PhD-FSTC-2013-18
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 04/07/2013 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Patricia RUIZ

Born on 5 April 1978 in Algeciras, (Spain)

EFFICIENT COMMUNICATION PROTOCOLS FOR AD HOC NETWORKS

Dissertation defense committee

Dr. Pascal Bouvry, dissertation supervisor
Professor, University of Luxembourg

Dr. Frédéric Guinand
Professor, University of Le Havre, France

Dr. Steffen Rothkugel, Chairman
Assistant Professor, University of Luxembourg

Dr. Grégoire Danoy
Scientific Collaborator, University of Luxembourg

Dr. Franciszek Seredynski, Vice Chairman
Professor, Cardinal Stefan Wyszynski University, Warsaw

Abstract

Mobile ad hoc networks, also called MANETs, are self-organised networks spontaneously created between neighbouring devices that do not need any infrastructure. The possibility of deploying a communication network at any moment and at any place opens a wide range of potential scenarios where the use of an ad hoc network is extremely important and useful. For example, we can imagine an area with no infrastructure, due to a natural disaster (flooding, earthquake, etc.), a war damage or simply a remote location. In such situations, the deployment of an ad hoc network for rescuing or evacuating people is key.

Efficient communication protocols are very important for providing good services, but in ad hoc networks is very challenging. It is necessary to deal with packet loss, collisions, mobility, network partitions, fading, energy constraints, obstacles, etc. Due to the intrinsic broadcast nature of wireless medium, dissemination algorithms are one of the most appropriate protocols for communicating devices. Additionally, broadcasting is one of the main low level operation as many applications and even other protocols rely on its service.

Therefore, in this thesis, we are proposing two different broadcasting algorithms for efficiently disseminate messages in the network. The first approach relays on an underlying tree topology, while the second is an energy aware algorithm that tries to reduce the energy consumption. In these unpredictable and changeable networks, protocols need to adapt to the current circumstances and this adaptability is usually obtained using thresholds. The values of these thresholds directly influence the behaviour of the algorithm. Finding them is a complex and critical task as protocols are usually very sensitive. We tackle this problem using approximation algorithms or metaheuristics, more specifically, we use a parallel iterative local search and two evolutionary algorithms. In order to ease this process, we have created a modular experimental framework for evaluating, validating and optimising any algorithm in MANETs.

To Berna and María

Acknowledgements

Looking backwards, honestly I must say, I am very happy for finally finishing my PhD, but at the same time sad for closing a marvellous period of time in my life. During these years, there have been many people I need to thank, I hope I will not forget any.

My deepest gratitude to my advisor, Pascal Bouvry. First, for encouraging me to start the PhD when I was not really convinced. And also, for all his time and patience during all our long discussions. Even when they were not planned, he always found time for me, and trusted on me sometimes more than myself. His wise advise not only related to work but also personal were very helpful.

I would also like to thank the members of my CET committee, Steffen Rothkugel and Grégoire Danoy, for kindly guiding me with their expertise during all these years. Specially to Grégoire, with his unlimited patience and politeness, always trying to give his opinion in the most well-mannered way, even though he knows I am a bit obstinate.

I can not forget to mention the warm embrace I got form Albert Zomaya, Bjorn Landfeldt and Young Choon Lee during my stay at university of Sydney.

Thanks to my colleagues at university of Luxembourg, specially the ones at *team Bouvry* for making coming to work not a duty but a pleasure. Not only the current people in the team, but also to some of them that already left. The interesting discussions that many times come out of the blue during lunch or at the office, and the very nice advise I got from them, really enriched me. There, I found more than colleagues, I have many friends that make easier life far from the family.

Also, thanks to all my friends to whom I bored to death trying to explain my work and that even without understanding a word of what I was saying, they kept asking about the progress of my thesis.

I am particularly indebted to my parents Paqui and Pepe, for their unconditional support not only during the PhD, but believing in me during all my life. Also to my sister Eva, who is always there for whatever I need. And in general, to all my family from Algeciras, Málaga, Granada or Madrid who always cared about me and gave me the strength to continue.

Last but not least, this thesis would not have been possible without the immeasurable support of Berna. He not only helped me with his valuable advise and expertise, but also with his patience and love. Specially during the last months, that were specially hard. I can only be grateful. And to my little María, the sunshine of my life. For making me laugh when I did not find the time to do it.

Contents

List of Figures

List of Tables

List of Algorithms

List of definitions

Chapter 1

Introduction

Contents

During the last decade, advances in technology have made possible what not so long time ago was just a dream: to be able to communicate with a remote person at any time, and at any place. It was materialised thanks to a chain of successes, not only to a single brilliant idea. For example, the wireless technology had a major role in the process, as it makes possible the connection between remote devices wirelessly. However, it was also very important the reduction of the size of the hardware, so that people can carry a small device in their pocket or pursuit, as well as, the cost of creating and assembling all the components. We can not underestimate the power of the crowd. If the production of the devices were expensive, only a few privileged people would have been able to afford this technology. Resulting in a spectacular failure. The success of this phenomenon is to be able to communicate with anyone, at any moment and at any place, but also at low cost.

Additionally, the telecommunication companies envisioned a great business and kept improving the devices and updating the network resources, so that now, it is not only possible to communicate between remote devices using the wireless technology, but also, devices are able to connect to the internet at high speeds.

Nowadays, 70% of the world's population have a mobile phone, that is over 5 billion mobile subscribers, and 9 out 10 people in countries like the US. Additionally, 27% of mobile phones are smartphones provided with internet access. By 2014, it is expected that mobile internet connection will take over desktop internet usage (according to [1]). All those statistics envision a near future where every person will always carry a mobile device with communication capabilities.

This boom in the technology and the magnificent acceptance in the society have led researchers, during the last years, to focus in a new kind of wireless networks: *the ad hoc networks*.

Definition 1 (Ad hoc network).
An ad hoc network is a self-configuring network composed of a set of devices with wireless capabilities that can communicate between them without any infrastructure [142].

These ad hoc networks are mostly created on the fly when devices with communication capabilities meet. There is no need of any carrier company or infrastructure to start the communication. The intrinsic characteristics of such networks, as the self-organisation, the heterogeneity of devices, the dynamism as well as the unpredictable behaviour have some inner drawbacks that makes them difficult to implement, e.g. the changing topology, the dependence on battery life, the limited transmission range, the partitioning of the network, etc.

As already mentioned, the possibility of enabling the communication between two devices no matter if there is any available infrastructure around is a very challenging field that has attracted the attention of many researchers. They are trying to overcome all the problems inherit to ad hoc networks so that they become a reality. Chapter 2 introduces further ad hoc networks, their characteristics, drawbacks, challenges, etc.

1.1 Importance of ad hoc networks

The possibility of deploying a communication network at any moment and at any place opens wide range of potential scenarios where the use of an ad hoc network is extremely important and useful. We can imagine an area with no infrastructure, due to a natural disaster (flooding, earthquake, etc.), a war damage or simply a remote location. In such situations, the deployment of an ad hoc network for rescuing or evacuating people is key. Allowing the communication between groups located in distant areas may be very helpful in critical situations in order to coordinate and reorganise them.

However, ad hoc networks are not restricted to safety of disaster situations, they can also be used for gathering sensed data and send it to a remote server, exchanging safety road messages between vehicles or simply for infotainment, just to mention a few.

Moreover, envisioning the saturation the telephone network might suffer due to the predicted extremely high use, ad hoc networks can be used by the carrier companies to alleviate their traffic when dealing with highly dense areas.

The flexibility and heterogeneity of this kind of networks make its deployment a certain success. Therefore, efficient communication protocols are of extreme importance in order to provide good services. As we will see in detail in Chapter 2, efficient communication in ad hoc networks is very challenging. It is necessary to deal with packet loss, collisions, mobility, network partitions, fading, energy constraints, obstacles, etc.

1.2 Motivations

Due to the intrinsic broadcast nature of wireless medium, dissemination algorithms are one of the most appropriate protocols for communicating devices. Additionally, broadcasting is one of the main low level operation as many applications and even other protocols rely on its service. In wireless networks, broadcasting algorithms are usually associated to the broadcast storm problem [249].

When designing broadcasting algorithms for mobile ad hoc networks, not only the inherent problems of dissemination algorithms must be considered, but also all the inherited ones from the ad hoc networks. Therefore, efficient communication in MANETs is not only crucial but also a very complex task.

This kind of protocols should reach as many devices as possible, even including devices in other partitions if the network is (as it is usually the case) partitioned, but at the same time reducing to the minimum the use of both the network and the device resources.

In order to avoid contextual limitations such as the broadcast storm problem, reducing the network resources used, or the number of collisions, protocols usually rely on some parameters that adapt the behaviour of the algorithms to the circumstances. Finding an optimal value for those parameters is difficult, and they are usually experimentally chosen. However, the performance of the protocol depends on the correctness of those value. Thus, finding the appropriate value is also very important when designing a protocol.

In this thesis, we focus on providing efficient communication in mobile ad hoc networks using broadcasting algorithms. For that, we have implemented a couple of efficient broadcast algorithms, and we have optimised their parameters using a experimental framework designed for that purpose.

1.3 List of contributions

The major contributions contained in this PhD thesis include:

1. An extensive state of the art in broadcasting protocols and energy aware algorithms is presented, providing also a new taxonomy. Additionally, a literature review in different optimisation techniques for problems dealing with ad hoc network is included.

2. The expenses of the creation and maintenance of a decentralised tree based topology in different types of ad hoc networks (mobile and vehicular) is studied.

3. Two different broadcasting algorithms are proposed, and compared to state of the art broadcasting techniques. The first approach relies on a tree topology, and the second one uses a cross layer design for reducing the energy consumption.

4. Creation of an experimental framework that gives clarity and modularity to the development process. It efficiently evaluates, validates and optimises any algorithm under different circumstances or using any tool.

5. Optimisation of the proposed broadcasting algorithm using a massively parallel iterated local search specifically designed for finding the best possible configuration of the algorithm. Additionally, the results obtained are validated using more complex metaheuristics (evolutionary algorithms).

6. Comparison the proposed iterative local search and two state of the art evolutionary algorithms in terms of some well known quality metrics: hypervolume, spread and epsilon.

1.4 Dissertation outline

The remainder of this document is organised as follows:

PART ONE : Mobile Ad hoc Networks and Literature Review

This first part of the dissertation starts with an introduction to the concept of ad hoc networks as well as its history in Chapter 2. A brief explanation of the different kinds of ad hoc networks, and the most common technology used in each of them is given. Chapter 3, gives an extensive review of some of the most relevant works in the literature that focus on the dissemination process. Additionally, an overview of the use of metaheuristics for solving problems in ad hoc networks is also presented in this chapter.

PART TWO: Research Objectives and Experimental Framework

Chapter 4 describes the research objectives we are pursuing during this thesis, and the mechanism we are using fro tackling them. The experimental framework proposed for evaluating, validating and optimising the proposed algorithms is introduced in Chapter 5.

PART THREE: Broadcast Algorithms

Two different approaches of broadcasting algorithms are proposed in this thesis. In Chapter 6, a novel broadcasting algorithm that uses an underlying tree topology for disseminating the broadcast message is proposed. Chapter 7 presents a cross-layer and energy aware approach that controls the transmission power of each node.

PART THREE: Optimisation Process

This problem is addressed in Chapter 8 using a multi-objective local search algorithm. Results of this local search are validated using two multi-objectives evolutionary algorithms in Chapter 9.

PART FOUR: Conclusions and Perspectives

This manuscript concludes the work presented during this dissertation in Chapter 10, presenting also some perspectives.

Part I

Mobile Ad hoc Networks and Literature Review

Chapter 2

Overview of Mobile Ad Hoc Networks

Contents

The first wireless communication network between computers was created in 1971 by Norman Abramson at the University of Hawaii, the AlohaNet [19]. It was composed of 7 computers distributed over four islands that were able to communicate with a central node in Oahu island using the radio communication. Additionally, the most well known random access protocol, ALOHA, was also developed and presented at that time [20]. The ALOHA channel is used nowadays in all major mobile networks (2 and 3G), as well as in almost all two-way satellite data networks [296].

Thanks to the reduction in the cost and size of the hardware needed, the wireless technology widely extends in our everyday life. The huge amount of devices that provide wireless technology nowadays, as well as the increasing number of people that not only carry a device with wireless capabilities but actually use it, make the field of wireless technology a key topic in research.

The current mobile wireless networks consist of wireless nodes that are connected to a central base station. When a device moves to a different geographical area it must connect to a different base station in order to continue with the service. Meaning that two nodes located in the same region cannot communicate unless there is a base station associated to that area. Researchers envisioned a possibility for communicating devices where the fixed infrastructure was not available, i.e., remote or disaster areas. This kind of networks are called ad hoc networks.

The term *ad hoc* has been extensively used during the last decades. According to the *American heritage dictionary of english language* it has two different meanings: (1) form for or concerned with one specific purpose; (2) improvised and often impromptu. These two definitions of the term *ad hoc* describe the purpose of a new kind of network that emerged with the wireless technology.

In an ad hoc network, all devices may also act as routers and forward packets to enable communication between nodes that are not in range. Two nodes are said to be in range when they are able to receive and properly decode packets sent by the other node.

Some examples where the deployment of an ad hoc network can be used and actually can be very useful are: relief in disaster areas, battlefield deployment, sensing areas, social events (e.g., a concert), etc. In these cases, devices can create a temporary network for a specific purpose, that is, an ad hoc network. When devices are mobile, they are called mobile ad hoc networks.

Ad hoc networks suffer from the typical problems of wireless networks such as interference, time varying channels, low reliability, limited transmission range, etc. Additionally, ad hoc networks face some specific characteristics we describe next.

1. self-organise: nodes must be able to create or join an existing ad hoc network by their own means. There is no infrastructure or central node(s) that is in charge of creating or maintaining the network;

2. decentralisation: nodes locally execute algorithms and take all decisions by themselves;

3. multi-hop: nodes must act as a router and relay messages to other nodes;

4. heterogeneity: any kind of device with wireless capabilities may be able to join the network;

5. energy constrained: devices usually rely on battery;

6. radio interferences: transmissions from nodes that are out of the transmission range are still received but with small reception power so that it is not possible to decode the signal. However, this reception is considered as interference and it affects by decreasing the signal to noise and interference ratio (SNIR) that is crucial for the successfully reception of packets;

7. limited network resources: the bandwidth in wireless is significantly lower than in wired networks. Additionally, nodes are energy constrained so that the node processing power is limited;

8. variable channel quality: the quality of the transmitted signal is influenced by the frequency used, the dimension of antennas and specially the environment itself, i.e. weather conditions, obstacles, physical phenomenons as attenuation or path loss, reflection, refraction, milti-path, scattering, fading, etc.;

9. hidden terminal problem: collisions might occur when two nodes that are not in range are targeting the same node. Nodes will detect an idle channel and transmit a packet creating a collision at the destination [329]. See Figure 2.1 for a graphical explanation. Both nodes C and B detect an idle channel and transmit at the same time a message to

node A. Node A will not be able to decode the message due to a collision. The RTS/CTS protocol is proposed in [176] to solve this problem (for point-to-point retransmission). When node C sends a *request to send* message (RTS) to node A. If node A is available, it sends the *clear to send* message to node C. In that case, node B will hear the CTS from node A, and refrains from transmission;

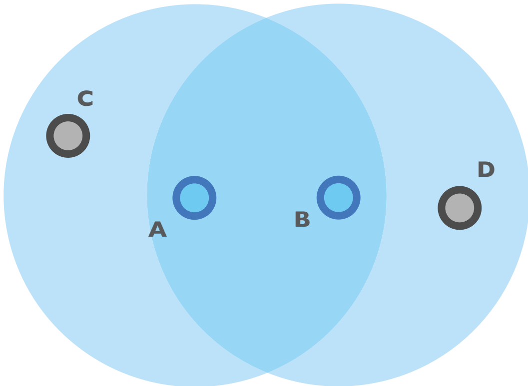


Figure 2.1: Example of the hidden terminal problem and the exposed terminal problem

10. exposed terminal problem: the RTS/CTS mechanism proposed for solving the hidden terminal problem provokes it. Two nodes in range can not simultaneously transmit even if the intended destinations are not the same [51]. See Figure 2.1. Node A wants to send a message to C and node B wants to communicate with D. However, after hearing the RTS message from node A, node B refrains itself from transmission even though the intended destination is not the same;

11. dynamism: the network is not only composed of mobile nodes, but also nodes can appear and disappear at any moment;

12. scalability: any node can join the network at any time, but no central unit is in charge of managing it, therefore, any algorithm for MANETs must be scalable;

13. security: the lack of central authority, the changing topology, and the vulnerability of the channel makes difficult guaranteeing secured communications.

Chlamtac *et al.* presented in [75] a classification of ad hoc networks in terms of the coverage of the devices (see Figure 2.2). They can be differentiated into five different classes:

- Body area network (BAN) is a communication network (usually wireless) composed of small wearable nodes (earphones, microphones) that provides connectivity between those devices. It is also extended to small sensors nodes implanted in the human body that collect information about the patient's health and send it to an external unit. The coverage needed is just to cover the human body, i.e. 1–2 meters.

- Personal area network (PAN) is intended to the communication of mobile devices carried by individuals, like smart phones, PDAs, etc. The range varies with the technology used, from 10 to 100 meters.

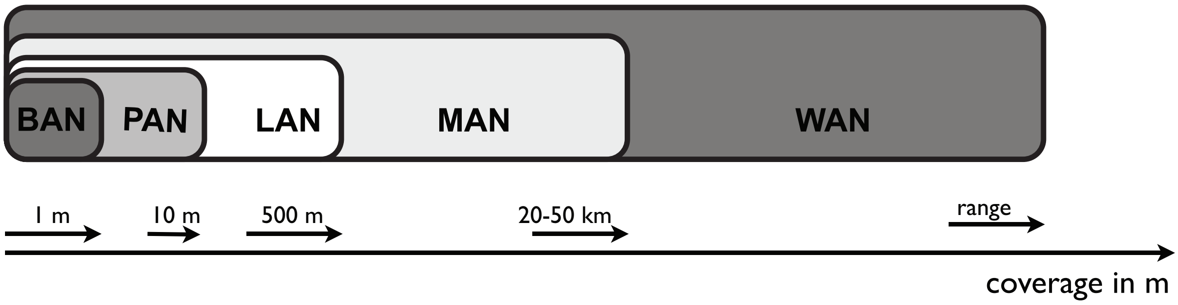


Figure 2.2: Classification of ad hoc networks in terms of the coverage area.

- Local area network (LAN) interconnects computer nodes among them or with peripheral equipments at high data transfer in a predefined area like an office, school, laboratory, etc. The communication range is restricted to a building or a set of buildings, between 100 and 500 meters.

- Metropolitan area network (MAN) spans a city or a large campus. It usually interconnects different LANs. The size is variable covering up to tens of kilometres.

- Wide area network (WAN) covers a large geographical area. It can relay data between different LANs or over long distances.

Both, MAN and WAN still need much more work to become a reality in a near future. There are many challenges that keep researchers working on the topic because they are not solved yet like communication beyond-line-of-sight, identification of devices, routing algorithms, etc. [155, 163, 166, 347].

