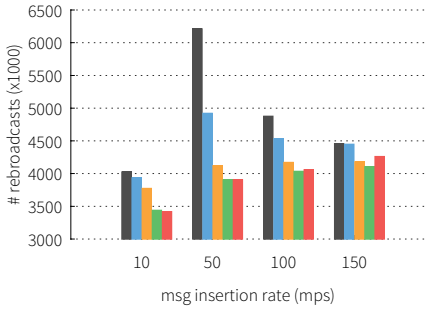
For offered loads as low as 10 mps congestion is relatively low, as can be observed also in the forwarding ratio of the nodes, Figure 5.3b. The forward-



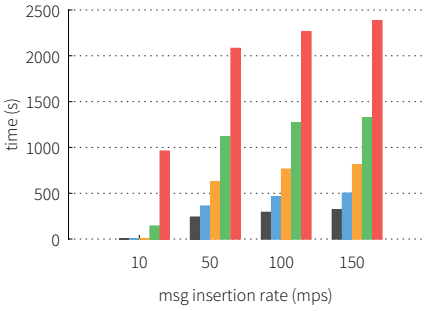
(a): Coverage



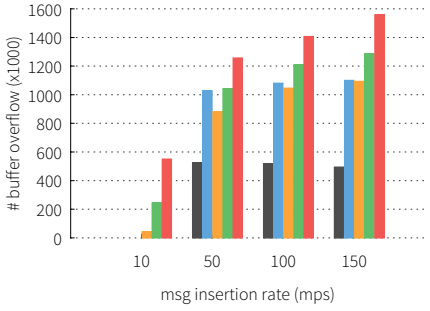
(b): Fwd probability



(c): Num. Rebroadcasts



(d): Dissemination Latency



(e): Buffer overflow

Figure 5.3: *Performance of adaptive Gossip3.* Evaluation of various configurations of the CSMA with various message generation rates. Nodes are randomly distributed with average inter-node distance $\Delta = 15m$.

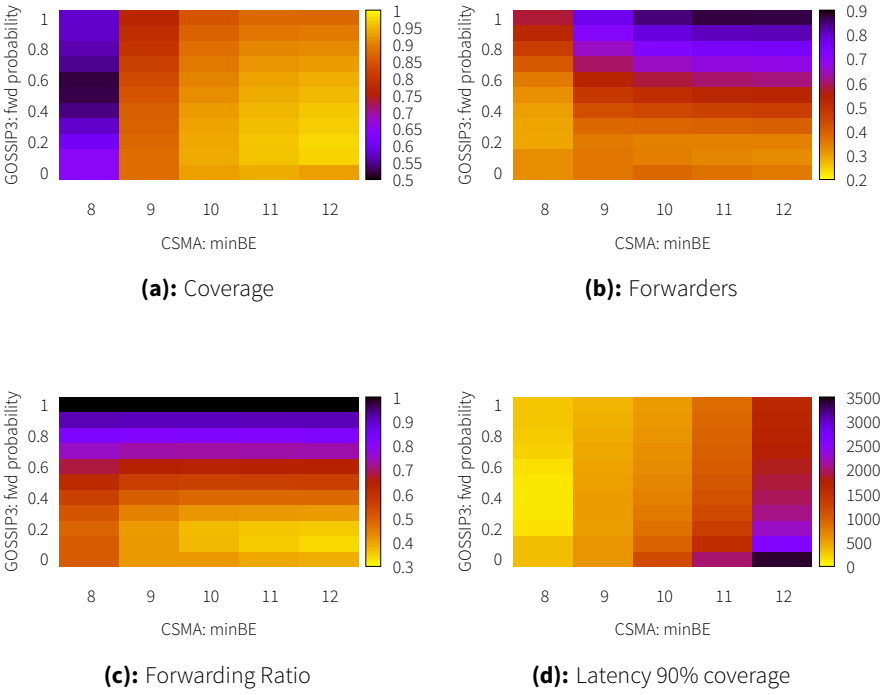
ing ratio is an indirect indicator of congestion: few rebroadcasts are needed when the quality of transmission is good. But for offered loads greater than 10 mps, low values of $minBE$ see a jump in the forwarding ratio. This regulator behavior of the self-configuring Gossip3 has an immediate effect in the number of messages nodes have to rebroadcast. Figure 5.3c shows that high congestion (e.g., $minBE = 8$ and 50 mps) leads to a much higher number of rebroadcasts. In comparison, a low congestion configuration (e.g., $minBE = 10$ and 50 mps) has 30% fewer packets to forward. This is a good indication of the vicious cycle during congestion: poor MAC performance causes high packet loss, which leads to more rebroadcasts at the gossiping layer, which ultimately results in more traffic.