Apart from this classification, the ad hoc networking field has three well defined research lines: (1) Mobile ad hoc networks, (2) Vehicular ad hoc networks, and (3) Sensor networks. The first one is defined as an ad hoc networks where devices do move, and includes all personal devices like smart phones, PDAs, laptops, gaming devices, etc. When devices move at high speeds, without energy restrictions and the network is able to use road side units for communicating, we are talking about vehicular ad hoc networks. Finally, in sensor networks devices are generally meant to acquire data from the environment and report it to a central node or gateway. Next sections give a more detailed view of these three types of ad hoc networks.

2.1 Mobile ad hoc networks

Mobile ad hoc networks, also called MANETs, are ad hoc networks where the devices conforming the network are mobile. Khan extended the previously mentioned ALOHANET including repeaters, authentication and coexistence with other possible systems in the same band. This new system was called the packet radio network, PRNET [172]. The PRNET project of the Defense Advanced Research Projects Agency, DARPA, started in 1973 and evolved through the years (1973-1987) to be a robust, reliable, operational experimental network. The nowadays called mobile ad hoc networks or MANETs, were first defined in

PRNET project. In [168], a detailed description of PRNET is presented and in [167] PRNET is defined as mobile ad hoc network.

Initially, MANETs were mainly developed for military applications. Specially for creating communication networks in a battlefield. In the middle of 1991 when the first standard was defined, IEEE 802.11 [355], and the first commercial radio technologies appeared, the great potential of ad hoc networks outside the military domain was envisioned. Apart from the military scenarios, all the previously mentioned applications for ad hoc networks (if we consider moving devices) are placed in this section. However, there are many others like emergency services, multi-user gaming, e-commerce, information services, mobile office, extending the cellular network, etc.

Advances in the technology made possible to have internet connection in portable devices. Mobile phones evolved to smart phones with big screens, cameras, GPS, bluetooth, high speed data access and a friendly operating system. At the end of 2013, the number of mobile devices will exceed the world's population, and by 2017 there will be 1.4 mobile devices per capita [247]. Moreover, meanwhile many people (not only industry) focused on developing applications for those smart phones, social networks such as *facebook* or *Twitter* appeared. The former had, in average, more than 655 million active users per day *facebook* during the month of March 2013 [15]. The latter has 140 millions of active users and 340 millions of Tweets a day [316] just after 6 years. No one could have predicted the amazing welcome of social networking. Actually, those applications are not only used in computers but also in smart phones and tablets, increasing the mobile data traffic. It is expected that in 2016 the mobile data traffic will be more than 8 times higher than in 2012, and only 0.3% of this traffic will be due to VoIP [247]. In Figure 2.3, the forecast evolution of the mobile data is shown.

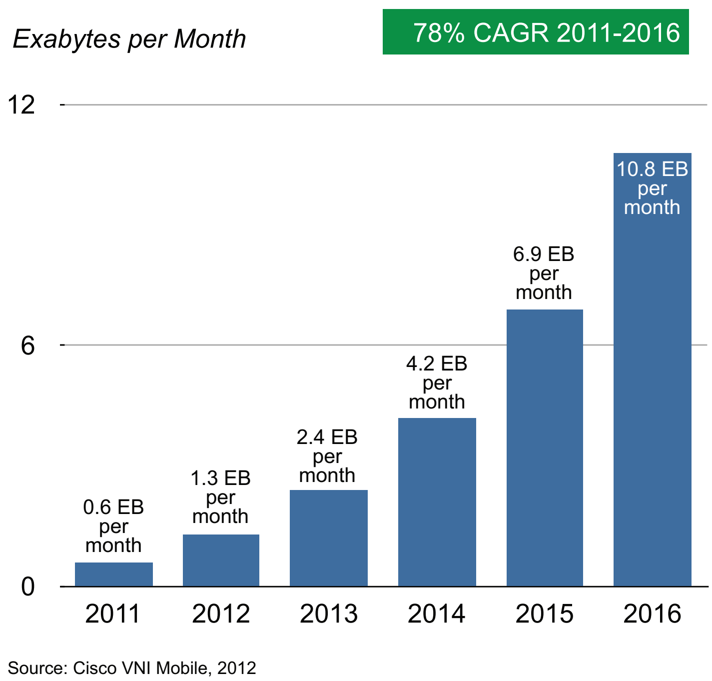


Figure 2.3: Cisco Forecasts of Mobile Data Traffic up to 2016.

With such numbers, the cellular network will be soon saturated. For alleviating this problem, part of the mobile data traffic can be delivered by a complementary network. This mechanism is known as *3G Offloading*. There exist some works presenting mobile ad hoc networks as this complementary network [25, 278].

Some of the main characteristics of mobile ad hoc networks that makes challenging their design are mentioned below:

1. the lack of any infrastructure forces the node to perform network setup, management, self-healing, neighbor discovery, etc.;
2. every node must have routing capabilities for communicating nodes out of range;
3. energy constraints as devices depend on battery;
4. network resources restrictions as in wireless network the medium is shared (limited bandwidth, collisions, etc.);
5. network partitioning due to the limited transmission range and the mobility of devices;
6. dynamic topology as the links are time varying because of the mobility of the nodes, and the appearance and disappearance of devices.

2.1.1 MANET Technology

Although vehicular ad hoc networks, and mobile sensor networks can be seen as a subclass of mobile ad hoc networks, the nodes composing the network are completely different. Therefore, the technologies used for each of the previously mentioned types of ad hoc networks are different. The main idea of mobile ad hoc network is connecting any device in range (considering WLAN). The most common technology that gives service for computer communication in WLAN is Wi-Fi, that is already included in most of the commercial devices making it the most suitable technology for mobile ad hoc networks.

Wi-Fi is a technology defined by the Wi-Fi Alliance [9] that allows wireless communication based on the IEEE 802.11 standards. The first IEEE 802.11 standard was published in 1997 [355], and there were two updates until the time of writing, one in 2007 and another in 2012. It uses two frequency bands, 2.4 and 5 GHz. There exists a big variety of amendments to each of the standards that focus on different characteristics in wireless communication. Some examples are IEEE 802.11n that allows MIMO antenna (multiple-input multiple-output), or the IEEE 802.11s for mesh networking, or IEEE 802.11aa for video transport stream. For a complete view on the amendments and the time line, please refer to [355].

The most common used standards are IEEE 802.11b (1999) and IEEE802.11g (2003) which are amendments to the original standard IEEE 802.11-1997. They both work on the 2.4 GHz band, being the latter more recent with higher data rate but still fully compatible with IEEE 802.11b hardware. The IEEE 802.11a works on the 5GHz band. IEEE 802.11a also appeared in 1999, however, due to the reduced cost and the fast arrival on the market of the IEEE 802.11b, it was widely adopted, making the acceptance of the late IEEE 802.11g, that was fully compatible, very easy and fast.

The IEEE 802.11n (2009) is an amendment to the IEEE 802.11-2007 that includes MIMO antenna (multiple-input multiple-output), a significant increase in the throughput (from 54 Mbit/s to 600 Mbit/s) and operates in both frequency bands. These amendments are the most used versions of the IEEE 802.11 standard that provide wireless capabilities for everyday life devices.

2.2 Vehicular ad hoc networks

Vehicular ad hoc networks, hereinafter, VANETs, are ad hoc networks where the devices conforming the network are vehicles. In VANETs, apart from the nodes there can also be base stations or fixed infrastructure called road side units.

VANETs should not be confused with Intelligent transportation systems (ITS). ITS cope with all kind of communications inside the vehicle, between cars or with the road side unit, but are not limited to road transport. It also includes rail, water and air transport. Thus, VANET is a component of ITS.

The idea of a network composed of base stations and vehicles is not new. Literature reveals that much effort has been applied to vehicular networks. Already in 1952, M. R. Friedberg discussed about how to place a mobile antenna on a vehicle in order to communicate with the driver [109]. Not only researchers were interested but also companies. In 1966, General Motors Research Laboratory was already designing a real-time system for safety aid. It was able to send voice messages alerting devices about dangers ahead. Later, they were also considering systems not only for making driving safer, but more convenient and more enjoyable [139]. At that time, they were already proposing a two way communication system, able to obtain road information but also able to ask for assistance. The system also provides: (1) audio signs for receiving emergency messages and road conditions in the vehicle; (2) visual signs reproducing roadside traffic signs; and (3) navigation assistance of a preselected route. An extensive review on studies related to motorist information was presented in [242].

The PROMETHEUS Eureka program (1985-1993) was intended for developing an intelligent co-pilot that helps the driver, but not creating an autonomous car. More than 60 participants from 5 different countries where involved and almost all car manufactures. The project was divided into different sub-programs: PRO-CAR, PRO-NET and PRO-ROAD. PRO-NET system depends on the communication links between vehicles [122]. In 1988, in the framework of the project they proposed vehicle to vehicle communications for increasing the driving security [84]. In 1989, the Commission of the European Community launched the DRIVE program. The objectives were similar to the ones proposed in PROMETHEUS: improve road safety, traffic and transport conditions and reduce the environmental pollution; but whilst PROMETHEUS focuses on assisting the driver, DRIVE focuses on the infrastructure. A review in both projects and their differences can be found in [122].

Anwar et al. proposed in [34] the use of packet radio networks for car to car communication in densely populated cities. They are considering mobile radio networks (MRN) where there are no central stations, thus, they are talking about a mobile ad hoc network. They created a scenario with one and two-way roads, traffic lights, buildings, collisions and shadowing. In the same conference, Davoli et al. presented an architecture and a protocol for car to infrastructure communication using the packet radio network [88]. But, the term VANET was first coined by Kenneth B. Laberteaux, who also conducted and promoted the first VANET workshop in 2004 as general co-chair [140].

Vehicular ad hoc networks can be considered as a subset of mobile ad hoc networks but they have specific characteristics that distinguish them from typical mobile ad hoc networks and that make challenging its design. Some of them are mentioned below:

1. very changing topology because devices move at very high speeds varying from 0 to 180 km/h. The changing topology comes out onto network partitioning. Not only due to the high speeds of vehicles but also because of arriving to rural areas were the density of devices is lower than in city centers;

2. variable network density mostly depending on the time and the area. At rush hours the traffic is high and it is usually low in rural roads;

3. as consequence of high speed and the limited transmission range, the link availability is low (less than 1 minute). Not only for devices moving in opposite directions but also cars driving in the same directions;

4. unlike mobile or sensor ad hoc networks, vehicular ad hoc networks are not energy constrained;

5. vehicles do not move at random, they move along lanes following routes. Additionally, a specific device might have predictable routes. Everyday, the driver goes from home to work and come back, at approximately the same hour;

6. there exist two different operation modes: (1) car to car communication and (2) car to infrastructure.

In 1999, the U.S. Federal Communication Commission allocated 75 MHz of Dedicated Short-Range Communication (DSRC) spectrum at 5.9 GHz to be used exclusively for vehicle-to-vehicle and infrastructure-to-vehicle communications [79]. DSRC technology allows high speed communications between vehicles and the road side or between vehicles that might be separated up to 1000 meters. There exist differences in the frequency allocation between North America and Europe, but the intention is to be able to use the same antenna and transmitter/receiver. Different organizations like the Institute of Electrical and Electronic Engineers (IEEE), International Standard Organization (ISO) or Car-to-Car Communication Consortium / GeoNet are working on developing an architecture for VANETs. There is no agreement between the different organizations on which of the different proposals is more convenient for vehicular networks, thus, each of them is working on their own proposal: WAVE by IEEE, CALM by ISO and C2CNet by C2C Communication Consortium. A general overview on the three schemes is given next.

2.2.1 VANET Technology

1. Wireless Access in Vehicular Environment (WAVE) *The IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments (WAVE), defines the architecture, communications model, management structure, security mechanisms and physical access for high speed (up to 27 Mb/s) short range (up to 1000m) low latency wireless communications in the vehicular environment. The primary architectural components defined by these standards are the On Board Unit (OBU), Road Side Unit (RSU) and WAVE interface.* [276].

IEEE 1609 is composed of different standards tackling different layers that are already published, i.e., IEEE1609.1 is the resource manager, IEEE 1609.2 copes with security services, IEEE 1609.3 with network services and IEEE 1609.4 is for channel switching. However, part of this family of standards is still under development as IEEE 1609.0 the architecture, IEEE 1609.5 the communication manager, IEEE 1609.6 remote management service, IEEE 1609.11 for secure electronic payment or IEEE 1609.12 identifier allocations, at the time of writing.

In 2003, IEEE and American Society for Testing and Materials (ASTM) adopted a first version of the DSRC PHY [40] that was based on IEEE 802.11a. In 2004, they agreed in creating 802.11p amendment within the IEEE 802.11 Working Group for adding wireless access in vehicular environments (WAVE). The 802.11p [16] is built on its predecessor ASTM E2213 and it defines the required enhancements to IEEE 802.11 for supporting ITS applications.

Additionally, SAE international standards J2735 [160] and SAE J2945.1 [286] (still under development) define a set of message formats for vehicular applications, and the rules (like rate or power constraints), respectively. Those standards operate with applications using DSRC/WAVE but they have been designed to potentially be also used with other wireless communication technologies.

Depending on the application requirements DSRC/WAVE can operate using the well-known IPv6, UDP and TCP (traditional internet) defined by Internet Engineering Task Force (IETF), or using WAVE Short Messages Protocol (WSMP) defined in IEEE 1609.3. The non-IP WSMP aims at exchanging non-routed data as safety messages.

The architecture proposed by IEEE has the IEEE 1609.x family as core standard, the IEEE 802.11p at the physical and MAC layers and the SAE J2735 and SAE J2945.1 at the top of the protocol stack. A detailed explanation of the architecture of the IEEE standard for DSRC can be found on [181].

2. Communication Access for Land Mobiles (CALM)

   ISO TC204 WG16 is developing a family of International Standards based on the CALM (Communications access for land mobiles) concept. This family of standards specifies a common architecture, network protocols and communication interface definitions for wired and wireless communications using various access technologies including cellular 2nd generation, cellular 3rd generation, satellite, infra-red, 5 GHz micro-wave, 60 GHz millimeter-wave, and mobile wireless broadband. These and other access technologies that can be incorporated are designed to provide broadcast, unicast and multicast communications between mobile stations, between mobile and fixed stations and between fixed stations in the "Intelligent Transport Systems" (ITS) sector [314].

   The CALM standards are communication-centric that block out the application layer from the communication protocols. The idea behind it, is that depending on the application needs, the availability of the different technologies, the channel conditions, etc. the CALM system will communicate using the most suitable communication technology. It is an heterogeneous system where devices have different interfaces and are able to support handover between the different technologies supported in CALM (cellular, infrared, DSRC, satellite, etc.). This is known as media independent handover.

As of year 2013, the set of CALM standards is still under development but some research projects like COOPERS [7], or CVIS [13], already consider this technology. The CALM architecture (ISO 21217) is composed of six parts: *applications*, *management*, *security*, *facilities*, *networking & transport* and *access*. As it is based on a modification and an extension of the layered Open Systems Interconnection (OSI) model [382], there exists a correspondence between the OSI layers and some of the previously mentioned parts. The first two layers of the OSI model are included in *access*; layers three and four correspond with *networking & transport*; and *facilities* contain the rest layers of the OSI model. For a more detailed explanation of the model, please refer to [315].

In CALM, the car is not only considered as one single device but more as a whole in-vehicle network with a variety of embedded and interconnected devices. The architecture must be able to cope with multiple technologies simultaneously and also with network mobility (NEMO). As vehicles move, the gateways to internet change but the internet connectivity to the in-vehicle network must be uninterrupted.

Similarly to WAVE, CALM operates using the IPv6 networking protocol, but for time critical safety messages a specific non-IP protocol called FAST is used (ISO 29281). FAST supports vehicle-vehicle and vehicle-roadside communications with a very light header.

3. C2C Network

C2C Network (C2CNet) is a communication layer defined by the Car-2-Car Communication Consortium [6] specifically for car-to-car communication. As it was first defined in [80], the C2C Communication Layers' architecture differentiate between three different type of applications: Active safety, traffic efficiency and infotainment. The first one relies on IEEE 802.11p and does not make use of the TCP/IP protocol. It uses a specific C2C Network and C2C Transport for vehicular communications. The traffic efficiency applications can use both the IPv6 or the C2C Network over the conventional wireless LAN technologies based on IEEE 802.11 a/b/g/n. For the last kind of applications the TCP/IP (or UDP) will be used on top of other wireless technologies like GPRS or UMTS.

The C2C-CC system does not force all vehicles to be equipped with all the previously mentioned technologies, but at least the on board unit must be able to communicate using the IEEE 802.11p radio technology for safety applications.

The C2C Network layer [269] is located between the network and the link layer. It supports geographical addressing and routing. The C2C header contains geographical locations. It does not use IP address, but IPv6 packets can be transmitted by encapsulating the IPv6 packet into a C2CNet packet (IPv4 will also be supported). That was defined in the GeoNet project [5] "IPv6 over C2CNet".

At the time of writing, these three architectures are still under development. Therefore, the final architecture will possibly differ from the brief overview given above. Moreover, the final decision about which standard to adopt may depend on car manufactures and authorities considering various technical, business and political aspects. A more detailed comparison between the three architectures was presented in [241].

2.3 Sensor networks

Nowadays, sensor networks are widely used in practice, for managing traffic lights, environmental conditions, system failures, security systems, etc. But one of the main areas of sensor networks is in the line with medicine, and it is most probably one of the oldest sensor applications. Already in the early 50s, doctors were using sensors for monitoring patients like electrocardiographs, blood pressure recorders, electroencephalograph, etc. In 1956, M.H. Davis [87] proposed an intercommunication system for communicating all the operating team, as well as, for stimulating the patient during the surgical treatment of epilepsy. Moreover, they were exploring the possibility of a wireless system at that time.

Indeed, in 1957 Mackay and Jacobson described a small unit (0.9 x 2.8 cm) that could be easily swallowed [232] able to simultaneously transmit pressure and temperature signals for two weeks. A survey on the techniques available at that time can be found on [231].

The advances during decades in micro electrical-mechanical systems (MEMS) technology made possible low cost and small size wireless sensor nodes. A sensor network is an ad hoc network composed of a large number of devices geographically distributed, able to monitor different environmental or physical conditions (the data of interest). Each node usually gets the raw sensed data, processes it locally and sends it to the node responsible for the data aggregation, the sink or gateway (see Figure 2.4). The user is able to access the gathered data from the gateway. There are many different configurations of sensor networks. It is possible a network with a single sink, where all the devices send the collected data to the sink and it uses the information locally. There could also be a gateway that connects the sink to other networks like Internet, so that the user can access to the data gathered (in this case, the gateway can also act as sink). For scalability reasons, having more than one sink is desirable. WSN can be programmed as self-organising, according to different network topologies (star, linear, clustered, mesh, etc.) based on the specific application requirements. Akyildiz et al. presented in [23] an extensive survey on sensor networks.

Sensor networks have been widely used. Initially, sensor networks were mostly limited to military applications (surveillance, intrusion detection, targeting systems, etc.). An example of the early military use is the deployment of the seismic intrusion sensors in the Vietnam war around the camp as part of the intrusion warning system [230]. Chong et al. explained in an invited paper [76] the history of sensor networks, the technology and the challenges. Nowadays, thanks to the reduction in cost and size, they are being used in many different applications like in health for monitoring patients, for environmental measurements (like temperature, pollution, pressure, humidity, etc.), for monitoring disaster areas, in commercial for managing inventory, intelligent buildings, vehicle monitoring, animal monitoring or machine monitoring.