While low values of $minBE$ are highly susceptible to the offered load, high values of $minBE$ keep the link quality high independently of the rate of message insertion. As a result, high values of $minBE$ help the gossiping layer to keep rebroadcasts to a minimum (see Figures 5.3b and 5.3c). The major drawback of slow MAC configurations is high latency, as depicted in Figure 5.3d. This is even more pronounced for high offered loads.

Figure 5.3e shows that for message insertion rates higher than 10 mps a considerable number of packets overflow the MAC buffer, independently of the $minBE$. However, it is surprising to see that the difference in buffer overflow between $minBE = 12$ and $minBE = 9$ is only 20%, considering that $minBE = 12$ transmits up to 8 times slower than $minBE = 9$ [1]. One probable explanation is related to the high frequency of backoffs in the presence of congestion (e.g., $minBE = 9$). This affects the buffering time of the outgoing packets and, as a result, the extent of buffer overflow. In contrast, in a non-congested network (e.g., $minBE = 12$) the medium is mostly idle, and the likelihood of backoff is much smaller. The large difference in transmission rate is smoothed out by the large extent of backoff of MAC configuration with $minBE = 9$.

As expected from the study with artificial traffic, the deployed MAC protocol has a great impact in saturated networks. In fact, in accordance with the conclusions made in the previous section, for the considered network topology, $minBE = 10$ seems to be a good trade-off. In fact, such a MAC configuration provides optimal support to the gossiping layer in terms of coverage, generated traffic, and dissemination latency (Figure 5.3). Moreover, we observed that buffer overflow occurs upon network saturation in all MAC configurations. However, a low-pace MAC can cope better with buffer overflow

[1] The contention window (CW) is invariably 2^{minBE}



(a): Coverage

(b): Forwarders

(c): Forwarding Ratio

(d): Latency 90% coverage

Figure 5.4: *Impact of CSMA and Gossip3 parameters on message dissemination.* Message generation rate is fixed to 50 mps. Network size is 529 and nodes are distributed randomly with inter-node distance $\Delta = 15m$.

compared to high-pace configurations because the high link quality ensures that neighboring nodes can make up for overflown packets.

Evaluation of Gossip3

Finally, we consider the original Gossip3 as introduced by Haas et al [27] and we evaluate various configurations of the network and MAC layer in a highly congested network with a realistic traffic pattern. Basically, the experimental setting is the same as in Section 5.2, with the only difference that traffic is now generated by broadcast packets inserted by all nodes at regular intervals. In particular, the message insertion rate is set to 50 mps. This rate has

shown in the previous section to provide a saturated network, yet reasonable dissemination performance for certain MAC configurations.

Figure 5.4a shows that the forwarding probability p has a great impact on the coverage. With the exception of the special case $p = 0$, we see that coverage drops as p increases. A quick look at Figures 5.4b and 5.4c shows that there is an inverse relation between coverage and the forwarding ratio: fewer rebroadcasts lead to better coverage. This indicates that unnecessary message redundancy can have a negative impact on the coverage, mainly because of buffer overflow. So, minimizing the forwarding ratio can indeed benefit dissemination performance for all the considered MAC configurations. It is important to note that the low fraction of forwarders for low values of $minBE$ (Figure 5.4b) are the result of very low coverage.

However, keeping the forwarding ratio at low levels has a negative impact on latency, as depicted in Figure 5.4d. A configuration with slightly poorer coverage (e.g., $minBE = 10$ and $p = 0.2$) can provide more than twice faster message propagation than a configuration with optimal coverage (e.g., $minBE = 12$ and $p = 0.2$). The choice of the parameters has to take into account these trade-offs between dissemination performance and latency.

5.4 Conclusions

In this chapter we investigated the interplay between the MAC and network layers under congestion. The results were very insightful. First, we observed that optimizations to individual layers are not effective when done in isolation from others. So, considering traditional Gossip3, an optimal forwarding probability p at the network level does not guarantee a good coverage if the underlying MAC layer is not properly configured. Vice versa, an optimal $minBE$ at the MAC layer does not guarantee optimal coverage unless the forwarding probability at the network layer is not properly configured.

Considering the fact that different network densities require different parameters at the MAC and network layers, the problem of assigning the most optimal parameters becomes significantly more challenging. However, we showed that our self-configured Gossip3 adapts p according to the perceived number of neighbors. Under congestion, nodes adjust their forwarding probability to ensure higher redundancy. However, for very high congestion levels, a higher forwarding ratio is not enough.

More often than not, trade-offs are necessary when it comes to choosing configuration parameters at the two layers. For instance, we have seen that

certain MAC configurations can reach slightly fewer nodes than others, but at much higher speed. We have also shown that keeping the rebroadcasts low can be beneficial for the coverage because MAC buffers are kept 'lighter', hence fewer packets overflow. But fewer rebroadcasts increase the dissemination latency. The choice of the parameters has to be driven by the application requirements and circumstances. Accordingly, one can choose to favor coverage over latency and vice versa.

CHAPTER 6

CONCLUSIONS

Managing people in crowded events has shown to be a challenging task because of the limited visibility and hindered mobility of the participants. Moreover, the large concentration of people often restrains any form of traditional communication, such as, cellular communication and WIFI hot spots.

In this dissertation, we looked only slightly ahead into the future and explored a decentralized solution for communicating in packed areas. In particular, we envisaged a massive adoption of coin-sized networked devices that, provided with low-power antennas and computing power, can form arbitrarily large ad hoc networks for relaying messages across the crowd. Interfaced with a smartphone, such a device could provide a power-efficient and unobtrusive way to keep track of various events in the crowd.

We considered a group communication application where group members keep in touch with each other by sharing location-based information through the ad hoc network. First, we focused on the security and privacy aspect of sharing private information (e.g., location) using an untrusted medium. Second, we tackled the problem of network congestion occurring as new messages are sent periodically and addressed the problem of effective message propagation. Finally, we showed that cross-layer optimizations can immensely benefit communication performance in congested ad hoc networks. In the following, we summarize the key results of this dissertation.

1. *Privacy-aware group communication.* We introduced a group monitoring use case in ad hoc networks and discussed a number of requirements from a functional perspective. Further, we studied possible privacy and security attacks that aim at disrupting communications between targeted nodes or groups of nodes, or simply at gathering information about nodes from such communications. To address these attacks, we proposed a protocol that prevents tracking messages throughout group information sharing. The proposed protocol uses flooding-based techniques to propagate anonymous messages among group members.

2. *Network-level optimizations.* Gossiping provides a compelling paradigm for ad hoc communication in the crowd. One of the main advantages of gossiping–robustness– can turn into an obstacle if message redundancy is not kept under control. In Chapter 3, we selected a well-known broadcast protocol, namely Gossip3, and performed a broad evaluation of its parameters in relatively large networks. We observed that the best-performing parameters differ significantly for different network densities. We proposed an adaptive version of Gossip3 where nodes adjust the rebroadcast probability of messages to their perceived local density. We showed that adaptive Gossip3 can reach optimal message dissemination while keeping message redundancy to a minimum for any network density.

3. *MAC-level optimizations.* In Chapter 4 we addressed the problem of maximizing the number of delivered packets (goodput) across the network for simple CSMA MAC protocols operating under congestion. We evaluated the impact of the medium access rate on the MAC performance of various topology settings. We showed to what extent an appropriate MAC configuration, tailored to the density of the network, can contribute in reaching maximal goodput. Moreover, we pointed out that high contention not only increases packet collisions, but it can also cause poor fairness in the distribution of goodput among nodes. Finally, through literature study we argue that adapting the CSMA parameters at the node level according to their local density would be particularly challenging, mainly due to hidden terminals and long interference range of wireless terminals. This is considered a possible direction for future work.

4. *Cross-layer optimizations.* In Chapter 5 we put together various configurations of the network and MAC layer and investigated the interplay

between the MAC layer and gossiping (network layer) under congestion. Through this study we observed a number of performance trade-offs and demonstrated the advantages of cross-layer optimizations. In particular, optimizations to the network layer seem to have little impact when the MAC layer configuration is not properly tuned and, vice versa. We showed that by setting an optimal cross-layer configuration, it is ultimately possible to push more messages per time unit while reaching good message dissemination.

6.1 Future Directions

In this dissertation we just scratched the surface of ad hoc communication in the crowd. As wearable devices are becoming widely available, we expect such communication paradigms to gain ground in crowded highly populated environments. The following directions could help to further improve the proposed work towards real-world deployments.

1. In order to deal with continuous loads of messages, the MAC layer is required to operate efficiently. We have seen the importance of MAC configuration in accordance with the number of interfering nodes, i.e., density. While MAC has been thoroughly investigated in the past, to the best of our knowledge, no approach has been proposed to deal with adaptive MAC configuration of large-scale, multi-hop, mobile networks operating under data-intensive conditions.

2. The protocol we proposed for private group communication (Chapter 2) uses 'blind' gossiping for message dissemination–the identity of the sender and the intended recipients is omitted in order to ensure privacy. While anonymity is a common technique for making sure that messages are untraceable and unlinkable to the communication parties, it also comes at a cost. Namely, messages are propagated across the entire network even if the recipient may be just a couple of hops away from the sender. In this context, an interesting research direction would be to investigate the trade-off between privacy and network traffic. In some circumstances, for instance, it may be wise to opt for more efficient communication at the expense of privacy and vice versa. Or even at group level, some groups may be more concerned about privacy than others.

3. Finally, a real-world deployment may consist of thousands of nodes [1] scattered throughout a vast area, such as the city center. Our proposed gossip-based message dissemination protocol focuses on non-uniform networks, while it does not take into account disconnected networks. Long-distance ad hoc communications may need additional investigation, especially in case of intermittent connectivity between nodes.

[1] Reportedly, the last King's Day in Amsterdam was attended by ca. 800 000 people.