The size of the node can vary depending on the application. In [171], Kahn et al. propose a prototype called Smart Dust, so small that could be suspended in the air for hours or even days (the volume is a few cubic milimeters). Regarding the mobility, the nodes are typically fixed but in applications like data acquisition of twister where the sensor nodes go inside the tornado, mobility is a key feature for capturing information.

In ad hoc networks, the network topology is not known a priori, thus, it must be constructed in real time. In sensors, the area that has to be sensed is sometimes known a priori, requiring optimal nodes distribution. Moreover, due to new deployments of sensors or node failures, the topology must be updated periodically. In these networks, where nodes only

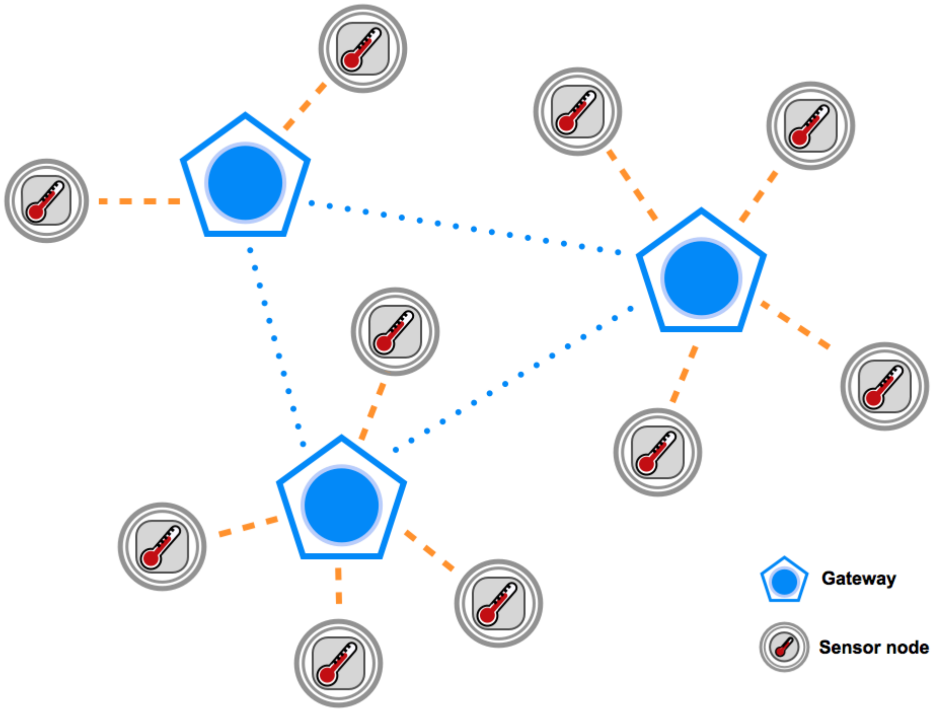


Figure 2.4: Example of a possible sensor network.

communicate with neighbours, distributed algorithms are attractive because they are robust to topology changes. In [76], authors claimed that decentralised algorithms are preferred to centralised one (even if they can collect data from multiple sensor nodes) because the latter are less robust, reliable and have higher communication cost.

There are specific and challenging key features when designing a wireless sensor networks that must be taken into account. Here, we mention some desired characteristics:

1. energy: the tiny size and the constant sensing activity of the devices, makes energy consumption the critical factor in its design [101]. Some decisions must be taken in order to balance the performance of the sensor network and the resource utilisation. For example, gathering sensed data from a higher number of nodes will give more accurate results, but more communication resources are needed (i.e. energy);

2. low latency: depending on the application the data gathered can be already out of date in high latency networks. The delay the raw sensed data experiences from its acquisition until its utilisation can be crucial depending on the application (i.e. patient monitoring);

3. scalability: the number of nodes deployed in an area can vary from tens to thousands of sensors, thus, algorithms used must be able to provide the desirable performance regardless the size of the network;

4. reliability: sensor nodes can fail due to the battery lifetime or because of the extreme environmental conditions, therefore, the algorithms designed must be resilient to failures,

and the network self-healing.

5. deployment: optimal distribution of the sensor over a spatial area.

There are some important differences between mobile ad hoc networks and sensor networks and also between their applications, that makes no straight forward reuse of algorithms and protocol of MANETs in sensor networks. The suitability of those algorithms must be checked before its actual implementation. Below, we mention some of those differences.

- In ad hoc networks the terminals are smart with high capacity while in sensor they are simple and the capacity rate in most of the applications is very low (few bytes);
- unlike in ad hoc networks, in sensors not all the nodes act as routers;
- although energy is considered as a key feature, capacity is also a relevant characteristic that must be taken into consideration when designing an ad hoc network; while in sensor networks the energy is the most important restriction that must be always considered for the design of a sensor network [47].

In sensors, communications protocols must be designed considering the energy restrictions. Indeed, the energy consumption needed for transmitting data is much bigger than the one needed for processing the data. However, the signal processing must not be neglected from the energy consumption as processing data sometimes can take much longer than transmitting the data, and therefore, consumes more than the transceiver in idle mode. Additionally, when the sleeping mode is assumed in sensors, suitable synchronization is needed for having an efficient communication between nodes.

As each sensor must sense, process and communicate using a limited amount of energy, a cross-layer design that takes into consideration all these requirements (communication protocols, signal and data processing) will be a good approach.

2.3.1 Sensor Technology

Unlike MANETs or VANETs, sensors are being used in some real world applications, thus, there exist many different technologies for sensors depending on the necessities of the targeted application. Next, we introduce some of the most well known technologies and standards that are available at the time of writing the thesis.

1. IEEE 1451

   The National Institute of Standards and Technology (NIST) [11] is developing a family of smart transducer interface standards IEEE 1451 *that describes a set of open, common, network-independent communication interfaces for connecting transducers (sensors or actuators) to microprocessors, instrumentation systems, and control/field networks. The key feature of these standards is the definition of Transducer Electronic Data Sheets (TEDS). The TEDS is a memory device attached to the transducer,which stores transducer identification, calibration, correction data,measurement range, and manufacture-related information, etc. The goal of 1451 is to allow the access of transducer data through a common set of interfaces whether the transducers are connected to systems or networks via a wired or wireless means.* [252].

IEEE 1451 allows the sensors to have capabilities for: self-identification, self-description, self-diagnosis, self-calibration, location-awareness, time-awareness, data processing, reasoning, data fusion, alert notification, standard-based data formats, and communication protocols [310]. It also provides plug-and-play capabilities. The definition of the Transducer Electronic Data Sheets (TEDS) is the key feature that can be seen as an identification card that contains specific data of the transducer (including manufacturer information) allowing the sensor to connect to different networks.

2. IEEE 802.15.4

   In 2003, the original standard of the Institute of Electrical and Electronics Engineers (IEEE) for low rate personal area networks (LR-PAN), IEEE 802.15.4 was approved. Unlike IEEE 1451, it only defines the two bottom layers of the OSI model considering very low power consumption, low complexity, and low cost. After this standard, the improved version was approved in 2006 (IEEE 802.15.4b), and in 2007, location capabilities were added in IEEE 802.15.4a. In order to make it compatible with the bands available in China and Japan, in 2009, the 802.15.4c and 802.15.4d were approved. Recently, in 2011, the IEEE 802.15.4 was extended, the ambiguities removed and improvements included [129].

   The network can have two different topologies: (1) Star, and (2) Peer-to-Peer. Moreover, two types of devices are defined: (1) Full-Function device (FFD) and (2) Reduced-Function device (RFD). The FFD has all network functionalities, while the RFD has low resources and is capable of very simple applications. There must exist at least one FFD for coordinating the network (PAN coordinator). In the star topology, nodes can only communicate with the PAN coordinator, while in the peer-to-peer configuration any two nodes in range can connect, and they are able to self-organise, which are the bases for an ad hoc sensor network.

   IEEE 802.15.4 serves as the low layers of many different specifications like ZigBee, 6LoWPAN, Wireless HART, ISA-SP100, MiWi, etc. We will briefly mention in the next subsections some of them.

3. ZigBee

   ZigBee is a standard based network protocol created by the ZigBee Alliance [10]. It is based on the 802.15.4 standard and defines layer 3 and uppers in the OSI model. The main purpose is to create a network with low rate and low power capabilities that still covers a long area and that gives extra features like security. In ZigBee there are two possible access modes: beacon and non-beacon. If the beaconing is not enabled, any node can transmit data whenever the channel is free. When beacons are enabled, the PAN coordinator assigns a time slot to every device for transmitting and sends beacon signals to synchronise all devices under its control.

   Three different topologies are considered in ZigBee: (1) Star, (2) Cluster Tree topology, and (3) Mesh topology. The cluster tree topology is similar to the star but there exists the possibility that other nodes rather than the PAN coordinator are able to communicate with each other. Unlike in the first two ones, in the mesh network any node can communicate with any other in range. Beaconing is not allowed in this latter topology.

The ZigBee Alliance offers two specification: ZigBee and ZigBee RF4CE. The former is intended for mesh networks offering all the features of ZigBee as self-configuring, self-healing, etc. Additionally, two feature sets are available: ZigBee and ZigBee PRO (being low power consumption and large network of thousands devices). The latter aims at providing simple device-to-device topology, thus, reducing the cost and the complexity. For a more detailed description of the ZigBee technology refer to [30, 123]

4. 6LoWPAN

The idea of having all devices IP-enabled connected to the internet and all the internet services monitoring and controlling those devices is called *Internet of things* and was first mentioned in 1999 [37]. It envisions trillion of nodes working under the Internet protocol IPv6. The problem rises when dealing with low power, low bandwidth, battery dependent devices, what is called the wireless embedded internet.

IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) working group of IETF is defining a set of standards for adapting IPv6 to those resource limited devices. In [303], we find a formal definition:

Definition 2 (6LoWPAN)**.**
6LoWPAN standards enable the efficient use of IPv6 over low-power, low-rate wireless networks on simple embedded devices through an adaptation layer and the optimization of related protocols.

IPv6 header is compressed and some functionalities are simplified, so that, IPv6 packets can be transmitted over a IEEE 802.15.4 network. In this case, the topology consider is a mesh.

At the time of writing several proposals are available. A more detailed explanation of them can be found in [371].

5. Bluetooth

In 1994, engineers at Ericsson invented the bluetooth, founding the Bluetooth Special Interest Group (SIG) in 1998 to expand and promote the concept [130]. But it was not until 1999 when the first specification was published.

The main idea of Bluetooth was to enable wireless information transfer between electronic devices via short-range ad hoc radio connections in wireless personal area network. It allowed the design of low-power, small size, low cost radios that can be embedded in existing portable devices. In [137], the bluetooth radio system and its ad hoc capabilities were presented.

Bluetooth works in master-slave mode, where the master is able to communicate with up to seven devices at the same time. The bluetooth ad hoc network formed by the master device and the slaves is called a piconet.

From its creation, different version of Bluetooth were released (v1.0, v1.0B, v1.1, etc.). At the time of writing this book, the last published version is Bluetooth v4.0 which includes *classic* Bluetooth technology, Bluetooth low energy technology (BLE) and Bluetooth high speed technology, which can be used combined or separately [337].

In their early stages, although being similar technologies focusing on short-range wireless communication, Bluetooth and ZigBee were aiming at different objectives. ZigBee had lower power consumption and was able to support larger network, while bluetooth had higher bit rate, what clearly differentiated the application fields for each of them. While Bluetooth was used for mobile devices, and peripherals, ZigBee focused on home automation and medical sensors. Lately, Bluetooth v4.0 includes BLE also aimed at very low power applications.

6. Wireless Industrial Automation System

   Both ISA100 or ISA-100.11a [14, 251], and WirelessHART [108] are specific for the process automation and manufacturing industries.

   WirelessHART, the first specification for wireless field instruments was released by the Highway Addressable Remote Transducer (HART) Communication Foundation (HCF) [8] in 2007.

   ISA-100.11a was started by the International Society of Automation (ISA) [161] in 2008, and it was intended to provide reliable and secure wireless operation for non-critical monitoring, alerting, supervisory control, open loop control, and closed loop control applications.

   There are many differences between the two standards. In WirelessHART, all field devices and adapters are routers capable of forwarding packets to and from other devices in the network, enabling a mesh network topology, while in ISA100.11a a node can have router capabilities or not, what means that not all devices are able to allow a new node to join the network. On the one hand, in WirelessHART there are a few optional parameters making it less flexible than ISA100.11a which has a complex specification with many parameters. On the other hand, the lack of flexibility makes easier the interoperability between different devices in WirelessHART. Additionally, as WirelessHART is an extension of the HART protocol, it is limited to this communication protocol. However, ISA100.11a is able to tunnel many different protocols even supporting IPv6 using 6LoWPAN. For a more detailed comparison between both protocol refer to [259].

Chapter 3

Literature Review

Contents

In ad hoc networks where the infrastructure is inexistent, a node that wants to communicate with another that is not within its transmission range, has to rely on intermediate nodes, provoking the appearance of a multi-hop network. That means the nodes collaborate and act as routers, i.e. forward packets that are not intended for themselves to other nodes.

As already mentioned in Chapter 1, we focus on broadcasting algorithms as they are essential in any communication network, and used by many other protocols and applications. The concept of broadcasting in mobile ad hoc networks differs from the conventional networks. In MANETs, it is not possible to guarantee full coverage anymore due to the network partition, the shared medium, the mobility of the devices, etc. The latency for the same reason is not bounded: it is not known when the last node will receive the message (if ever) as the number of hops is not known. Additionally, the nodes are energy constrained, so that not only energy aware protocols are desired, but also protocols that consider the use of the network resources.

Therefore, many challenges must be addressed when designing communication protocols. In order to make them efficient and adaptive to different possibles scenarios, the algorithms usually rely on different parameters or thresholds that change their behaviour according to the current circumstances. Fine tuning the values of those thresholds is a complex and critical task, as protocols are usually highly sensitive to small changes. Slightly changing the value of one threshold can provoke a considerable change in its behaviour. In this thesis, we address the fine tuning problem using approximation algorithms.

In this chapter, we first review the state of the art in broadcasting techniques, and later, we give an overview of existing works that apply approximation algorithms for solving problems in ad hoc networks.

3.1 Classification of the studied systems

Before reviewing the literature, we briefly present some of the main characteristics that differentiate the algorithms we are explaining. We are considering if there exists a central node that is in charge of managing the system, what kind of network information is used, whether or not the execution of the algorithm depends on random variables, and also if the algorithm is executed at runtime or before hands.

3.1.1 Centralised & decentralised systems

A *centralised system* consists of a central unit that decides on behalf of the whole system using global knowledge, or the different network nodes locally make a partial decision, but send the information to the central unit (or decision maker), that will decide in terms of the information gathered from all the nodes. It requires significant coordination between the components as well as communication overhead and delays. Additionally, the whole system depends on the central unit for the proper functioning, therefore, a failure on this unit implies the failure of the complete system.

On the contrary, when each node locally executes an algorithm, and according to the results obtained, the node modifies its own behaviour, it is called *decentralised system*.

In MANETs, the use of centralised systems at run time is not realistic, as the existence of the central unit is contrary to the essence of ad hoc network.

3.1.2 Global or local knowledge

Considering the information used during the process, it is possible to differentiate between algorithms that use information about the whole network, or just local information gathered by the nodes locally.

It is said that algorithms use *global knowledge* when information of the complete network is required, e.g. when every node uses the position of every single node in the network for making decisions.

However, when the algorithm only uses local information, or information that can be gathered by itself, it is said to use *local knowledge*. This local information does not only concern to the node itself, but also information from neighbouring nodes obtained by exchanging beacons, messages or just eavesdropping.

Unless the knowledge is acquired before hands (some specific cases in sensor networks for example), if there is no central unit, nodes need to exchange and collect the information from all other nodes in order to get the global knowledge. In very small networks, that could be done by using the beacons, but as density grows this mechanism is not scalable and becomes unrealistic. Thus, generally in MANETs, nodes are very unlikely to obtain global knowledge.

3.1.3 Deterministic & stochastic process

It is also possible to differentiate in terms of the predictability of the algorithms: *deterministic* and *stochastic* approaches.

On the one hand, a process is considered *deterministic* when no random decisions are taken. That is, given a particular input (or scenario), the output or the result is always the same. Its behaviour is predictable.

On the other hand, when there are random choices a process is said to be stochastic. Two executions of the same process in the same conditions can give different results.

3.1.4 Online & offline techniques

Regarding the optimisation process, literature reveals two different approaches when solving problems in mobile ad hoc networks: online and offline techniques. As explained in [55, 374], the main difference between them lies on the moment when the optimisation algorithm is applied.

Online optimisation algorithms are used for correcting behaviours or making decisions during runtime, trying to find the best next step. They can be implemented either in the network node or in a central unit, but usually requires intensive computation. Therefore, it must be considered the benefit of using these techniques in energy constrained devices, or using a centralised infrastructure which is contrary to the essence of ad hoc networks.

Offline optimisation algorithms are executed beforehands. The main goal is to find the best possible configuration, settings, decisions, etc., that will be later used during runtime. The algorithm stops after performing a predefined number of generation or when the optimal value is found (in case it is known). The quality of the solutions found are usually tested by simulations, thus, it directly depends on the modelling of the system. However, there is a compromise between the accuracy of the model and the optimisation time. These offline approaches are useful when the system does not need to adapt to changes during runtime.

3.2 Broadcast algorithms

It is important to properly define the broadcast problem before explaining the mechanism for achieving it. In Definition 3, we first explain what broadcasting is and then, we define the broadcast problem.

Definition 3 (Broadcast)**.**

Broadcast is an operation mode in wireless medium whereby messages are sent to all neighbouring nodes [48].

Definition 4 (Network wide broadcast problem).

In the network wide broadcast problem, a single node sends a packet to all other nodes in the network [350]. The different existing techniques for solving this problem are called broadcast algorithms.

From now on, we will refer to network wide broadcast by 'broadcasting' for the reminder of this thesis.

For disseminating a message in the already mentioned multi-hop networks, nodes must forward packets that are not intended for themselves and act as routers. For performing the network wide broadcast, the straight forward approach is that all nodes resend all the packets received. This is known as *flooding*. The main problem associated to this technique is the *broadcast storm problem*, i.e. as the density of the network increases, also increase the number of collisions, the packet loss, the contention of the shared medium, the traffic, the network resources, etc. In dense networks, the *broadcast storm problem* can even lead to the network congestion [249]. In such resource constrained networks, this approach is not advisable as it not only waste the battery life of the node, but also the network resources.

Addressing broadcast in mobile ad hoc networks needs a complete different perspective with respect to conventional networks. Wired networks are static, have reliable and predictable links that are unicast and transmission do not interfere with each other. However, wireless networks suffer from unique problems like low throughput, dead spots, inadequate support for mobility, high bit error rate and their characteristics could vary over short time scales. Additionally, MANETs have a huge variety of specific characteristics that differentiate them from any other instances in the networking paradigm. Some of the most relevant ones where presented in Chapter 2. Literature reveals many different works trying to overcome those problems and presenting different approaches for efficiently disseminating a message in MANETs.

First, we briefly present some existing classifications of broadcasting algorithms, that will help to have a global view of the related work reviewed afterwards.

Williams and Camp classified the broadcasting algorithms into four different families: Simple Flooding, Probability Based Methods, Area Based Methods and Neighbour Knowledge Methods [350] .

1. In **Simple Flooding**, a source node starts the dissemination process and all nodes rebroadcast the message exactly once. There is no intention of reducing the number of forwarding nodes, and network information is not required.

2. In **Probability based methods** for reducing the number of retransmissions, a node resends the received message with a predefined probability. It does not need any network information. Simple Flooding can be considered as a specific case of this family where the forwarding probability is set to one.

3. **Area Based Methods** calculate the additional area covered in case of rebroadcasting the received message. The decision is made in terms of a predefined threshold. In this technique, the location of the nodes or the distance to the source node is needed. This can be achieved either by equipping the nodes with GPS, triangularisation or using the signal strength.

4. **Neighbour knowledge methods** use information about the neighbours for deciding to rebroadcast the message or not. The neighbour knowledge is obtained by the using the *hello message* or *beacon*. Every node periodically sends this beacon so that all neighbours around are aware of its presence. More precise knowledge is obtained if some information is included in this beacon, e.g. the list of neighbours in order to have 2-hop neighbours knowledge.

Stojmenovic and Wu proposed another classification for broadcasting algorithms in [318]. The taxonomy presented is listed here and explain after: determinism, network information, reliability, *hello message* content, and broadcast message content.

1. **Probabilistic and deterministic** approaches are differentiate whether decisions rely on randomness or not.

2. **Network information** is divided into global or local knowledge as it was previously mentioned. A more detailed classification of this specific case was proposed in [359]. There four different type of information are available:

    - *global* information: any centralised or decentralised algorithm that uses global information of the system;
    - *quasi-global* information: when an algorithm uses only partial global information, i.e. there is global coordination, the information starts at a central unit and sequentially propagates to the entire network;
    - analogy, *quasi-local* information algorithms use mainly local information and occasionally, partially global information, i.e. sequentially propagation is still assumed but no central unit;
    - a *local* distributed algorithm only uses information obtained locally.

3. A broadcast algorithm is said to be **reliable** if it can reach all nodes in the network, otherwise it is *unreliable*.

4. The ***beacon* content** can include different information so that it is possible to react to different situations. It can include the node ID, the degree, the position, the neighbours, etc.

5. The **broadcast message content** can be altered and can include information that the dissemination algorithm uses for a better performance. This data can be the one already listed above for the beacons, or a list of forwarding neighbours, or a list of 1-hop neighbours, etc.

Additionally, in [370] Yi et al. proposed a different classification: heuristic based protocols and topology based protocols.

1. In **heuristic based protocols** the three first categories of Williams and Camp are grouped: probabilistic, counter based, distance based and location based algorithms.

2. **Topology based protocols** exploit topological information obtained mostly by the exchanged beacons. Within topology based methods, authors still differentiate between:

   - *neighbour topology based protocols* that make the forwarding decision in terms of the 1 or 2-hop neighbourhood information;
   - *source tree based protocols* construct a tree rooted at the source node in the network. Nodes forward the broadcast packets in terms of their status inside the tree;
   - *cluster based protocols* group nodes into clusters and elect a representative or clusterhead. The broadcast strategy usually depends on the node status.

In terms of the broadcasting process, it is possible to distinguish between *source dependent* and *source independent* approaches. In the former, the broadcasting strategy depends on source node, it decides the forwarding neighbours. However, in the source independent method, the forwarding decision is taken by the receiving node.

Although the already mentioned classifications reflect most of the techniques used in the existing works dealing with broadcasting, in this thesis we propose a new classification that reflects better the current state of the art. As ad hoc networks are energy constrained networks, one of the main goals of all the proposed protocols is to reduce the energy consumption. Literature reveals mainly two different approaches for reducing the energy expenditure when disseminating a message in the network: (1) trying to reduce the number of rebroadcasts and network resources, and (2) reducing the transmission range. So, our first classification relies on whether the transmission power used is fixed or adjustable. In both, the fixed and the variable transmission range approaches we could still differentiate whether the protocol uses an underlying topology or not. Algorithms that do not use any topology are divided into: (1) context oblivious, (2) context aware, and (3) neighbour knowledge. Additionally, for those relaying in topologies, three different possible types are considered: (1) connected dominating sets, (2) tree based topology, and (3) cluster based topology. An overview of the proposed taxonomy is shown in Figure 3.1.

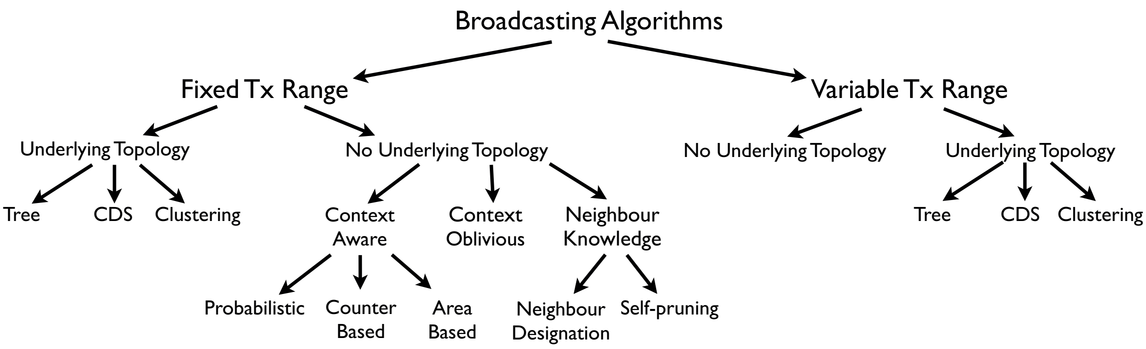


Figure 3.1: Taxonomy of the broadcasting algorithms

3.3 State of the art in broadcast algorithms

Next, we are reviewing the literature according to the already presented classification. It is possible to find some surveys on broadcasting algorithms in [115, 194, 224, 240, 318, 342, 376], however, they are not as through as the one included next. Unless explicitly mentioned, the presented approaches are deterministic and use local knowledge.

3.3.1 Fixed transmission range protocols

Most of the existing broadcasting approaches in the literature deal with energy consumption. Algorithms that consider only nodes were the transmission power used for communicating with any device is fixed, try to decrease the total energy used in the dissemination process by reducing the number of rebroadcast needed to flood the network, the packet overhead, etc. Next, we are presenting the most relevant algorithms that, using a fixed transmission range, try to make good use of the network resources for broadcasting.

3.3.1.1 Algorithms not using an underlying topology

When a topology exists in the network, the broadcast process takes advantage and uses this information for better disseminating the message. However, is not always possible or desired to create and maintain it. In those cases, for performing an efficient broadcast, nodes try to make a wise forwarding decision by obtaining some information of the current situation (context aware) or knowledge of their neighbours or neighbours' strategies (neighbour knowledge). Nevertheless, there also exist context oblivious approaches that base the forwarding decision only on probability. Next, we are reviewing the most relevant works in broadcast algorithms differentiating between the already mentioned types: context oblivious, context aware and neighbour knowledge.

Context oblivious

Nodes do not periodically exchange information with other nodes, or eavesdrop the channel in order to make intelligent forwarding. They simply forward the message with a predefined probability.

In the pure *probabilistic* approach, nodes do not consider the environment nor the current network situation. The forwarding decision is made according to a predefined probability. However, this probability can be set in terms of some behaviour of the node or devices around. Thus, we can find both context oblivious and context aware probabilistic approaches.

Flooding or *blind flooding* [249] as already mentioned, consists in retransmitting the message the first time it is received and ignoring any other copy of the message. It does not require any kind of information of the network or neighbour knowledge. The main drawback associated to this algorithm is the high number of collisions, redundancy, and the contention provoke in dense networks. Additionally, it is not aware of network partitions or temporary disconnections, i.e. the message only reaches nodes within a partition or connected component. This is a deterministic approach.

Probabilistic flooding [249] reduces the number of forwarding nodes in order to overcome the broadcast storm problem by assigning a probability to every node receiving a message. When the probability is 1, the algorithm behaves as *blind flooding*.

In order to deal with partitions and/or high mobility *hyper-flooding* was proposed [341]. The message is retransmitted whenever a new neighbour is met in order to increase reliability. However, in dense networks, the reliability could even decrease due to collisions and failures. Both, *blind flooding* and *hyper-flooding* are the only probabilistic approaches that are deterministic.

Context aware approaches

In the context aware approaches, the algorithm looks for external information about the current situation for making an intelligent forwarding decision. We differentiate between: (1) *probabilistic* that sets a forwarding probability, (2) *counter based* that is aware of the network traffic, and (3) *area based* that is aware of the density of the network.

1. **Probabilistic**

   In vehicular ad hoc networks, different protocols based on probabilistic schemes have been proposed. In [351], three distributed probabilistic and timer-based broadcast suppression techniques were presented: weighted p-persistence, slotted 1-persistence, and slotted p-persistence schemes. They rely on GPS information (or received signal strength when a vehicle cannot receive a GPS signal). Denoting the relative distance between the source and the receiver nodes and the average transmission range, the forwarding probability can be calculated assigning higher probability to nodes that are located further.

   Also based in a probabilistic scheme, [379] presents a *nth* power p-persistent broadcasting protocol for dense vehicular ad hoc networks. The basis are the same as the weighted p-persistence in [351], but the probability is elevated to the *nth* power. This proposal is proved to be efficient in very dense networks.

   Slavik proposed in [308] another probabilistic scheme that is anonymous and scalable. The main goal of this protocol is the privacy. The driver is not willing to adopt any technology that allows third parties to monitor their movements, so that no information between drivers can be exchanged. Two approaches are performed: (1) nodes are given a uniform and constant retransmission probability, and (2) the retransmission probability is dynamically determined based on the distance between the receiver node and the last hop.

   The *speed adaptive probabilistic flooding* algorithm (SAPF) was proposed in [245]. The rebroadcast probability is adaptively regulated based on the vehicle speed, to optimally reduce message delivery delays caused by increased contention, in areas with high density of vehicles. The goal of this work was to find a relation between the speed of the device and the forwarding probability that finds the optimal value for the latter in terms of the former.

   In the *gossip based broadcast* approach, the source node randomly selects k neighbours and send the broadcast message to them. Any node receiving the message for the first time does the same [182]. A survey on gossip protocols can be found in [203].

2. **Counter based**

The stochastic *counter based* approach [249] refrains the node from rebroadcasting in case multiple copies of the message have been received. Upon reception of the broadcast message, the node waits for a random time before rebroadcasting. If during this waiting time several copies of the same message are received, the rebroadcasting is cancelled. In this technique, the node is aware of the network traffic conditions and uses this information in the forwarding decision. All the *counter based* approaches reviewed next are stochastic.

An adaptive and probabilistic counter based approach where the threshold (the maximum number of repeated copies allowed) depends on the status of the neighbourhood is presented in [334].

The *color-based* broadcast algorithm is a variant of the counter based scheme [183]. It colours broadcast messages and all nodes rebroadcast the message after the random timeout unless it already heard η colours at that time.

Chen et al. proposed *DIS RAD* in [68]. This approach includes the concept of distance based into the counter based approach. Nodes closer to the limit transmission range have higher probability of rebroadcasting and shorter RAD. How nodes are aware of the distance to the source node is not specified.

An adaptive approach that uses two different threshold values, for sparse and dense networks was proposed in [24]. It compares the current number of neighbours to the average, and if it is lower the network is considered sparse, otherwise dense (how to calculate the average number of neighbours is not specified). The random delay and the counter threshold value are set accordingly.

An adaptive probabilistic counter scheme is presented in [217]. In *ProbA*, a node receiving a message sets random delay. During this waiting time the node counts the number of repeated copies received. The forwarding probability is set in terms of the number of copies. An extension, the *fuzzy logic probabilistic* (FLoP) is proposed in [218]. In it, the interval of the hello message is adaptive depending on the variability of the topology. This is estimated in terms of the difference in the number of neighbours between two consecutive *hello messages*.

3. **Area based approaches**

Two other stochastic techniques, that were introduced in [249], are the *distance based* (DB) and the *location based*. Both approaches start a random delay when the message is received for the first time, before rebroadcasting. In the former, nodes only resend the broadcast message depending on the distance to the source node. In the latter, the additional coverage that the retransmission will provide is estimated. In this case, GPS information is needed. Nodes are aware of the network density by estimating the distance to the 1-hop neighbours (or the additional expected coverage area). Next, we mention some of the most relevant work based on those approaches.

In Figure 3.2, the *distance based approach* mechanism is shown. Only nodes that are further than a predefine value (located in the forwarding area, i.e. grey zone) retransmit the message. In this figure, the nodes retransmitting the message sent by *Node A* are *Node E* and *Node F*.

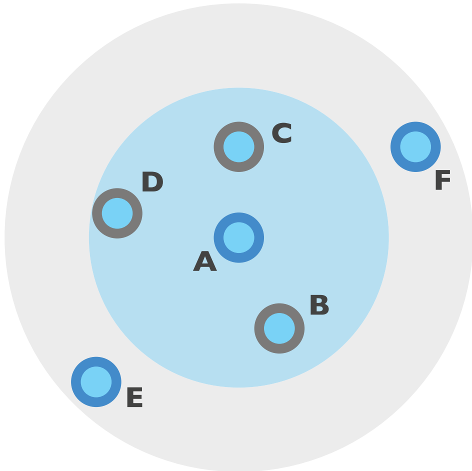


Figure 3.2: Distance based approach

Two adaptive versions of the DB were presented in [69], where nodes are sorted according to the received signal strength. The first one, *DAD-NUM* specifies a predefined number of forwarding nodes. The second one, *DAD-PER* specifies a percentage of nodes that will rebroadcast the message.

The difficulty to set up the thresholds is highlighted in [325]. Authors propose a broadcast algorithm that sets a defer time (timeout) before resending that is inversely proportional to the distance between the sender and the receiver node. Additionally, an angle based scheme is proposed for reducing the number of redundant transmissions.

The *area based beacon less algorithm* (ABBA) for 2D and 3D needs precise location information [253]. The co-ordinates of the sender are included in the header of the broadcast packet (assumes GPS). Nodes calculate the portion of the transmission range that is not covered, and set a timeout inversely proportional to this value. After the waiting time if some more copies were received and the transmission range is fully covered, the node cancels the retransmission. Two different approaches for the timeout are proposed; (1) random value between [0 1] seconds (stochastic approach), and (2) proportional to the already covered perimeter (deterministic). Similar beacon less approaches that include the location of the source node in the header of the message are the *optimised broadcast protocol* [100] and the *geoflood* [35]. The former divides the network in a honeycomb. The receiver calculates the distance to the 2 closest vertices of the honeycomb, and sets the timeout proportional to the minimum distance. The latter divides the transmission area in a cartesian plane. A timeout inversely proportional to the distance is set when receiving the message. As soon as a message is received from each of the four quadrants the retransmission is cancelled. Both approaches are deterministic. Similar approaches are the *six-shot broadcast* [114] and the *optimised flooding protocol* [255] that considers strategic forwarding positions.

The probabilistic and the distance scheme are combined in [60] to design a stochastic broadcasting algorithm. The broadcasting probability is set according to the local density of the network and the distance to the source node. However, no positioning information is used but only the 1-hop neighbours of the sender is included in the header of the broadcast message. Additionally, the neighbour elimination scheme is used, i.e. when no new neighbours are expected to be covered, the node cancels the retransmission.

In [59], an improved version of DB is presented adding counter based features to the border-aware scheme. The distance to neighbouring nodes is calculated using the using free space wireless propagation models. This stochastic approach also considers the remaining energy of the nodes. An adaptive location based algorithm that also uses 1-hop information was proposed in [334]. It is a probabilistic approach where the threshold of the area covered is not a fix value but it adapts to the neighbourhood size.

Da Li et al. proposed in [205], a distance-based directional broadcast for VANET using directional antennas. In the *efficient directional broadcast*, only the furthest receiver is responsible of forwarding the packet in the opposite direction where the packet arrives. Moreover, there are repeaters located at intersections to help disseminating the data. A probabilistic approach was also proposed.

In [154], an opportunistic relay scheme for cooperative collision warning in VANET that helps to mitigate the impact of transmission failure caused by shadow fading was presented. Devices periodically exchange motion information with their neighbours (location based). Each node not only calculates its own collision risk but also its neighbours', so that if a local vehicle has detected a danger and the neighbours keep quiet or reply incorrect responses, the vehicle may generate a relay packet that contains the motion information of the two vehicles involved in the danger and broadcast it.

In [113], a directional distance based broadcasting algorithm that uses percolation theory to select the direction of forwarding is presented. It uses directional antennas for estimating the direction of the source node and the additional coverage area. The forwarding probability depends on the value of this additional coverage area. Therefore, it is a stochastic model where no beacons are exchanged.

A distance based that takes into account the remaining battery level of devices is proposed in [177]. The *broadcasting method considering battery and distance* (BMBD) sets a timeout that is inversely proportional to the distance (uses GPS) and the battery level, i.e. nodes with higher remaining battery and higher distance from the source node set a lower delay. If a copy of the message is heard while waiting the retransmission is cancelled.

Three different stochastic approaches are presented in [188]. The first one, *distance based handshake gossiping* (DBHG) is a distance based protocol where the source node indicates the forwarding probability of neighbouring nodes in terms of their distance to itself. The second one *valency based handshake gossiping* (VBHG), is adding the nodes' degree knowledge to the first approach. Finally, the *average valency based handshake gossiping* considers also the experience for calculating the forwarding probability.

In [309], the *distance-to-mean broadcast* method is introduced. A node receiving a message for the first time sets a delay that is inversely proportional to the distance to the source node. When the timeout expires it calculates the spatial mean of all the neighbours it received the message from, and calculates its distance-to-mean. If that distance is higher than a predefined threshold, the message is retransmitted. It requires positional information.

A stateless distance and probability based approach that does not need to exchange *hello messages* is proposed in [44]. Nodes within 1-hop rebroadcast in terms of the distance to the source. The retransmission probability of nodes located h hops away is calculated and depends on the distance (hops) and network density.

A distance based algorithm *DibA* is proposed in [219], nodes do not exchange any *hello* messages. Moreover, the distance threshold varies according to the number of retransmissions heard of the same packet. It is a distance based that also considers the counter based approach.