SUMMARY

Crowds are often inconvenient for people participating in them. Consider for instance a group of friends participating in a city festival. Even if the friends stay close to each other, the limited visibility and the hindered mobility may make it difficult for them to stay together as a group. To make matters worse, communication through usual everyday devices such as cellphones is almost impossible. In vast crowded areas, the load on cellular networks is typically several times higher than usual. As a result, cellular communication faces serious limitations and text messages are delivered with huge delays–if at all.

Yet, a large number of people equipped with wireless devices, occupying a vast area, can potentially enable a distributed communication paradigm: ad hoc networking. Independent from any central infrastructure, ad hoc networks offer a compelling way to transport messages, hop by hop, from and to nodes participating in the network.

In this dissertation we introduce a protocol for group monitoring that relies solely on ad hoc networks and takes into account the privacy of the participants. We start with a thorough analysis of the requirements for exchanging messages in a group monitoring application. This analysis shows that in order to ensure untraceable communications, traditional end-to-end routing must be ruled out. Based on the analysis of the requirements, we propose a group monitoring protocol where nodes broadcast anonymous messages at regular, predetermined time intervals in a gossip-based fashion.

While gossip-based message propagation is instrumental for anonymous communications and copes well with nodes' mobility, it introduces a lot of redundancy. Since we expect ad hoc networks to be dense, high traffic is likely to collapse the entire message dissemination process. To prevent this, we look at a cross-layer optimization, by attempting to minimize message redundancy at the network layer and maximize channel utilization at the MAC layer.

At the network layer, we consider a well-known gossiping protocol called Gossip3 that is based on probabilistic rebroadcasting. We perform extensive experimental analysis of this protocol, simulating high network utilization in diverse network densities. The study shows that the protocol's level of redundancy is highly sensitive to the density of the network. In order to achieve minimal redundancy in the light of arbitrary network densities, we propose a novel algorithm that alleviates this shortcoming of Gossip3. Our algorithm tunes a node's rebroadcast probability based on the perceived density of the nodes. This way, the dissemination protocol can retain its optimal performance irrespectively of the density of the network it operates in.

At the MAC layer, we choose the CSMA protocol with binary exponential back-off (BEB), because it is simple and robust and deals well with dynamic networks. This protocol has two main parameters, namely the size of the contention window and the rate at which the contention window increases at each back-off. We perform a broad study of the performance of various CSMA-BEB parameter values for a number of network densities and node distributions. To evaluate the MAC performance in isolation from the network layer we emulate synthetic traffic. We observe the impact of the CSMA-BEB parameters with respect to goodput, fairness and latency. While the contention window increase rate has a certain impact on fairness, the contention window size is crucial for all three metrics we used. Finally, we identify the optimal parameters for a number of network densities.

To demonstrate the effectiveness of a cross-layer approach we consider both MAC and network layer in a set of experiments with realistic traffic– all nodes send new messages at regular time intervals. We show that a good cross-layer configuration allows nodes to send more messages per time unit while maintaining a high average delivery ratio (i.e., coverage) per message. Moreover, we demonstrate the interplay between the MAC and the network layer under congestion: a poorly performing MAC leads to a high number of collisions, which in turn induces the gossip protocol to forward messages more aggressively. This, ultimately, results in higher traffic at the MAC layer. Finally, an important point that emerges from these experiments is that when it comes to choosing configuration parameters at the two layers, trade-offs are often necessary. For example, a slightly lower dissemination coverage can provide a much faster message dissemination and vice versa.

In conclusion, this dissertation presents the first comprehensive study towards private ad hoc communication in the crowd. It gives an overview of principal challenges from the application level down to the MAC level. In particular, the broad study of the parameters of the MAC and network

layer provides an understanding of the influence they have on each other and of the trade-offs with respect to their parameters. We finally show that cross-layer optimizations can significantly improve the overall performance of group communications.

SAMENVATTING

Verkennen van Cross-Layer Afhankelijkheden in Drukke Draadloze Ad Hoc Netwerken

Mensenmenigten zijn dikwijls lastig voor de mensen die zich er in bevinden. Neem bijvoorbeeld een groep vrienden die deelneemt aan een stadsfestival. Zelfs als de vrienden dicht bij elkaar in de buurt blijven kan het beperkte zicht en de gehinderde bewegingsvrijheid het hen moeilijk maken om als een groep bij elkaar te blijven. Dit wordt verergerd doordat communicatie met behulp van alom aanwezige apparaten zoals mobiele telefoons nagenoeg onmogelijk is. In grote overvolle gebieden is de druk op cellulaire netwerken doorgaans vele malen hoger dan gebruikelijk, met als resultaat dat het cellulaire verkeer ernstige hinder ondervindt en tekstberichten met grote vertraging afgeleverd worden–als ze sowieso aankomen.

Toch kan een groot aantal mensen met mobiele telefoons verspreid over een uitgebreid gebied een gedistribueerde vorm van communicatie mogelijk maken, namelijk ad hoc netwerken. Doordat ad hoc netwerken onafhankelijk van een gecentraliseerde infrastructuur zijn, bieden ze een aantrekkelijke manier om berichten te transporteren, van, stapsgewijs via, en naar de knopen in het netwerk.

In deze dissertatie introduceren we een protocol voor het monitoren van groepen dat enkel van ad hoc netwerken afhankelijk is en de privacy van de deelnemers in aanmerking neemt. We beginnen met een grondige analyse van de vereisten voor het uitwisselen van berichten door een applicatie voor het monitoren van groepen. Uit de analyse blijkt dat om ontraceerbare communicatie te kunnen bieden, de traditionele aanpak van het gericht routeren van berichten naar hun bestemming uitgesloten moet worden. Op basis van de eisenanalyse stellen we een groepsmonitoringsprotocol voor waarin de

knopen anonieme berichten verspreiden in regelmatige, vantevoren vastgestelde tijdsintervallen op een roddel-gebaseerde manier.

Hoewel roddel-gebaseerde verspreiding van berichten essentieel is voor anonieme communicatie en goed omgaat met mobiliteit van de knopen, introduceert het een grote hoeveelheid redundantie. Aangezien we verwachten dat ad hoc netwerken een hoge densiteit hebben, is er een grote kans dat een grote hoeveelheid verkeer leidt tot het falen van het gehele proces van berichtverspreiding. Teneinde dit te voorkomen kijken we naar cross-layer optimalisatie, door te proberen in de netwerklaag de redundantie van berichten te minimaliseren en in de MAC-laag het gebruik van communicatiekanalen te maximaliseren.

In de netwerklaag bekijken we een welbekend roddelprotocol dat Gossip3 heet en gebaseerd is op probabilistische herverspreiding. We verrichten een uitgebreide experimentele analyse van dit protocol, waarbij we een hoge benuttingsgraad simuleren bij verschillende netwerkdichtheden. De studie laat zien dat het redundantieniveau van het protocol sterk afhankelijk is van de dichtheidsgraad van het netwerk. Om minimale redundantie te bereiken bij arbitraire netwerkdichtheden stellen we een nieuw algoritme voor dat deze tekortkoming van Gossip3 verlicht. Ons algoritme stemt de waarschijnlijkheid van herverspreiding af op de waargenomen dichtheid van de knopen. Op deze manier kan het verspreidingsprotocol zijn optimale prestaties behouden ongeacht de dichtheid van het netwerk waarin het opereert.

In de MAC-laag kiezen we voor het CSMA-protocol met binair exponentiële terugtrekking (BEB), omdat het simpel en robuust is en goede ondersteuning voor dynamische netwerken biedt. Het protocol heeft twee hoofdparameters, namelijk de groote van de twistvenster en de snelheid waarmee de twistvenster toeneemt na elke terugtrekking. We verrichten een brede studie van de prestaties van verschillende CSMA-BEB parameterwaarden voor een aantal netwerkdichtheden en verspreidingen van knopen. We emuleren synthetisch netwerkverkeer om de MAC-prestaties onafhankelijk van de netwerklaag te kunnen evalueren. We registreren de invloed van de CSMA-BEB parameters op daadwerkelijke aflevering van berichten, gelijke verdeling, en snelheid van aflevering. Hoewel de snelheid van toename van de twistvenster een bepaalde invloed op de gelijkheid van verdeling heeft, heeft de grootte van de twistvenster een invloed op alle drie de gebruikte maatstaven. Tenslotte stellen we de optimale parameters vast voor een aantal netwerkdichtheden.

Teneinde de effectiviteit van de cross-layer aanpak te tonen, nemen we zowel de MAC-laag als de netwerklaag in beschouwing in een reeks experimenten met realistisch verkeer, waarbij alle knopen nieuwe berichten ver-

sturen met regelmatige tijdsintervallen. We laten zien dat een goede cross-layer-configuratie de knopen in staat stelt meer berichten per tijdseenheid te versturen terwijl een hoog gemiddeld gehalte van aflevering (dekking) per bericht behouden blijft. Bovendien tonen we de wisselwerking tussen de MAC-laag en de netwerklaag bij congestie: een slecht presterende MAC leidt tot een hoog aantal botsingen wat vervolgens het roddelprotocol berichten agressiever doet laten doorsturen. Dit leidt uiteindelijk tot meer verkeer op het MAC-niveau. Tenslotte is een belangrijk punt dat blijkt uit deze experimenten dat het bij het kiezen van configuratieparameters voor de twee lagen vaak nodig is om keuzes te maken tussen gewenste doelstellingen. Zo kan een minder complete verspreidingsdekking zorgen voor een veel hogere afleveringssnelheid en andersom.