	Neighbour info.	Hops knowledge	Determinism	GPS	Forwarding decision
Simple flooding [249] *Probabilistic flooding [249]* *Hyper-flooding [341]* *SAPF [245]* *[44]* *[113]* *DibA [219]* *[308]*	local	–	no	no	source independent
weighted/slotted 1/p-persistent [351] *Rewarn[154]* *Distance-to-mean [309]* *Defer time [325]*	local	1-hop	no	yes	source independent
gossip based broadcast [182] *nth power p-persistent[379]* *Adaptive counter [334]* *color-based [183]* *DIS RAD [68]* *[24]* *ProbA [217]* *fuzzy logic probabilistic [218]* *DBHG/VBHG/AVBHG [188]*	local	1-hop	no	no	source independent
DAD-NUM/PER [69] *BMBD [177]*	local	1-hop	yes	no	source independent
DAD-NUM/PER [69] *BMBD [177]*	local	1-hop	yes	no	source independent
optimised broadcast protocol [100] *geoflood [35]* *six-shot broadcast [114]* *optimised flooding protocol [255]*	local	–	yes	yes	source independent
[60] *[59]* *[334]*	local	1-hop	yes	yes	source independent
[113]	local	1-hop	yes/no	yes	source independent
ABBA [253]	local	–	yes/no	yes	source independent

Table 3.1: Classification of the context aware and context oblivious approaches

Neighbour knowledge approaches

By exchanging *hello messages*, nodes are aware of the neighbours within their transmission range. It is possibly to obtain 2-hop neighbours information by including in the beacon the list of 1-hop neighbours. The 2-hop neighbours knowledge gives more topology information, but at the same time, in highly mobile networks, it is more difficult to have accurate and up to date information. There are many works that use the neighbour knowledge for making an efficient forwarding decision.

Flooding based on 1-hop neighbours information and adaptive holding (FONIAH) was presented in [201]. This approach combines neighbour knowledge and area based flooding. When receiving the message the node can hold it for some time before rebroadcasting. This timeout is inversely proportional to the distance to the source node. The source node includes in the broadcast message its position and the distance to the furthest neighbour. If all 1-hop neighbours already received the broadcast message, the forwarding is cancelled.

Tonguz et al. present in [330, 331] a broadcasting algorithm that depends only on the local topology information (obtained using GPS), and that is robust against different types of vehicular traffic conditions. It handles both, the disconnected network and the broadcast storm problem [249] in a completely distributed fashion. *DV-Cast* specifies the region of interest (ROI) and disseminates the message towards that direction. The main criterion to determine how to handle the rebroadcast is based on the list of 1-hop neighbours and their relative distance.

1. **Self-pruning**

   flooding with self pruning is an approach in which nodes send the list of neighbours in the header of the broadcast packet. If additional neighbours are covered with the retransmitssion, the node schedules the rebroadcast in RAD seconds. In the meantime, it calculates the additional neighbours with the redundant copies. After RAD, the message is either sent or discarded [220].

   The stochastic *neighbour-coverage scheme* is proposed in [334]. It uses 2-hop neighbours information to know if there is at least one 1-hop neighbours that did not receive the broadcast message. When receiving the broadcast message for the first time, it creates a list with the 1-hop neighbours and sets a random delay. From this list all nodes that are supposed to received the message are eliminated. If all the 1-hop nodes did received the message, the retransmission is interrupted.

   The *scalable broadcast algorithm* (SBA) that uses 2-hop neighbours knowledge is presented in [256]. It is similar to *flooding with self pruning* but the information is included in the beacons not in the header of the packet. They also proposed a RAD that dynamically adjust to the network conditions, see Equation 3.1:

$$RAD : r \in [0, \alpha * \frac{max\_degree}{num\_neighbours}] \tag{3.1}$$

   Delayed flooding with cumulative neighbourhood (DFCN) [148], includes the 1-hop neighbours list in the packet header. The additional coverage in case of forwarding is calculated, and if it is higher than a predefined value, the message is scheduled to retransmit after RAD. Additionally, it adapts to the local network density: when the number of 1-hop neighbours is lower than a threshold, the predefine value is set to 1 (it rebroadcasts if there is at least one neighbour to cover). Moreover, if the node density is considered to be low, and therefore the forwarding probability 1, whenever a new neighbour is met, the RAD is stopped and the message is rebroadcast. Therefore, flooding with self pruning [220] can be considered a special case of DFCN.

In [184], a hybrid broadcasting algorithm that uses 2-hop information combines the neighbour designating scheme with self pruning. Source node selects at most one forwarding node from the set of 1-hop neighbours. At reception, if the node is not selected as relay, it creates a list with the neighbours, and remove from there the neighbours of the sender and the neighbours of the selected forwarding node. If the list is not empty, the message is retransmitted.

Total dominant pruning (TDP) and *partial dominant pruning* (PDP) are both proposed in [228]. Both uses 2-hop neighbours information. In the former, the senders includes in the broadcast packet the 2-hop neighbourhood, so that the receiver can consider all those nodes already covered. The later induces that information from the 2-hop neighbours knowledge, without piggybacking any information to the broadcast message.

The *lightweight and efficient network-wide broadcast* (LENWB) was proposed in [324]. It uses 2-hop neighbours information and the degree of the 1 and 2-hop neighbours that is included in the beacon. The source node does not select forwarding nodes, but nodes rebroadcast in terms of the knowledge of which of the 1 and 2-hop neighbours are expected to forward the message. Nodes have priority according to the degree, i.e., nodes with more neighbours have higher priority.

2. **Neighbour designating approaches**

   It is a source dependent technique, that is, the source node decides the next forwarding nodes from the 1-hop neighbours. Depending on the source node, the set of retransmitting nodes change. Different process for selecting the forwarding nodes have been proposed. Next, we will review the most relevant ones.

   Dominant pruning also requires 2-hop neighbours knowledge [220]. In the header of the packet a list of forwarding neighbours is included. The node that is included in this list has to calculate its forwarding nodes. Neighbours of the source node are considered already covered. For that the Greedy set cover algorithm is used. The nodes choose the 1-hop neighbour that covers the most 2-hop neighbours. This is repeated until all 2-hop neighbours are covered.

   In *multipoint relay* (MPR) [270], the source node chooses the next forwarding nodes. Unlike dominant pruning, the selected relaying nodes are included in the beacons exchanged not in the header. The selecting mechanism works as follows: first, check the 2-hop neighbours that can only be reached by one 1-hop neighbour and select them as MPRs. Then, from the remaining 1-hop neighbours select those that cover the most 2-hop neighbours that are not covered yet. Continue until all 2-hop neighbours are covered. For a graphical explanation please see Figure 3.3. Only the blue nodes are in charge of retransmitting the broadcast messages.

   The well known routing protocol for ad hoc networks Optimized Link State Routing Protocol (OLSR) is based on this. For a more detailed explanation please refer to the internet draft RFC 3626 [77]. A good survey on multipoint relay and connected dominating sets approaches for broadcasting is presented in [214].

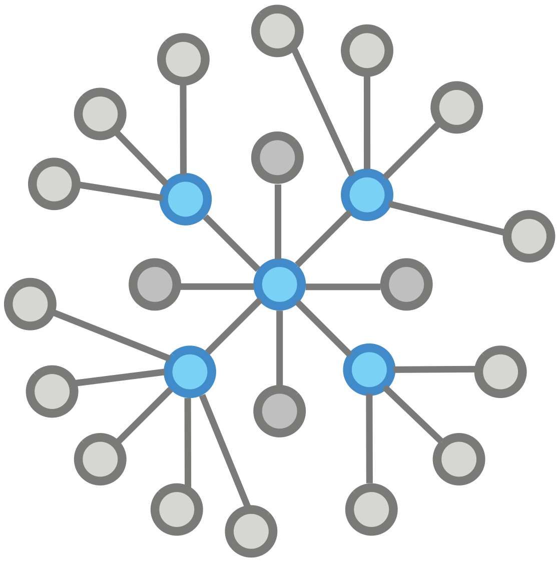


Figure 3.3: Multipoint relay mechanism

The *ad hoc broadcast protocol* (AHBP) is presented in [258]. Similarly to MPR, source nodes select the forwarding nodes. In this case, the forwarding nodes are called *broadcast relay gateway* (BRG). The mechanism for selecting the BGRs is identical to MPR. The differences are: (1) nodes are informed about becoming a BGR in the header of the packet, not using the beacons. (2) 2-hop neighbours knowledge is used to determine which neighbours already received the packet and consider them as already covered. An extension of the protocol for considering mobility is also proposed, AHBP-EX. If a node receives a message from another but did not previously exchanged beacons (i.e. due to the mobility), it will assume it is a BGR.

Total dominant pruning (TDP) and *partial dominant pruning* (PDP) are both proposed in [228]. Both uses 2-hop neighbours information. In the former, the senders includes in the broadcast packet the 2-hop neighbourhood, so that the receiver can consider all those nodes already covered. The later induces that information from the 2-hop neighbours knowledge, without piggybacking any information to the broadcast message.

There are plenty of broadcasting algorithms based on MPR but that modify the MPR selection procedure in order to reduce the number of collisions, the cardinality of the MPR or just to efficiently manage the power consumption. We will just briefly mentioned some of them. In [233], authors propose different models like the *in-degree greedy set cover*. It first checks the 2-hop neighbours that can only be reached by one 1-hop neighbours and selects them as MPRs. Then, randomly picks up a node from the uncovered 2-hop nodes, among all 1-hop neighbours nodes that can cover this 2-hop node, and have not been selected as MPRs by the source node S, selects a node as an MPR that has minimum number of uncovered 2-hop neighbours. Continue until all 2-hop neighbours are covered. Authors also proposed, *MPR Selection with Minimum Overlapping*, for reducing the traffic and the risk of collisions. As the others approaches, nodes that can only be reached by one 1-hop neighbours are selected first. And then, the node with the minimum covering ratio is chosen as an MPR among the 1-hop neighbours that are not selected as MPRs yet. This is done until all 2-hop neighbours are covered. The covering ratio is defined as the ratio of the covered 2-hop neighbours over uncovered 2-hop neighbours of a node. Moreover, it was also presented the *weighted set cover* that gives weights to nodes according to a desired property, e.g. bandwidth, and selects the 1-hop neighbours with the largest number of uncovered nodes over the weight ratio. The *MPR Selection with secondary priority* that gives priority to nodes with maximum number of neighbours is also proposed. This is applied OLSR, in case two nodes has the same number of uncovered neighbours.

In [222], authors selects the MPR nodes in terms of the *forwarding utility* of the 1-hop neighbours. This *forwarding utility* is computed as a function of the power utility of each node (remaining battery), and the forwarding utility (the ratio of uncovered 2-hop neighbours over all the 2-hop nodes that a 1-hop neighbours node covers). As the previous approaches, the algorithm first checks the 2-hop neighbours that can only be reached by one 1-hop neighbours and select them as MPRs. Then selects as MPR the node with highest *forwarding utility* value. Continue until all 2-hop neighbours are covered. Later in [223], the first steps of the MRP approach were eliminated and MPR nodes are only selected in terms of the *forwarding utility* value.

Additional MPR nodes are chosen in order to provide reliable broadcast in [22, 77]. The redundancy method proposed in [22] chooses additional MPR to cover only *selected* 2-hop nodes. In [77], the redundant MPR cover all the subset of 2-hop neighbours, reducing therefore, the cardinality of MPR while at the same time increasing the delivery ratio. The idea is that the 2-hop MPR nodes must be covered at least *m-times*.

The *double-covered broadcast* (DCB) algorithm [229] is a source dependent protocol where the source node chooses the set of 1-hop neighbours that cover all 2-hop neighbours and the non selected 1-hop neighbours at least twice. The retransmissions of the forwarding nodes are considered as acknowledgements, thus if the source does not detect all the forwarding nodes retransmissions it will resend the packet.

A vehicular multi-hop broadcasting protocol (VMP) for fast dissemination on the opposite direction was introduced in [43]. VMP designates multiple forwarding nodes (i.e. it is source dependent) with different delays to disseminate an alert message in a specific area. To ensure high reachability a cooperative forwarding mechanism is used and also, a duplicated packet detection procedure is provided to discard unnecessary rebroadcasts.

3.3.1.2 Algorithms using an underlying topology

Literature reveals many works using topological information to give a structure to the unstructured and highly changing topology of mobile ad hoc networks. In order to collect this data, it is necessary to periodically exchange *beacons* as there is no pre-existing infrastructure in MANETs. As already mentioned, within this category, we differentiate between *connected dominating sets, tree based* and *cluster based* topologies.

	Neighbour info.	Hops knowledge	Determinism	GPS	Forwarding decision
FONIAH [201]	local	1-hop	no	yes	source independent
DV-CAST [331]	local	1-hop	no	yes	source independent
VMP [43]	local	1-hop	yes	yes	source dependent
flooding with self pruning [220] *DFCN[148]*	local	1-hop	no	no	source independent
neighbour-coverage scheme [334] *SBA [256]*	local	2-hop	no	no	source independent
TDP/PDP[228] *LENWB[324]*	local	2-hop	yes	no	source independent
[233]	local	2-hop	no	no	source dependent
Dominant pruning[220] *multipoint relay [270]* *ad hoc broadcast protocol [258]* *[222]* *[223]* *[22]* *[77]* *double-covered broadcast [229]*	local	2-hop	yes	no	source dependent
[184]	local	2-hop	yes	no	source independent/source dependent

Table 3.2: Classification of the neighbour knowledge approaches

Connected dominating set

Connected dominating set (CDS) is a source independent technique that builds a sub network, also called virtual backbone, for covering all the nodes in the network. Nodes either belong to the backbone or are adjacent to at least one of them. An extensive review can be found in [372]. In graph theory, a connected dominating set is defined as follows:

Definition 5 (Connected dominating sets).
A connected dominating set for $G = (V, E)$ *is a subset* $V' \subseteq V$ *such that for all* $u \in V - V'$ *there is at least a* $v \in V'$ *for which* $(u, v) \in E$ *and the sub-graph induced by* V'*,* $G[V']$ *has only one connected component.*

A connected dominating set contains a group of connected nodes that covers all the network forming a virtual backbone. Efficient broadcasting can be achieved by identifying a connected dominating set, preferably small, and allowing only the nodes in the set to relay the broadcast message. Peng and Lu propose an efficient CDS-based broadcasting algorithm in [257]. The algorithm uses 2-hop knowledge and includes the information about the dominant nodes (or nodes belonging to the backbone) in the message. An important difference with AHBP is that not only the nodes covered by the source node are not considered for belonging to the CDS anymore, but also the nodes covered by the other dominant nodes that were selected before.

Wu and Li proposed in [357] a marking process for efficiently and quickly determining a CDS. This approach was first proposed for undirected graphs using the notion of dominating set only and was later extended to cover directed graphs by introducing another notion called absorbent set [356]. Specifically, a node becomes a gateway if it has two unconnected neighbours.

Wu et al. proposed in [361] a broadcasting and routing algorithm based on connected dominating sets based on dynamic selection of the dominating nodes. The algorithm alternates the nodes belonging to the CDS when possible because they consume more energy. Additionally, nodes with higher remaining battery level are preferred.

In [362], Wu and Lou use 3-hop neighbours knowledge in order to have complete 2-hop information for obtaining a relatively stable CDS. Each node chooses a pair of nodes, a 1-hop neighbours and a 2-hop neighbours to cover all its 2-hop neighbours.

The *Rule k* was proposed in [85] to reduce the overhead for creating connected dominating sets. It is fully localised and supports unidirectional links. This protocol was modified in [318] to eliminate the messages exchanged needed to create the CDS.

Stojmenović et al. proposed a broadcasting algorithm based on dominating sets and neighbour elimination schemes using 2-hop knowledge [320]. Only nodes belonging to the CDS are able to rebroadcast the message. Upon reception, nodes in the CDS set a delay inversely proportional to the number of uncovered neighbours. During the waiting period, if copies of the message are heard, all the neighbours that received the message are also removed from the forwarding list. If there is no neighbour to cover, the neighbour elimination rule cancels the retransmission.

In order to improve reliability and solve disconnection problems, in [321] authors proposed a broadcasting algorithm that works like in [320], but that all nodes can rebroadcast a packet. Nodes not included in the CDS set a longer waiting time, and additionally, every time a new neighbour is met, the broadcast message is sent.

Recently, a general framework for broadcasting that seamlessly (without using any parameter) adjust itself to any mobility scenario was introduced in [317]. It is built over several recent algorithms using 2-hop topological or 1-hop positional knowledge. When the message is received the first time, nodes set a timeout and also generate 2 lists: R with the nodes believed to have received the packet, and N with the nodes that did not get the message yet. Nodes belonging to the CDS have shorter timeouts than those not in the CDS. Once the waiting time is finished, the node retransmits only if N is not empty, and updates both R and N. After receiving each hello message, nodes reevaluates the status in the CDS and also update N. If N was empty and a new neighbour not included in R is detected the timeout is reactivated.

Literature reveals yet different decentralised CDS that use local knowledge in dynamic ad hoc networks. Two proposals designed to create k-vertex connected m-vertex dominating set virtual backbones in an asynchronous and computer effective fashion are presented in [204, 295]. For a more detailed explanation on CDS please refer to [293].

Tree based topology

All the already mentioned approaches have to store the ID of the broadcast message in order to just rebroadcast the first reception of the message. The most straight forward method for avoiding this without storing anything is to forward the broadcast message on an acyclic subgraph.

A network can be modelled as a graph $G=(V,E)$, where nodes are represented by the set of vertices V and the links are the set of edges E. It is possible to define an underlying structure in the network, as done with the CDS and the clustering, that forms a tree. In order to disseminate a message to all nodes in the network, we can consider the spanning tree structure where only nodes belonging to the tree that are not leaves rebroadcast the message. Next, we define both concepts: tree and spanning tree.

Definition 6 (Tree topology).
A tree is an undirected and connected graph that has no cycles.

Definition 7 (Spanning tree)**.**
A spanning tree of a graph is an undirected and connected subgraph that has no cycles and contains all nodes. It can be defined as the maximal set of edges of the graph that contains no cycle, or as a minimal set of edges that connect all vertices of the graph. The minimum spanning tree (MST) is the minimal set of edges that connect all vertices of the graph, or in case the graph is weighted, the minimum sum of the weight of all branches.

In case the network is partitioned, we can talk about *spanning forest*, where a spanning tree is formed in every connected component, see Figure 3.6.

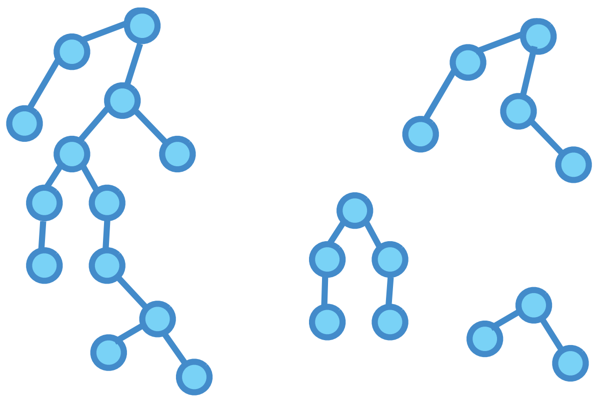


Figure 3.4: Illustration of a tree and a forest

The maximum leaf spanning tree problem is equivalent to finding the minimum connected dominating set [110]. In tree topology edges are chosen and play an important role, while in CDS are nodes. However, the construction of the spanning tree from a CDS is quite straight forward. In Figure 3.5, we can see a graphical representation.

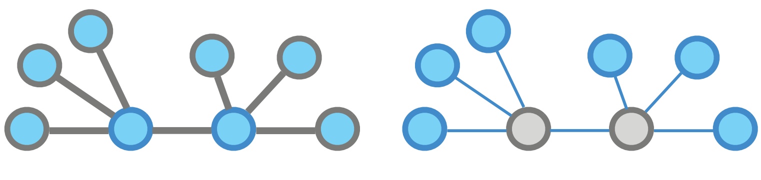


Figure 3.5: Illustration of a minimum spanning tree and and minimum CDS

Using a tree topology has been widely used in telecommunication networks [3], but most of these algorithms need a stable network, that is not available in MANETs. Therefore, specific algorithms for mobile ad hoc networks must be designed.