Concluderend, dit proefschrift presenteert de eerste uitgebreide studie naar privacy-beschermende ad hoc communicatie in menigten. Het biedt een overzicht van de belangrijkste uitdagingen van het applicatieniveau naar beneden tot aan het MAC-niveau. In het bijzonder biedt de brede studie van de parameters van de MAC- en netwerklagen een inzicht in de invloed die deze lagen op elkaar hebben en in de mogelijke keuzes die gemaakt moeten worden wat betreft hun parameters. Tenslotte tonen we dat cross-layer optimalisaties de algehele prestaties van groepscommunicatie aanzienlijk kunnen verbeteren.

REFERENCES

[1] OMNeT++ discrete event simulator. `http://www.omnetpp.org/`. [Online; accessed 07-October-2013].

[2] TinyOS. `http://webs.cs.berkeley.edu/tos/`. [Online; accessed 20-June-2014].

[3] IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2003*, 2003.

[4] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. A cross-layer protocol for wireless sensor networks. In *40th Annual Conference on Information Sciences and Systems*, pages 1102–1107. IEEE, 2006.

[5] H. Anouar and C. Bonnet. Optimal constant-window backoff scheme for ieee 802.11 dcf in single-hop wireless networks under finite load conditions. *Wireless Personal Communications*, 43(4):1583–1602, 2007.

[6] P. Ball. Every move you make: Patterns of crowd movement. In *Why Society is a Complex Matter*, pages 7–12. Springer Berlin Heidelberg, 2012.

[7] C. Barrett, A. Marathe, M. V. Marathe, and M. Drozda. Characterizing the interaction between routing and mac protocols in ad-hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 92–103. ACM, 2002.

[8] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the 4th annual international conference on Mobile computing and networking*, MOBICOM '98, pages 76–84. ACM, 1998.

[9] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE, 1997.

[10] Y. Bi, L. X. Cai, X. Shen, and H. Zhao. A cross layer broadcast protocol for multihop emergency message dissemination in inter-vehicle communication. In *IEEE International Conference on Communications*, ICC '10, pages 1–5. IEEE, 2010.

[11] L. Bononi and M. Di Felice. A cross layered MAC and clustering scheme for efficient broadcast in VANETs. In *4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, MASS '07, pages 1–8. IEEE, 2007.

[12] F. Bouabdallah, N. Bouabdallah, and R. Boutaba. Load-balanced routing scheme for energy-efficient wireless sensor networks. In *Global Telecommunications Conference*, GLOBECOM '08, pages 1–6. IEEE, 2008.

[13] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual international conference on Mobile computing and networking, MOBICOM '98*, pages 85–97. ACM, 1998.

[14] L. Buttyán and J. Hubaux. *Security and cooperation in wireless networks*, volume 188. Cambridge University Press, 2007.

[15] L. Cai and H. Chen. Touchlogger: inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX conference on Hot topics in security*, HotSec'11, pages 9–9, 2011. USENIX Association.

[16] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, pages 21–21, 2010.

[17] M. Cattani, S. Guna, and G. P. Picco. Group monitoring in mobile wireless sensor networks. In *In International Conference on Distributed Computing in Sensor Systems and Workshops*, DCOSS '11, pages 1–8. IEEE, 2011.

[18] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 12(4): 609–619, 2004.

[19] S. Chatterjea, L. Van Hoesel, and P. Havinga. AI-LMAC: An adaptive, information-centric and lightweight MAC protocol for wireless sensor networks. In *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 381–388. IEEE, 2004.

[20] M. Conti, I. Zachia-Zlatea, and B. Crispo. Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 249–259, 2011. ACM.

[21] S. Das, B. Manoj, and C. Ram Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 24–35. ACM, 2002.

[22] M. Dobson, S. Voulgaris, and M. van Steen. Merging ultra-low duty cycle networks. In *41st International Conference on Dependable Systems & Networks*, DSN '11, pages 538–549. IEEE, 2011.

[23] J. Douceur. The sybil attack. In *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin Heidelberg, 2002.

[24] A. El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *IEEE International Conference on Communications*, volume 5 of *ICC '02*, pages 3418–3423. IEEE, 2002.

[25] J. Garcia-Luna-Aceves and E. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1380–1394, 1999.

[26] D. Gavidia and M. Van Steen. A probabilistic replication and storage scheme for large wireless networks of small devices. In *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, MASS '08, pages 469–476. IEEE, 2008.

[27] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking (ToN)*, 14(3):479–491, 2006.

[28] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *SIGCOMM Comput. Commun. Rev.*, 35(4):121–132, 2005.

[29] Y.-C. Hu, D. B. Johnson, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad hoc networks*, 1 (1):175–192, 2003.

[30] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 134–147, 2004. ACM.

[31] H. Hung and D. Gatica-Perez. Estimating cohesion in small groups using audio-visual nonverbal behavior. *IEEE Transactions on Multimedia*, 12(6):563–575, 2010.

[32] R. Jain and D. C. W. Hawe. Quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, DEC Research Report TR-301, 1984.

[33] K. Jamieson, H. Balakrishnan, and Y. Tay. Sift: A MAC protocol for event-driven wireless sensor networks. In *Wireless Sensor Networks*, pages 260–275. Springer, 2006.

[34] J. Jetcheva and D. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 33–44. ACM, 2001.

[35] L. Ji and M. Corson. Explicit multicasting for mobile ad hoc networks. *Mobile Networks and Applications*, 8(5):535–549, 2003.

[36] Y. Jian and S. Chen. Can CSMA/CA networks be made fair? In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MobiCom '08, pages 235–246. ACM, 2008.

[37] W. Jie and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *International Journal of Foundations of Computer Science*, 14(02): 201–221, 2003.

[38] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[39] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2):293–315, 2003.

[40] C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 162–175. ACM, 2004.

[41] P. Karn. MACA-a new channel access method for packet radio. In *Amateur radio 9th computer networking conference*, *ARRL/CRRL*, volume 140, pages 134–140, 1990.

[42] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254. ACM, 2000.

[43] Y. Ko and N. Vaidya. Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[44] J. Kong and X. Hong. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 291–302, 2003. ACM.

[45] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. Haneveld, T. E. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, Simutools '08, pages 71:1–71:8. ICST, 2008.

[46] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43 (4):499–518, 2003.

[47] L. Lazos, R. Poovendran, and S. Čapkun. ROPE: robust position estimation in wireless sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, 2005. IEEE Press.

[48] S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. In *1st Annual Workshop on Mobile and Ad Hoc Networking and Computing*, MobiHoc '00, pages 37–44. IEEE, 2000.

[49] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24(3):353–363, 2001.

[50] Y. Lin and Y. Hsu. Multihop cellular: A new architecture for wireless communications. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3 of *INFOCOM '00*, pages 1273–1282. IEEE, 2000.

[51] W. Lou and J. Wu. Double-covered broadcast (DCB): A simple reliable broadcast algorithm in manets. In *The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3 of *INFOCOM '04*, pages 2084–2095. IEEE, 2004.

[52] S. Mank, R. Karnapke, and J. Nolte. MLMAC—an adaptive TDMA MAC protocol for mobile wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 8(1-2):57–78, 2009.

[53] G. Mao, B. Fidan, and B. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51(10):2529–2553, 2007.

[54] A. Mei and J. Stefa. Routing in outer space. In *The 27th Conference on Computer Communications*, INFOCOM '08, pages 2234–2242. IEEE, 2008.

[55] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen. A power-aware broadcasting algorithm. In *17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5. IEEE, 2006.

[56] A. Mohammed, M. Ould-Khaoua, and L. Mackenzie. An efficient counter-based broadcast scheme for mobile ad hoc networks. *Formal Methods and Stochastic Models for Performance Evaluation*, pages 275–283, 2007.

[57] N. S. Networks. Understanding Smartphone Behavior in the Network. `http://nsn.com/system/files/document/Smart_Lab_WhitePaper_27012011_low-res.pdf/`. [Online; accessed 07-October-2013].

[58] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 151–162, 1999. ACM.

[59] B. O'hara and A. Petrick. *IEEE 802.11 handbook: a designer's companion*. IEEE Standards Association, 2005.

[60] T. Ozaki, J. Kim, and T. Suda. Bandwidth-efficient multicast routing for multihop, ad-hoc wireless networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2 of *INFOCOM '01*, pages 1182–1191. IEEE, 2001.

[61] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd Workshop on Security of ad hoc and Sensor Networks, SASN '04*, volume 4, pages 88–93. ACM, 2004.

[62] S. Park and D. Park. Adaptive core multicast routing protocol. *Wireless Networks*, 10(1):53–60, 2004.

[63] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3 of *INFOCOM '97*, pages 1405–1413. IEEE, 1997.

[64] T. Pazurkiewicz, M. Gregorczyk, and K. Iwanicki. NarrowCast: A new link-layer primitive for gossip-based sensornet protocols. In *EWSN 2014: Proceedings of the 11th European Conference on Wireless Sensor Networks*, pages 1–16. Springer-Verlag LNCS 8354, Oxford, UK, February 2014.

[65] W. Peng and X.-C. Lu. AHBP: an efficient broadcast protocol for mobile ad hoc networks. *Journal of Computer Science & Technology*, 16(2): 97–102, 2001.

[66] W. Peng and X.-C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '00, pages 129–130, 2000. IEEE Press.

[67] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4):234–244, 1994.

[68] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, WMCSA '99, pages 90–100. IEEE, 1999.

[69] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity — a proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[70] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 95–107. ACM, 2004.

[71] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS '02*, pages 3866–3875. IEEE, 2002.