Creating and maintaining a tree topology in mobile ad hoc networks is challenging as nodes have knowledge only about direct neighbours, the link stability is very changeable and the data structure and maintenance of the tree must be minimal. It is possible to differentiate between source dependent and shared trees. In a *source dependent tree*, the source node is the root of the tree and for each pair of source-multicast group a different tree is built. In the *shared tree* only one tree is built over the network and the source node just needs to be able to send the data to the root of the tree or to a member.

In [272], authors proposed a distributed spanning tree to forward data packets. However, some operations require centralised designs as for merging two spanning trees, the root node is deciding which nodes are merging.

Using the overlay of a gossip protocol, the *push-lazy-push multicast tree* (Plumtree) is constructed in [202]. The links from which a node receives the messages are considered to form the spanning tree, and when a message is received twice, the node prunes this link. However, for repairing the tree, every time a node receives a message it sends an *IHAVE message* to the links that do not belong to the spanning tree. Whenever a node requests this message, the link between these two nodes becomes part of the spanning tree.

A fully distributed and decentralised method, *TreeCast* is proposed in [170]. It is not only capable of dealing with mobility of nodes, but also forces nodes with high mobility to be leaves, thus, the backbone tree is composed of more static nodes.

MaxCST [107] is a deterministic and self stabilising algorithm that builds a cluster in the network with diameter of 2 at most and then, construct a spanning tree on the network.

When considering decentralised systems the construction of the tree is not trivial, most of the proposals deal with static or low mobility, as nodes usually compute the local tree and then merge adjacent trees (e.g. the LMST [212]). However, the *dynamicity aware graph relabeling system* [65] is a high level abstraction model for creating and maintaining tree topologies efficiently. It is based on local rules that are able to cope with topological changes. Also in [67], a decentralised algorithm that is able to cope with the creation and maintenance of distributed spanning trees in highly dynamic networks is presented. Both, the merging and splitting process of two trees are purely localised.

There are also many works trying not only to efficiently build the topology, but also to merge nodes in a tree that have some specific features. For example, in [263], nodes are selected for joining the tree in terms of their level of trust in the network, creating thus, more reliable topologies. For a more in detail overview of kind of tree based topologies, please refer to [262].

Clustering based topology

A clustered network is a 2-level hierarchical network that divides nodes into groups and elects only one node as representative of the group, the clusterhead (CH). The CH directly connects to all other members of the cluster. Nodes belong only to one CH, but sometimes can hear more than two CH. Those nodes declare themselves *gateway*.

The set of clusterheads forms a dominating set, but not a connected dominating set. In order to have a CDS, clusterheads must elect as forwarding nodes the *gateway* for connecting two neighbouring CHs.

The main goal of both CDS and trees is to cover the whole network using the minimum number of nodes in the CDS or obtaining the maximum number of leaves. However, in clustering the main goal is to group nodes in terms of some common feature or joint goal.

There are mainly two different approaches in clustering: (1) *active clustering*, where nodes cooperate to elect the CH by exchanging info periodically, and (2) *passive clustering* where

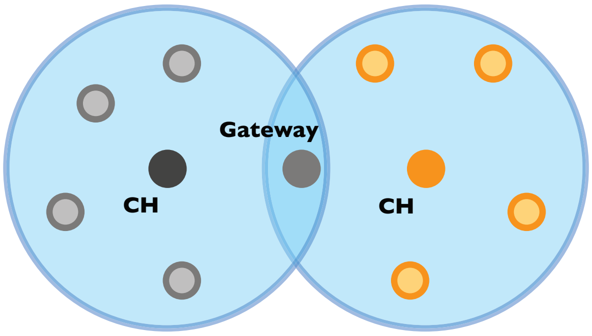


Figure 3.6: Illustration of two adjacent clusters with a gateway

the CH election starts with the on going traffic. Information is propagated in the packets and collected through eavesdropping, so there is no overhead or setup.

In the *Lowest ID algorithm* (LID) clustering algorithm [102], initially all nodes are coloured white. A white node that finds itself with the lowest ID among its non cluster-head neighbours chooses itself as CH and turns black. And all white neighbours join this CH and turn themselves grey. The process finishes when there are no more white nodes. The *Highest degree algorithm* (HD) [120], considers first the node degree in the clustered decision. In case of tie, the lowest ID node is selected. A survey on different clustering techniques can be found in [373]. Next, we present some of the most relevant broadcasting algorithms based on cluster approaches.

In [121], a broadcasting algorithm based on passive clustering is proposed. Nodes decide to become either CH, ordinary nodes or gateway in terms of some predefined thresholds. Both CH and gateway rebroadcast the message.

Not only location information but also the battery power level is used in [208] for electing the clusterheads. The proposed *vote-based clustering* considers nodes with high degree and battery level better candidates for becoming CHs. However, this results in frequently changing topology.

In order to deal with the frequently CH changes and thus, not very stable structure, the stability of the link is also considered in the *node and link weighted clustering algorithm* (NLWCA) [32]. It uses the node state (device power and signal strength) and the link stability for electing CHs. For broadcasting, clusterheads disseminate the message within its own cluster but also send it directly to all stable neighbouring clusterheads by unicast [33].

Wu and Lou proposed in [358] a clustered network where each clusterhead selects the forwarding nodes inside the cluster so that all clusterheads in the vicinity are covered. Information about the clusterheads that will receive the message is included in the broadcast message so that CHs can deduce the coverage area by the pruning technique. Instead of the 3-hop covered area, in this paper a novel idea of the 2.5-hop coverage is introduced where each clusterhead just covers the CHs that have members within 2 hops.

3.3.2 Variable transmission range protocols

As we mentioned before, the energy consumption in mobile ad hoc networks is a hot topic, since devices can run out of battery provoking the network degradation. Two different ap-

		Neighbour info.	Hops knowledge	Determinism	GPS	Forwarding decision
CDS	[362]	local	3-hop	yes	no	structure
	all the rest	local	2-hop	yes	no	structure
Trees	[272]	local/global	1-hop	yes	no	structure
	TreeCast [170]	local	2-hop	yes	no	structure
	PlumTree [202]	local	1-hop	no	no	structure
	all the rest	local	1-hop	yes	no	structure
Clustering	[121]	local	–	yes	no	structure
	vote-based clustering [208]	local	1-hop	yes	yes	structure
	NLWCA [32]	local	1-hop	yes	no	structure
	Lowest ID algorithm [102] *Highest degree algorithm* [120]	local	2-hop	yes	no	structure
	[358]	local	2.5-hop	yes	no	structure

Table 3.3: Classification of the approaches using underlying topology

proaches are considered for reducing the energy consumption in ad hoc networks. The first one deals with reducing the number of retransmissions of a message, so that the overall energy consumption is minimised. All the above reviewed papers focus on that. The second approach considers reducing the transmission power so that the overall energy consumption is reduced. The first one can be seen as the *fix power approach* that was already reviewed, and the second family is the *variable power* approach that we are reviewing next.

In the *variable power* approach each node can transmit using different transmission radii, therefore, the number of neighbouring nodes reached when broadcasting a message varies according to the transmission power used. There are several works that try to find a common power level that guarantees a low node degree [179], or that the communication graph is connected with at least k-neighbours over a uniformly distributed network [54]. Every node has to communicate with each other for selecting the common transmission power, provoking overhead. Such approaches are not scalable as the overhead increases with the size of the network. However, in [124], it was shown that a variable transmission range can outperform a common transmission range approach in terms of power saving, increasing as well the capacity of the network. They also claim that there is an optimum setting for the transmission range, not necessarily minimum, which maximises the capacity available of the nodes in presence of mobility. There is a good survey on conserving power by employing the transmission power control in [134].

When considering the variable transmission power of nodes, adjusting each node with the optimal transmission radius is one of the main problems. Both the *minimum range assignment* and the *minimum energy broadcast problem* deal with that.

Definition 8 (THE MINIMUM h RANGE ASSIGNMENT PROBLEM).
The minimum range assignment problem consist of finding the transmission power of the a given set of nodes or stations so that the total power consumption is minimised whilst guaranteeing the communication between them in h *hops [78].*

Definition 9 (THE MINIMUM ENERGY BROADCAST PROBLEM).

Considering a source node wants to broadcast a message to all nodes in the network, finding the set of relaying nodes and their transmission ranges so that the total energy consumed is minimised is called the minimum-energy broadcast problem [339].

Both, the *minimum assignment problem* and the *minimum energy broadcast problem* are NP-hard problems. It was demonstrated in [78] and in [339], respectively. The main difference between them, is that in the former, the source of the broadcasting algorithm is not considered while in the latter, it is. It can be seen as the previously explained concept of source independent and source dependent algorithms.

3.3.2.1 Algorithms not using an underlying topology

Cartigny et al. presented the *RNG broadcast oriented protocol* in [61]. It runs locally at each node but it requires the distance to all neighbouring nodes and distances between its neighbouring nodes. The RNG graph stands for the relative neighbourhood graph. It is an undirected graph where an edge between two points is added if it does not exist a node closer to both nodes (see Figure 3.7, where there can not be any neighbour in the striped area in other to have two RNG neighbours). A node receiving a broadcast message from a RNG neighbours retransmits it using the minimum transmission power needed to reach the furthest RNG neighbour that did not received it yet. Otherwise, it starts a timeout and creates a list with all the RNG neighbours that did not receive it. If after the waiting time the list is not empty, it rebroadcasts the message with the transmission power enough to reach the furthest RNG neighbour in the list. A different version where the timeout is set every time a message is received is proposed in [63].

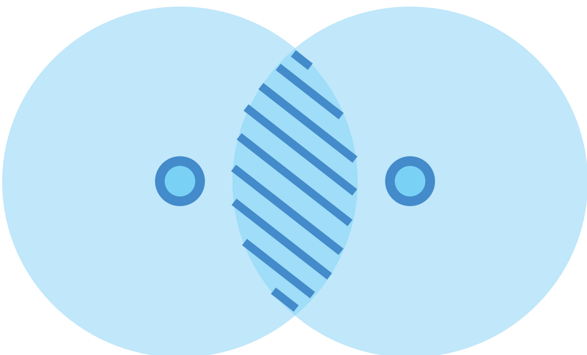


Figure 3.7: Relative neighbourhood graph

The notion of forbidden set in the relative neighbourhood graph is incorporated in [344]. This forbidden set prevents nodes with low remaining battery to act as forwarders in the dissemination process. Additionally, when redundant retransmissions are received the node will change the relative neighbour or will ask to be covered by only one.

Faloutsos et al. proposed the *power adaptive broadcasting with local information* (PABLO) in [70]. Nodes exchange information in the beacons in order to know the transmission power needed to reach the 2-hop neighbours. The source node examines whether it is worth or not to exclude the furthest node from the 1-hop neighbourhood and reduce the transmission range to reach the new furthest neighbour (during the random back-off time). See Figure 3.8 where *node A* excludes *node C* from the 1-hop neighbourhood if the sum of the power *node A* needs to reach *node B* plus the power *node B* needs to reach *node C* is lower than the power *node A* needs to directly reach *node C*.

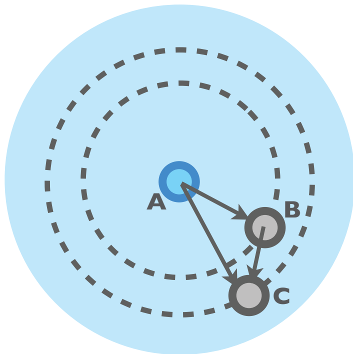


Figure 3.8: Mechanism of PABLO

In [175], an extension to PABLO is proposed that applies neighbourhood pruning, i.e. after performing the optimisation the algorithm considers the possibility of excluding the furthest node(s).

An approach to estimate the local density using an analytical model is used in [213] to set the transmission range according to this estimation.

The *inside-out power adaptive approach* (INOP) is presented in [74]. It uses 2-hop neighbours knowledge information to obtain a good energy utilisation for covering all direct neighbours. The difference with other existing approaches is that each node sorts the neighbours in terms of the power needed to reach them, and then, computes the optimal energy strategy starting from the closest neighbour to reach the next neighbour directly or indirectly. Two different versions for selecting the relay nodes are proposed: (1) *INOP with self pruning* that set a random timeout for an interval that is inversely proportional to the number of uncovered neighbours. If after the delay the number of uncovered neighbours is empty the rebroadcast is cancelled. And (2) *INOP with neighbor designation* considers the possibility of increasing the transmission power of an already selected relay node instead of adding a new relay for covering an uncovered neighbour. Considering the timeout is random, this approach is stochastic.

In [354], different versions of *dominant pruning, total dominant pruning, partial dominant pruning, PABLO and INOP* using different termination conditions and variable transmission power are compared.

In vehicular ad hoc networks it is also a tendency to adjust the transmission range used. In [277], nodes exchange periodically beacons containing information about the path loss. Neighbours are sorted according to the average path loss and when a broadcast message is received, the node checks the transmission power necessary to reach a targeted number of nodes.

In [254], the *efficient reliable 1-hop broadcasting* (EROB) algorithm is introduced. It considers simultaneous transmission over three different channels. Two control channels and another for data packets. It proposes the use of a control packet called BIP (broadcast in progress) to avoid the *hidden terminal problem* while broadcasting. Additionally, nodes can use different transmission levels for enhancing the network throughput and the network lifetime, as well as reducing the power consumption and the number of collisions.

In *RandomCast* [221], the algorithm uses the *power saving mechanism* of 802.11 PSM [31] that allows the device to be in a low-power sleep state if it is not addressed in the data transmission. However, nodes in *RandomCast* consistently operate in the PS mode. The transmitter can specify the level of overhearing for unicast packets. Additionally, the probability of rebroadcasting is also is based on number of 1 and 2-hop neighbours.

	Neighbour info.	Hops knowledge	Determinism	GPS	Forwarding decision
RNG broadcast oriented protocol [61] [63] [344]	local	2-hop	yes	yes	source independent
PABLO [70]	local	2-hop	no	no	source independent
INOP [74]	local	2-hop	no	no	source independent/source dependent
[277]	local	1-hop	yes	no	source independent
EROB [254]	local	1-hop	no	yes	source independent
RandomCast [221]	local	1-hop	no	no	source independent

Table 3.4: Classification of the variable transmission range approaches with no underlying topology

3.3.2.2 Algorithms using an underlying topology

A solution of the *minimum energy broadcast problem* is the *minimum energy broadcast tree*. Different approaches have been proposed in the past years.

Tree based topology

In order to solve the *minimum energy broadcast problem*, the well known *broadcast incremental power* (BIP) was presented in [348]. BIP constructs a tree at the source node and calculates the node not belonging to the tree that is reached using the minimum power. That will be the next node included in the tree. The *sweep* procedure is also proposed in the same work for further reducing the total energy consumption. BIP requires global information.

BAIP, the *average broadcast incremental power* [343] is a variant of BIP that considers the average incremental cost for selecting the new node that is included in the tree. This value is the ratio of the minimum additional power increased by a node in the current tree to reach some new nodes to the number of these new nodes.

The *hop constrained minimum broadcast incremental power* (HC-BIP) is proposed in [58]. It ensures that all nodes in the network receive the broadcast message in less than k hops. It is a globalised implementation.

The *iterative maximum-branch minimization* (IMBM) algorithm was presented in [207]. It focuses on the construction of the minimum energy broadcast tree routed at the source node that reaches all the destinations. In a initial step, the source builds a *basic broadcast tree* to all destinations. Then, it tries to reduce the required power by replacing the maximum branch by the less power alternative. Liang proposes in [215] an approximation algorithm that uses global knowledge and is based on Steiner trees.

In [339], the *embedded wireless multicast advantage* (EWMA) algorithm uses the minimum spanning tree (MST) as the initial feasible solution and then, it is improved by exchanging some existing branches in the initial tree for new branches so that the total energy necessary to maintain the broadcast tree is lower. Two EWMA approaches are proposed one centralised and another distributed. The distributed approach, however, requires information from multiple hops away, so it might not work properly in frequently changing topology.

Wieselthier proposed in [349] two distributed versions of the well known BIP: *distributed-BIP-all Dist-BIP-A*, and *distributed-BIP-gateways Dist-BIP-G*. Both use 2-hop neighbours information for constructing trees. In the former, only nodes that can be reached directly are included in the tree. Each node builds a local tree that are then pieced in a global tree. In the latter, nodes that can be reached using a relay are considered gateways. Only, those gateways are in charge of building local trees. Once the global tree is formed, the *sweep* operation is applied. This operation is centralised.

The *local minimum spanning tree* (LMST) [211, 212] needs local positioning exchange in the beacons to build the neighbourhood graph. Each node apply the Prim's algorithm in order to find the local minimum spanning tree. Once, the tree is constructed it reduces the transmission power to reach the furthest neighbour. The *directed LMST broadcast oriented protocol* (LBOT) [62] is based on LMST but uses directional antennas and lies on the LMST topology. The source node sends the message and using the neighbour elimination (or self pruning rule) on the LMST tree, the message is propagated.

In [239], the *local tree-based reliable topology* (LTRT) motivated from LMST is proposed. It ensures k-edge connectivity of the network in order to get reliable communications.

The *broadcast on local minimum spanning tree* (BLMST) is proposed in [210]. It first constructs an underlying topology using LMST, and then, the source node broadcasts the message and a node receiving the message from all its neighbours cancels the forwarding. In this work, there is also an analytical study that indicates under some circumstances it is more efficient to use lower transmission power in a multi-hop fashion than using longer transmission ranges.

The redundant radius scheme is introduced in [364] where two different transmission radius are used. First, a smaller range is considered for building the broadcast tree in terms of the neighbourhood, and then, a longer radius is used for the actual transmission.

In [159], the optimal transmission range that considers both the number of relays and the energy consumption is calculated. Moreover, two broadcasting algorithms are proposed the *target radius LMST broadcast oriented protocol* (TR-LBOP) and the *target radius and dominate set based protocol*. The former considers the neighbour elimination to reduce the subset of direct neighbours and reduces the radius to preserve connectivity, increasing it up when possible. The latter computes a CDS trying to choose as relays nodes as close as possible to the target radius.

Connected dominating sets

A broadcasting algorithm that uses variable transmission range based on connected dominating set is presented in [360]. As nodes belonging to the CDS consume more energy, they alternate when possible. Additionally, the transmission power of each node is reduced during the broadcast process without sacrificing the reachability.

Some distance rules where nodes transmission power might be adjusted based on the distance are proposed in [318]. Moreover, nodes with less distance to their neighbours have higher preference for becoming a node belonging to the CDS.

In [206], two decentralised approaches are proposed for constructing the minimum total communication power connected dominating set. The goal is to find the CDS that minimises the sum of the transmission power of nodes in the CDS.

	Neighbour info.	Hops knowledge	Determinism	GPS	Forwarding decision
broadcast incremental power [348] *average broadcast incremental power* [343] *HC-BIP* [58] *iterative maximum-branch minimization* [207] [215] *embedded wireless multicast advantage* [339]	local	global	yes	no	structure
Dist-BIP-A/Dist-BIP-G [349]	local	2-hop/global	yes	no	structure
local minimum spanning tree [211] *LBOT* [62] *LTRT* [239] *BLMST* [210]	local	1-hop	yes	yes	structure
[364]	local	1-hop	yes	no	structure
TR-LBOP [159]	local	1-hop	no	yes	structure
[360] [318] [206]	local	1-hop	yes	no	structure
[250] [248] [21]	local	2-hop	yes	no	structure

Table 3.5: Classification of the variable transmission range approaches with underlying topology

Clustering topology

An autonomous cluster scheme is proposed in [250]. It adapts the transmission power of every node in terms of the distance between the node and the neighbouring nodes to have a specific number of neihbours.

In [248], the node residual energy, the nearby topology, the relative location and the relative mobility are used for electing the CH. Additionally, cluster members are able to estimate the distance to the CH and reduce the transmission power accordingly. When the residual energy of the CH is lower than a predefined threshold, the re-clustering operation is triggered.

In [21], 2-hop neighbourhood information is used in a distributed clustering algorithm to divide the network, where clusters may overlap. Then, a distributed *sweep* operation is used for finding nodes whose transmission power can be decreased, whilst still guaranteeing every node belongs to at least one cluster. Finally, it runs the DMST [156] algorithm for constructing the directed minimum spanning tree to join the clusters.

3.4 Metaheuristics

Technology in networking is evolving faster than information systems; tiny devices are already provided with communication capabilities. But the existing communication systems are not appropriate or efficient for such heterogeneous networks: self-organisation mechanisms able to handle heterogeneity, dynamic nature, resource constraints, scalability, or failures are needed.

It is possible to find similarities when analysing biological systems: self-organisation, recovering from failures, collaborative behaviour, minimisation resources, finding stability, etc. Most of these systems achieved this behaviour after evolving for millions years. Many researchers are developing algorithms inspired by Nature in order to efficiently tackle different problems. For example, they have been widely applied for networks design and optimisation in the literature [97].

There exists a large number of real-life problems that are complex and difficult to solve. Exact algorithms are not appropriate or take too long for using them. Therefore, approximate algorithms are needed. Metaheuristics [119] are iterative stochastic optimisation tools that are able to provide good solutions in reasonable time for highly complex optimisation problems. Generally, metaheuristics make no assumptions about the problem to solve, so they are generic tools that only need an adequacy (or fitness) function to guide the search towards better solutions. Most of metaheuristcs mimic natural process for solving these complex optimisation problems (e.g. ant or bee colonies, evolution of species, swarms, etc.).

In [328], approximate algorithms are decomposed in two subclasses: specific heuristics and metaheuristics. According to the *Oxford dictionary*, a heuristic can be defined as:

Definition 10 (Heuristic).
Proceeding to a solution by trial and error or by rules that are only loosely defined.

The term heuristic comes from old Greek and means *the art of discovering new strategies (rules) to solve problems* [328]. The suffix *meta* means *upper level methodology*. A formal definition of the term metaheuristic was proposed by Talbi in [328]:

Definition 11 (Metaheuristic).

Metaheuristic search methods are upper level general methodologies (templates) that can be used as guiding strategies in designing underlying heuristics to solve specific optimisation problems.

Specific heuristics are designed for a particular problem, thus, they are problem dependant. Metaheuristics are more general approximate algorithms that can be applied to a large number of optimisation problems. A optimisation problem is defined in [26] as follows:

Definition 12 (Optimisation Problem).

An optimisation problem is formalised after a pair (S, f), where S represents the search space of the problem, while f is a quality criterion known as the objective function defined as:

$$f : S \rightarrow \mathbb{R} \tag{3.2}$$

In this case where a single criterion f is optimised, referred to as single-objective optimisation, the objective is to find a global optimum element s^*, minimum or maximum depending on the problem. The global optimum can be formulated as follows:

Definition 13 (Global Optimum).

Considering a minimisation problem, a solution $s^* \in S$ is a global optimum of an objective function f if:

$$f(s^*) \leq f(s)\ \forall s \in S \tag{3.3}$$

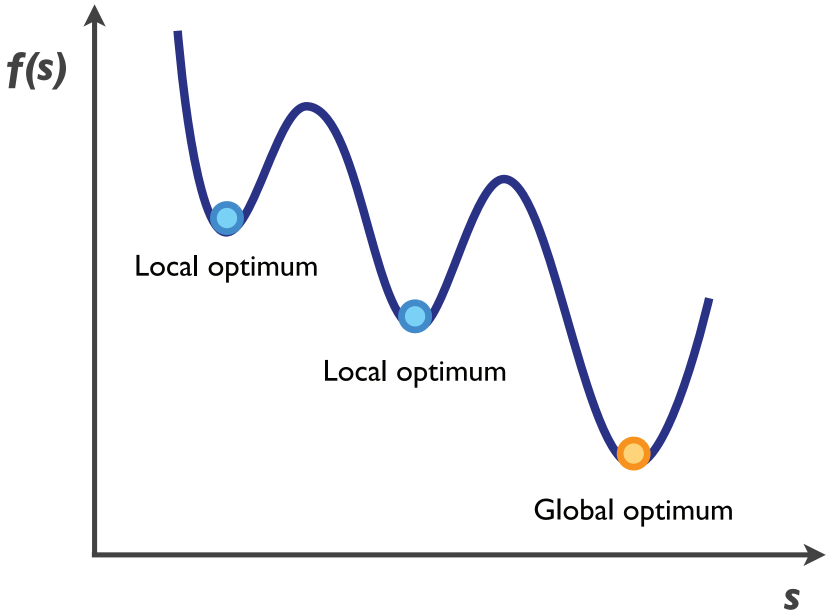


Figure 3.9: Global and local optima in the search space

In Figure 3.9, the global optimum is shown within the search space considering a minimisation problem.

These single-objective optimisation algorithms are used when the optimisation problem focuses on a single objective. However, most of the real-world problems have to deal with different objectives that are usually in conflict (e.g. obtaining highest possible reachability in a broadcast process but using the minimum possible number of nodes forwarding the message). This is known as multi-objective optimisation. The main difference between single and multi-objective optimisation is that for the latter there is not a single optimal solution that satisfies all the objectives but a set.

A general multi-objective optimisation problem (MOP) is to find vectors $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]$ that are optimising the vector of functions $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]$. Each $f_i(\vec{x})$ is a single-objective optimisation problem, and it is one of the objectives to optimise in our MOP. The different objectives must be in conflict with the others, meaning that an increase in the quality of one of them will lead to a decrease in the values of (some of) the others. If the objectives were not in conflict, then we could reformulate the problem as a single-objective one.

Definition 14 (Multi-objective Optimisation).
Find a vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]$ *which satisfies the* m *inequality constraints* $g_i(\vec{x}) \geq 0, i = 1, 2, \ldots, m$, *the* p *equality constraints* $h_i(\vec{x}) = 0, i = 1, 2, \ldots, p$, *and minimises the vector function* $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T$, *where* $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ *is the vector of decision variables.*

In multi-objective optimisation, it is not trivial to decide whether one solution is better than another or not, because it could be better for several objectives, but worse for some others. Therefore, we say that a solution A dominates a solution B if A is strictly better than B in at least one objective and better than or equal to B in the rest of objectives. Two solutions are said to be *non-dominated* if none dominates the other.

The goal of multi-objective optimisation is to find the optimal set of non-dominated solutions to the problem, called the *Pareto optimal set*. The projection of the Pareto optimal set in the objectives domain is called the *Pareto optimal front* (i.e., the $\vec{f}(\vec{x})$ values for every $\vec{x}$ in the Pareto optimal set). Because the Pareto optimal front might contain a large number of solutions, a good multi-objective algorithm must look for a Pareto front with a limited number of solutions, and it should be as close as possible to the optimal front. Additionally, these solutions should be uniformly spread along the Pareto optimal front; otherwise, they would not be very useful to the decision maker.

3.4.1 Challenges tackled in MANETs using metaheuristcs

It is possible to find in the literature approaches that use both single and multi-objective optimisation approaches for addressing the existing problems in ad hoc networks. Before reviewing the literature, we first list and briefly explain the most relevant challenges, that have been addressed in mobile ad hoc networks:

1. energy efficiency: nodes depend on battery life, therefore, reducing the energy consumption will increase the network lifetime. There are different approaches for reducing the energy consumption of a node: decreasing the transmission power, turning devices into sleep mode, reducing the number of communications or retransmissions, etc.;

2. broadcast: a node sends a message to all nodes in the network. As there is no central structure, guaranteeing full coverage is not possible. Therefore, the main goal is to cover as many nodes as possible relying on multi-hop forwarding.

3. routing: a node sends a packet to a destination. As there is no central station or infrastructure in the network, if the destination is not within the source range, intermediate nodes must act as routers and forward the packet. The length of the route found directly impacts on the energy consumption and the network resources, as well as, the maintenance of the routes, the route failures, etc.;

4. network connectivity: devices do move, and the channel varies in time, so that, distant nodes might not be always connected. Maintaining connectivity within the whole network, so that there is always a path between any two nodes, provides more robustness and resilience to failures;

5. clustering: nodes are grouped into clusters, and each cluster has a clusterhead responsible for the main operations. This approach reduces the network overhead, increasing the system capacity;

6. node deployment: specially used in sensors networks where it is important to minimise the number of nodes needed for covering a specific area, maximising at the same time the network lifetime. An efficient node allocation is crucial for the efficiency of the network;

7. selfish behaviour: due to the lack of infrastructure, multi-hop communication is needed for the proper operation of the network. However, nodes are battery limited and acting as routers will only decrease their capabilities with no reward. Selfish nodes are not motivated to collaborate in the forwarding process and drop all packets that are not intended for themselves;

8. security: malicious nodes can perturb the network by compromising its integrity or the availability of a resource. Intrusion detection systems are used for preventing those attacks;

9. quality of service: ad hoc networks usually face adverse conditions as fading, interferences, packet losses, abrupt bandwidth changes, etc. Therefore, providing QoS is challenging.

3.5 State of the art in metaheuristics applied to MANETS

Metaheuristics have been widely used in many different fields for solving many different problems [244, 261, 326]. Next, we are reviewing the literature in bio-inspired algorithms applied only to solve problems in ad hoc networks. We briefly present some of the most relevant works, and classify them according to the taxonomy presented above, and also in terms of the challenges tackled and the techniques used for solving them.

3.5.1 Topology management

Considering that nodes are able to change their transmission power, topology control is about deciding the transmission range that provides a desired property to the network (e.g. connectivity). The main goals of topology control are to reduce the consumption of nodes, and increase the capacity and extend the lifetime of the network. For a more detailed explanation, please refer to [291]. Different techniques have been applied for achieving those objectives, like turning devices into sleep mode, power allocation or node deployment. Next, we are reviewing some works addressing them.

3.5.1.1 Sleep mode

In sensor networks, in order to save energy and extend the lifetime of nodes, and thus the network lifetime, there are some approaches that consider turning off some nodes for a period of time. There always exist two different set of nodes, a set that is active and gathering the data, and a set of sleeping nodes. Efficient synchronisation and scheduling is needed when using this technique.

Finding the minimal set of active sensor that covers a targeted area, while maintaining the rest in sleeping mode is the NP-complete problem addressed in [165]. The multi-objective genetic algorithm NSGA-II is used in order to maximise the coverage (the targeted area) and minimise the cost (the number of nodes). Authors use an offline and centralised approach that requires global knowledge.

In [105, 106], authors use a genetic and a memetic algorithms respectively, for optimising 7 different objectives in an aggregative function. The main goal is the minimisation of some energy-related parameters and the maximisation of sensing points' uniformity, subject to some connectivity constraints and the spatial density requirement. Decisions about the set of active nodes, the role of clusterheads or the transmission range of the active nodes are given by the algorithm. It is implemented repeatedly over different network configurations, providing then, a dynamic sequence of operation modes. It runs offline in a centralised way using global knowledge.

An ant colony optimisation based method, mc-ACO, is used in [378] for prolonging the network lifetime by dividing sensors in two layers. The first layer is the active set of nodes, and the second layer is the successor set, that are in sleep mode until an active sensor runs out of battery. A genetic algorithm is proposed in [152] for the same problem, that focuses on finding the maximal number of disjoint cover sets. Both algorithms were compared in [378] and the results show that mc-ACO performs better for the studied cases. The two approaches are centralised and offline using global knowledge.

In [133], a routing protocol based on ant colonies for MANETs is presented. One of the main characteristics is that nodes turn to sleep mode when the value of the pheromone reaches a predefined threshold. When the node is in sleep mode, it only processes packets that are destined to it. This routing protocol based on ants runs online, is decentralised and uses only local knowledge.

3.5.1.2 Power allocation

The most straight forward approach for reducing the energy consumption of a device is reducing the transmission power. However, this can not be done without considering the impact on other aspects of the network, as its connectivity.

It is usually assumed that networks are homogeneous, where all nodes use the same transmission range, have the same energy consumption, the same battery life, etc. But as it was shown in previous section, nodes caould adjust their transmission power according to their neighbours location, so that the network is still connected but the node can save energy.

We will next present some works that try to find the best possible assignments for the transmission ranges using bio-inspired algorithms.

An improved adaptive Particle Swarm Optimisation (PSO) algorithm for solving the joint opportunistic power and rate allocation in static wireless ad hoc networks, in which all links share the same frequency band, is proposed in [135]. The goal is to find a configuration that maximises the sum of all source utilities while minimising the total power consumption for all links. It is an offline and centralised algorithm that uses global knowledge.

Both linear programming and genetic algorithms are used in [365] to decide the transmission range of the nodes in static networks, so that, the overall energy consumption is optimised, subject to some QoS constraints. It is an offline and centralised technique that uses global knowledge.

A reversed engineered approach is used in [273]. First, near optimal networks using a GA are created, local features of those networks are discovered, and then, local adaptive rules are obtained. Once all the nodes have been deployed and have selected an operating radius based on the heuristic, the local rules are applied to the nodes, so that a heterogeneous network characterised by low short paths and congestion is obtained. The GA is executed offline, centralised and has global knowledge (nodes are assumed to know their global position).

Differential evolution and PSO algorithms are used in [380, 381] respectively, to optimise the power allocation for parallel interference cancellation in wireless code division multiple access system (CDMA). CDMA is not specific for ad hoc networks, but is an access channel method suitable for them. The maximum number of users can be increased using the technique proposed instead of the uniform power distribution. It is an offline and centralised optimisation that uses global knowledge.

A memetic algorithm is used in [192] for tackling the minimum energy network connectivity problem in wireless sensor networks. It uses a genetic algorithm with a problem-specific light-weighted local search that looks for strongly connected networks (they propose a repair method to apply to non-strongly connected networks), with the minimum overall energy consumption. The targeted networks are static and nodes positions are known. It runs offline on the sink of the sensor network, which distributes the solution to the sensors using either multi-hop broadcasting or direct-communicating broadcasting. Therefore, it is an offline and centralised approach that uses global knowledge.

Some authors are considering the joint problem of node location and transmission range assignment at the same time. A multi-objective evolutionary algorithm based on decomposition (MOEA/D) is proposed in [193] to solve the deployment and power assignment problem in static WSN, by maximising the coverage and lifetime of the network. They propose problem-specific evolutionary operators that adapt to the requirements of the specific subproblems into which the original problem is decomposed. It first obtains the Pareto front and Pareto set, and then, choose the best network topology depending on the scenario. The same algorithm is applied to the k-connected deployment and power assignment problem in [191] for optimising the network coverage and lifetime, while maintaining a connectivity constraint in small scale dense WSN. An hybridisation of the same MOEA/D with a problem-specific local search is presented in [190]. All works propose centralised and offline approaches that use global knowledge.

Nowadays, most devices have more than one communication interface, therefore, using a combination of the available interfaces for extending the connectivity of the network has been already addressed in the literature. In [86, 95], authors consider the network connectivity point of view, and propose the use of bypass links through hybrid networks to optimise the network connectivity. Different GAs (coevolutionary, cellular, panmictic) where used to find the most appropriate devices to connect in order to maximise connectivity (high clustering coefficient and low characteristic path length), and to minimise the number of bypass links used (aggregated function). Later, in [96], the same concept was used for vehicular ad hoc networks. Some nodes were selected for connecting to a distant device using other kind of technology. All the approaches are centralised and offline using global knowledge.

3.5.1.3 Node deployment

In wireless sensor networks, arranging the location of nodes to maximise the coverage of a targeted area is one of the first steps of their design. An efficient deployment of sensor nodes is crucial for covering the targeted area while at the same time maintaining the connectivity, reducing the communication cost and improving the resource management for extending the network lifetime.

A genetic algorithm is presented in [136] for the topological design of ad hoc networks with static and mobile nodes for collaborative transport applications. ns2 is used as fitness function. They try to find the best nodes' position and speeds to maximise the communication distances. The optimisation process is centralised, offline and it requires global knowledge.

In many sensor networks, nodes can be differentiated into sensors and actuators. The latter should collect and process the data from the sensors nodes, among other functionalities. In [197], a genetic algorithm is used for finding the location and the minimum number of actors that cover all the sensors. This approach is centralised and offline using global knowledge.

A new multi-objective optimisation algorithm MOEA/DFD for node deployment is presented in [298]. It looks for the optimal arrangement that maximises the coverage and the network lifetime, minimises the energy consumption, and the number of deployed sensor nodes while maintaining connectivity between each sensor and the sink node for proper data transmission. It introduces the concept of fuzzy Pareto dominance for comparing solutions. It is compared to a wide number of other state-of-the-art algorithms outperforming all of them. It is an offline and centralised approach that uses global knowledge.

A multi-objective particle swarm optimisation is presented in [267]. It tries to optimise both, coverage and lifetime of the network, while considering connectivity as a constraint. The algorithm outperforms NSGA-II in terms of the three considered metrics: (1) the size of dominance space, (2) the set coverage metric, and (3) the non-uniformity of the Pareto front. Additionally, a fuzzy based mechanism is used to find out the best compromised solutions. Another PSO for node deployment in sensor networks is presented in [41]. Both approaches are centralised, offline and use global knowledge.

We should also consider the possibility of an autonomous distribution of the nodes of the targeted area when nodes are mobile. In [216], the problem of the optimal sensor deployment in WSN is addressed with a glowworm swarm optimisation algorithm. The goal is to enhance the global coverage in a self organised way (sensors are able to move). Nodes emit luciferin and its intensity depends on the distance to the neighbouring nodes. Sensors are atracted and move to neighbours with higher intensity. This method was proposed for unknown deployment environment, and/or dynamically changing ones. The bio-inspired distributed algorithm is run online. It needs global knowledge as nodes calculate the distance with neighbours.

A distributed and scalable genetic algorithm that also uses traditional and evolutionary game theory for self-spreading autonomous nodes uniformly over a dynamic area is proposed in [199]. Initially, the nodes are placed in a small section simulating a common entry, and the goal is to completely cover the targeted area with a uniform node distribution. Once the location of nodes conform a stable topology, there is no incentive to change the location in the future. It is a decentralised approach that runs online using local knowledge.

Different GAs are studied and compared to Hill Climbing and random walk for obtaining a uniform distribution of nodes over a geographical area [336]. Each node contains a mobile agent running a GA which decides the next direction and speed of the node. It is an online and distributed approach that uses local knowledge.