[72] M. Rahman, N. Nasser, A. Inomata, T. Okamoto, M. Mambo, and E. Okamoto. Anonymous authentication and secure communication protocol for wireless mobile ad hoc networks. *Security and Communication networks*, 1(2):179–189, 2008.

[73] N. Raiman, H. Hung, and G. Englebienne. Move, and I will tell you who you are: detecting deceptive roles in low-quality data. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 201–204. ACM, 2011.

[74] A. Rao and I. Stoica. An overlay MAC layer for 802.11 networks. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services, MobiSys '05*, pages 135–148. ACM, 2005.

[75] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[76] I. Rhee, A. Warrier, J. Min, and L. Xu. DRAND: distributed randomized TDMA scheduling for wireless ad hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06*, pages 190–201. ACM, 2006.

[77] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, 2008.

[78] D. Roggen, M. Wirz, D. Helbing, and G. Tröster. Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods. *Networks and Heterogeneous Media*, 6(3):521–544,

2011.

[79] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda, and S. Tanaka. A network-aware MAC and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '03*, pages 88–97. ACM, 2003.

[80] E. M. Royer, S.-J. Lee, and C. E. Perkins. The effects of MAC protocols on ad hoc network communication. In *Wireless Communications and Networking Confernce, WCNC '00*, volume 2, pages 543–548. IEEE, 2000.

[81] E. Royer and C. Perkins. Multicast operation of the ad hoc on-demand distance vector routing protocol. In *Proceedings of the 5th annual international conference on Mobile computing and networking, MOBICOM '99*, pages 207–218. ACM, 1999.

[82] A. Sarwate and A. Dimakis. The impact of mobility on gossip algorithms. *Information Theory, IEEE Transactions on*, 58(3):1731–1742, March 2012.

[83] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Wireless Communications and Networking, WCNC '03*, volume 2, pages 1124–1130. IEEE, 2003.

[84] D. Scott. Dynamic probabilistic retransmission in ad hoc networks. In *In Proceedings of the International Conference on Wireless Networks, ICWN '04*, 2004.

[85] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Moving away from collision avoidance: Towards collision detection in wireless networks. In *ACM HOTNETS*, pages 1–6, 2009.

[86] S. Seys and B. Preneel. ARM: Anonymous routing protocol for mobile ad hoc networks. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications, AINA '06*, pages 133–137, 2006. IEEE Computer Society.

[87] C. Shen and C. Jaikaeo. Ad hoc multicast routing algorithm with swarm intelligence. *Mobile Networks and Applications*, 10(1):47–59, 2005.

[88] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual international conference on Mobile computing and networking, MOBICOM '98*, pages 181–190. ACM, 1998.

[89] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast core-extraction distributed ad hoc routing. In *Wireless Communications and Networking Conference, WCNC '99*, pages 1313–1317. IEEE, 1999.

[90] R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, 1999.

[91] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, 80(8):1125–1131, 1997.

[92] Y. Tseng, S. Ni, and E. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, 52(5):545–557, 2003.

[93] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *1st International Workshop on Networked Sensing Systems, INSS '04*, pages 205–208, 2004. Society of Instrument and Control Engineers (SICE).

[94] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *ACM SIGCOMM Computer Communication Review*, 39(4):3–14, 2009.

[95] K. Wessel, M. Swigulski, A. Köpke, and D. Willkomm. MiXiM: the physical layer an architecture overview. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools '09*, pages 78:1–78:8. ICST, 2009.

[96] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '02*, pages 194–205. ACM, 2002.

[97] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking, MOBICOM*

'06, pages 146–157. ACM, 2006.

[98] A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking, MOBICOM '01*, pages 221–235. ACM, 2001.

[99] C. Wu and Y. Tay. AMRIS: A multicast protocol for ad hoc wireless networks. In *Military Communications Conference Proceedings, MILCOM '99*, volume 1, pages 25–29. IEEE, 1999.

[100] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In *Global Telecommunications Conference, GLOBECOM '02*, volume 1, pages 72–76. IEEE, 2002.

[101] M. B. Yassein, S. F. Nimer, and A. Y. Al-Dubai. A new dynamic counter-based broadcasting scheme for mobile ad hoc networks. *Simulation Modelling Practice and Theory*, 19(1):553 – 563, 2011.

[102] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '02*, volume 3, pages 1567–1576. IEEE, 2002.

[103] Y. Yi, M. Gerla, and T. Kwon. Efficient flooding in ad hoc networks: a comparative performance study. In *IEEE International Conference on Communications, ICC '03*, volume 2, pages 1059–1063. IEEE, 2003.

[104] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Citeseer, 2001.

[105] Q. Zhang and D. Agrawal. Dynamic probabilistic broadcasting in manets. *Journal of Parallel and Distributed Computing*, 65(2):220–233, 2005.

[106] Y. Zhang, W. Liu, and L. Wenjing. Anonymous communications in mobile ad hoc networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '05*, volume 3, pages 1940–1951, 2005.