Literature reveals in [198] that particle swarm optimisation algorithms have been extensively used in wireless sensor networks for determining the position of the minimum number of nodes that provides the coverage, connectivity and energy desired. A couple of the works mentioned in [198] also consider the mobility of the sensors. In both cases [209, 346], the optimisation criterium is to maximise the coverage using a centralised approach with global knowledge. [209] considers a PSO and borrows the crossover and mutation operator from a

GA, while [346] combines a coevolutionary algorithm with the PSO.

PSO is used in [92] for connectivity management in MANETs, by defining some agents that move around to improve network. Authors optimise the movements and locations of these agents. The connectivity is measured using a maximum flow formulation. It is a centralised and online approach that requires global knowledge.

3.5.2 Broadcasting algorithms

There exist in the literature different algorithms inspired in Natured, that focus on solving the broadcast problem. Next, we are reviewing the most relevant.

In [118], an ant colony algorithms was used to minimise the total energy consumption and the lifetime of BAOA, an energy efficient broadcasting algorithms for wireless sensor networks. Authors consider a stationary multi-hop wireless network, where the location of every node is known, so that each sensor is able to estimate the distance to any node. This distance is the weight of the corresponding edge. The path of each ant is stored in a tabu list, and the pheromone on every path is updated in terms of the number of ants that passed. The goal is to find a path where source and destination are the same node, and that passes through all other nodes in the network. It is an online and decentralised approach that uses global knowledge.

As already mentioned, the minimum energy broadcast (MEB) is a NP-hard problem, defined as finding the tree rooted at the source node, that minimises the total energy used to cover all nodes in the network. Different optimisation techniques have been applied to solve this problem in static wireless ad hoc networks. PSO [151], GAs [363], ACO [143, 144], Evolutionary Local Search [353], Iterated Local Search [174], or hybrid GAs [306] are some examples. In all cases, the approaches are centralised, offline and using global knowledge. A multi-objective approach for creating quality tree-based backbones is presented in [266].

In [18], the authors use ESBEA, a multi-objective genetic algorithm that applies binary mechanisms to real numbers. It optimises the performance of a probabilistic broadcast strategy for every node according to their local network density for an efficient broadcast in vehicular ad hoc networks. Four objectives are defined, focusing on the minimisation of the channel utilisation and the broadcasting time. ns-2 simulator is used to evaluated the fitness function. Authors do not provide enough details of the simulation procedure, thus, there is no information about working with local or global knowledge, but the approach is centralised and offline.

GrAnt, a greedy ACO (aco with a greedy transition rule) is proposed in [340] for finding the most promising forwarders from node's social connectivity in delay tolerant networks. The algorithm calculates the degree centrality, the betweenness utility, and the social proximity using global knowledge (the total number of nodes in the network) for characterising the connectivity of nodes. It outperforms two state of the art protocols: PROPHET and Epidemic in the studied scenarios. It is an online and decentralised approach that uses global knowledge.

Another NP-hard problem it is possible to find in the literature is the minimum power symmetric connectivity. It is defined as finding the spanning tree that minimises the energy used to connect all nodes of the wireless network using bidirectional links. Wolf et al. proposed in [352] an iterated local search that outperforms the state of the art. A genetic algorithm is used in [363] for finding the minimum power broadcast problem in wireless ad hoc networks. It outperforms the well known BIP algorithm. These two approaches are also centralised, offline and use global knowledge.

3.5.3 Routing protocols

In ad hoc networks, a packet that is sent from a source node to an intended destination that is not in range, must be relayed by intermediary nodes to be delivered. This is known as multi-hop communication. Routing algorithms are in charge of finding a reliable route between any source and destination. The lack of central infrastructure, the changing topology, the limited resources, and the decentralised nature of ad hoc networks make routing a challenging service.

There are mainly two different approaches in routing algorithms: (1) proactive, and (2) reactive. The former approach periodically exchanges topology information, thus, maintaining routing tables that are available immediately. The drawback of this approach is the cost of maintaining such routing tables, specially if the topology is highly changeable. The reactive strategy only establishes a route when it is needed. Some hybrid approaches have also been proposed with characteristics from both reactive and proactives strategies. A survey on routing algorithms can be found in [57].

As it is a challenging problem in ad hoc networking, literature reveals many works trying to efficiently route a packet to the destination by means of bio-inspired algorithms. There is a big community proposing routing protocols based on ant colony optimisation algorithms. The reason is that it can be executed online with local knowledge, making it directly applicable to real ad hoc networks. There are different surveys on ant-based routing algorithms, some of them can be found in [162, 173, 274, 305, 307]. An extensive survey for swarm intelligence based routing protocols in sensor networks is presented in [287].

Next, we will briefly mention some of the most relevant routing protocols in the literature that are based on ants. Ant-AODV is proposed in [236]. It combines AODV with a distributed topology discovery mechanism based on ants, providing low end-to-end delay. ANRBA is another ant based algorithm that selects the routing paths based on the node status and the network link [112]. It is shown to outperform AODV in MANETs. ARAMA [158] is also an ACO for MANETs that pursues fare resources usage across the network. FACO [127] presets a fuzzy ACO that uses fuzzy logic to take decisions according to the several considered parameters in the routing optimisation. A routing algorithm, called DAR, for critical connectivity based in ants is presented in [279]. DAR outperforms AODV in terms of signalling load and convergence time. Two novel routing algorithms for data networks with dynamic topology based on ants are proposed in [323]. Robust routing is achieved in [111] using routing history. A distributed and autonomic ant based algorithm for efficient routing to maximise the WSN lifetime is proposed in [89]. It uses information on battery life to update routing tables. Yet other ant based routing algorithms are EARA [226], MABR [141], and Adaptive-SDR [178].

HOPNET is presented in [345], a hybrid routing algorithm for MANETs based on ACO and zone routing framework of bordercasting (ZRP). It is compared versus AODV and AdHocNet (not based on zone routing framework), showing a better performance both for low and high mobility and a remarkably higher scalability. AntHocNet is presented in [93], a hybrid ACO based routing algorithm for MANETs that combines proactive and reactive behaviour. The hybridisation of Dynamic MANET On-demand (DYMO) protocol with ACO to design MAR-DYMO (mobility-aware ant colony optimisation routing DYMO) as a routing protocol for VANETs is proposed in [81]. It was validated versus other protocols (AODV and DYMO) on an idealistic urban scenario. A combination of ACO and Zone Based Hierarchical Link State (ZBHLS) protocol is proposed in [36].

There are some works dealing with swarm intelligence based on bees, i.e. BeeSensor and BeeAdHoc [104, 288]. They are bee-inspired power-aware routing protocols that outperforms other state of the art routing protocols. NISR [131], is a scalable routing protocol combining both, ants and bees intelligence. Bees are in charge of finding new routes to the destination and the quality, while ants are in charge of updating the pheromone path. They are all online approaches that use local knowledge.

In the literature, it is possible to also find other metahuristics for the routing problem in ad hoc networks. All these approaches are centralised and offline techniques that use global knowledge for optimising the protocol. A genetic algorithm is proposed in [46] for routing in MANETs with different QoS considerations (delay time, transmission success rate, and communication cost). They consider a multi-objective problem that is solved using the multi-division group model which evolves solutions in the domains of the different objectives separately. There are some limitations on the network changes: they can only occur after some period of stability.

Another GA is used in ad hoc underwater acoustic networks in [312]. It maximises the network lifetime. Each node sends a table to the master node (sink) with the ID and the required power level of every neighbour. The master node gives all this information to the GA that computes the optimal routes. The final optimised routing tree is sent to the nodes.

In [369], both NSGA-II and MODE (multi-objetive differential evolution) algorithms are applied for finding optimal routes in fully connected ad hoc networks. Authors focus on minimising two objectives: energy consumed and end-to-end delay. It is assumed global knowledge so that the source node can evaluate the cost of each potential route to the destination. Results showed that MODE finds solutions closer to the true Pareto front than NSGA-II, and also converges faster.

Different approaches of genetic algorithms are studied in [366], for finding the shortest path in mobile ad hoc networks. Several immigrants and/or memory schemes are integrated into the GAs. It adds individuals to an already evolved population, and memory schemes to reuse stored useful information from previous generation (best individuals in this case). In this work, two approaches are tackled: (1) the elite from the previous generation is used for creating the immigrants, (2) apart from the elite, additional random individuals are created and introduced in the population. Both, immigrant and memory schemes enhance the performance of GAs for finding the shortest path in MANETs.

3.5.4 Multi-path routing

Due to the mobility of the nodes and the variability of the quality of the shared medium, the path obtained by routing algorithms between a source and a destination can usually fail. Multi-path routing consists in finding several routes from source to destination, so that, the routing service is more robust providing reliability of data transmission, load balance (congested nodes), energy conservation (for nodes that are routing most of the packets), QoS, etc. For a more detailed explanation, please refer to [243].

Next, we collect some of the most reliable works focusing on multi-path routing in ad hoc networks.

Similarly to the previously mentioned routing algorithms, many works use the ant colony inspired algorithms for finding multiple path between a source and a destination. They are all online, decentralised and using local knowledge. Multi-Path Dynamic Source Routing Algorithm (MP-DSR) is hybridised in [38] with ACO for better performance. [238] presents an efficient energy aware multi-path routing protocol based on ant colony optimisation. They validate their protocol versus EAAR, AODV, AntHocNet, and MMBCR. A probabilistic routing based on ACO (PERA), that finds several paths that are used as backup of the best ones is presented in [45]. Another ant based routing protocol, ARA, is proposed in [132]. ARA is a reactive protocol that broadcasts ants on demand, thus, reducing the overhead. Its performance is compared to AODV and DSR. In [225], authors proposed AMR, an on-demand routing protocol that combines swarm intelligence and node-disjoint multi-path routing for achieving robustness. It is compared to DSR and ADRA, outperforming both of them in terms of the packet delivery ratio, end to end delay and routing load.

A novel swarm intelligence algorithm, based on the behaviour of termites, is proposed in [280] for dynamic routing in MANETs to minimise the load of nodes using alternative paths. An optimised version of the protocol was recently proposed and validated versus AODV in [150]. As the ant colony optimisation algorithms, Opt-Termite is also an online and decentralised technique that uses local knowledge.

Genetic algorithms have also been applied to the multi-path routing problem. In [292], a new hybrid protocol, GZRP, that uses both proactive and reactive behaviour for finding routes is presented. It is an extension of the Zone Routing Protocol (ZRP), and it uses a genetic algorithm for providing a set of alternative routes to the destination.This approach is centralised, offline and requires global knowledge.

3.5.5 Multicast Routing

Multicast routing is an important network service, consisting on the optimal delivery of information from a source node to a number of destinations or a group. It is an NP-problem and its scalability becomes a very important factor when increasing the network size. In [52], a complete review of the state of the art is given, as well as a taxonomy of the different kinds of protocols.

The use of a GA to solve the dynamic QoS multicast problem in MANETs is proposed in [71]. The GA quickly adapts to the tracked topology changes, and adapts the solutions accordingly, producing high quality ones. Another GA for multicast routing was presented in [73]. Both are a centralised and offline approaches requiring global knowledge.

This problem has also been addressed in vehicular ad hoc networks. [53] presents a multicast routing protocol with QoS considerations. The protocol implements an EA (BLA–Bees Life Algorithm) that assumes global knowledge to look for the optimal multicast tree for every node. They work on a static network and it is centralised and offline.

A decentralised multicast routing algorithm that uses local knowledge online is proposed in [153]. The protocol adapts to the topology variations and satisfies some multimedia QoS requirements using PSO in the devices. The packets priority schedule at every node is also considered and optimised with the PSO. Thus, every device executes 2 PSOs using local information, one for the multicast and the other one for the schedule.

Multicast routing has been extensively addressed by means of the spanning tree. All the approaches found are centralised, offline and using global knowledge. [367] and [368], construct a multicast tree for dealing with multicast routing. Both, use a genetic algorithm for obtaining near-optimal routes on demand. The second approach adds QoS by considering multiple constraints. Another approach that uses a GA for optimising the spanning tree is proposed in [39]. A hybrid discrete PSO presented in [17] also looks for the multicast tree. An aggregate function is used to optimise the packet delivery ratio (PDR), network routing load (NRL), and end-to-end delay (E2ED). Another multi-objective approach was presented in [83] that guarantees some QoS.

3.5.6 Clustering approaches

As already mentioned, arranging the network into groups, introducing some hierarchy into the network, so that some nodes have a special role (usually known as clusterhead), and thus, control the neighbouring devices is called clustering. It offers advantages as making routing tables more stable, higher-layer protocols more scalable, extending the network lifetime, etc. However, the way nodes are grouped, or the selection of the clusterhead are not trivial. Indeed, clustering is an NP-hard problem. Next, we review some works using metaheuristics to solve this problem.

A genetic algorithm is proposed in [185], for finding an optimised clustering for energy efficient routing in static WSN, and therefore extend network lifetime, its stability period, its throughput (number of packets sent from clusterheads to sink nodes), and the total energy left in the network. The obtained protocol is compared versus LEACH, SEP, and HCR (another GA for clustering [157]), clearly outperforming all of them. It is a centralised, offline approach that uses global knowledge.

Another genetic algorithm is used in [271] for finding the optimal cluster configuration in a sensor network. Additionally, a protocol for maximising the network lifetime is presented, and an upper bound is obtained. This approach runs offline in a centralised manner, and uses global knowledge.

Authors proposed in [227] a modified version of the LEACH protocol using Chaos-PSO. Chaotic motion is included in the traditional PSO in order to avoid getting into local optima. Unlike LEACH, the proposed algorithm considers the residual energy of the nodes and the distance of the clusterhead to the sink. It is an online approach that uses global knowledge. Another online approach is found in [186] that uses a reduced complexity GA for obtaining the optimal number of clusters and clusterheads in a sensor network using global knowledge.

There are some works dealing with clustering in mobile ad hoc networks. For example, [72] proposes a number of GAs for the dynamic load balanced clustering problem in MANETs. That is, to find a clustering scheme such that the size of all clusters is balanced. They use static topologies in which some nodes appear/disappear at every change (so nodes do not move). The frequency of topology changes is given by the GA (every 20 generations). Centralised and offline algorithm with global knowledge.

In [335], a GA is used to optimise the number of clusters in a mobile ad hoc network. It first uses the weighted clustering algorithm (WCA) as initial information in order to evolve to a better configuration. It is a centralised and offline technique that uses global knowledge. A similar work was later presented in [164] using a decentralised PSO. In this work, the number of nodes depending on a clusterhead is restricted in order to ensure efficient access to the shared medium. Nodes are divided into groups and four nodes of each group run the PSO. These are offline techniques that make use of global knowledge.

3.5.7 Protocol optimisation

As already explained, part of this thesis focuses on the optimisation of a protocol. Due to the unpredictable and changing topology of mobile ad hoc networks, communication protocols usually rely on some parameters that adapt their behaviour to the current circumstances. The performance of the protocol is highly sensitive to small changes in the set of those configuration parameters. Therefore, fine tuning them for optimally configuring a communication protocol is a complex and critical task.

Some researchers are using metaheuristics for finding the optimal configuration of the parameters conforming a specific protocol. Next, we are mentioning some of the most relevant works in this topic.

A multi-objective approach to find optimal configurations of DFCN broadcasting protocol accounting for network use, coverage, and time is proposed in [29]. It uses a custom simulator, *madhoc* [146], for evaluating the fitness function over a set of different networks. It is a centralised and offline approach that uses local knowledge.

Recently, several evolutionary algorithms are used for optimising the parameters of OLSR routing protocol for vehicular ad hoc networks in [117, 332, 333]. They consider different objectives in a weighted fitness function. In both, [117, 332], PSO, GA, DE and SA are applied for finding the best configuration parameters, while in [333] a parallel genetic algorithm that tries to reduce the power consumption is used. All the proposed techniques are offline, centralised and use local knowledge.

3.5.8 Modelling the mobility of nodes

Creating test beds for mobile ad hoc networks is not only costly but also very difficult. Reproducible experiments are needed for designing and testing protocols, as well as large scale networks for studying the scalability. For small static networks, a test bed could be feasible, but not for large scale mobile ad hoc networks.

Because of all the previously mentioned difficulties, most of the existing works in the literature rely on simulations. Therefore, the accuracy of the simulation is crucial for obtaining the real behaviour of the algorithm studied, thus, the realism of the simulator chosen for the experiments is very important. There are many signal propagation models already implemented, however, modelling the mobility is still flawed.

In vehicular ad hoc networks, devices move at high speeds, on roads, one after the other. Addressing real mobility patterns has been tackled by [302]. It uses real data obtained from counters on the road (number of cars), and information about the most attractive places in Luxembourg for accomplishing a realistic traffic simulator. A genetic algorithm is used to improve the accuracy of the mobility model obtained. This technique is offline, centralised and uses local knowledge.

Traffic routing is addressed in [196] by means of a modified ACO, where ants stop being attracted by the edges that would be most probably chosen by the other ants. Vehicles are prevented of choosing potential congested roads, thus alleviate traffic jams. It uses an online technique that requires a priori knowledge.

3.5.9 Selfish behaviours

Another problem in MANETs is the question of how likely is the cooperation of nodes in packet forwarding. Most of the nodes run on battery, therefore, acting as a router and relying messages to other nodes is battery consuming. Discarding all the messages that are not intended for the node itself and saving energy is tempting [82]. This is known as selfish behaviour, and it is a real threat in ad hoc networks where cooperation between nodes for forwarding packets on behalf of others is crucial. Consequently, performance of MANETs is very likely to suffer from selfish behaviour [235]. Some researchers are solving this problem by equipping nodes with a reputation management system, where each node of the network is rated in terms of their own experience and reputation data from other nodes. In the literature, the approaches analysing how to cope with non-cooperative nodes in MANETs typically combine game theory with GAs.

In [300, 301], a genetic algorithm is used for finding good strategies in network cooperation. Nodes that do not cooperate are not able to use the network for their own purposes. A game theoretical model of a MANET is combined with a GA to discover robust packet-forwarding strategies. The strategies are based on trust information about nodes asking for the forwarding service. The work demonstrates that as soon as the evolved strategies are used by the majority of nodes, the performance of selfish nodes is very poor. It is a centralised and offline technique that uses local knowledge.

A service-based negotiation mechanism is presented in [94] to encourage node cooperation in ad hoc networks. The model uses a GA for generating the offer or counter offer considering the opponent's offer for obtaining a quick agreement. The algorithm is run online in every negotiation agent, using local knowledge.

In [299], the authors combine evolutionary game-theoretical model with GAs to demonstrate that packet forwarding based on direct reciprocity leads to a cooperative network. The most efficient forwarding strategy is based on the conditionally cooperative tit-for-tat principle (TFT), which assumes initial cooperation and then copying the last move of the opponent. In [189], the authors also use a GA to analyse the evolution of forwarding strategies. They show that if nodes use the evolved strategies and trust evaluation mechanisms then the cooperation level is very high.

3.5.10 Security issues

In mobile ad hoc networks, the lack of central authority, the changing topology, and the vulnerability of the channel makes difficult guaranteeing secured communications. The deployment of security mechanisms is needed.

In [297], genetic programming and grammatical evolution is used for evolving intrusion detection programs in MANETs. It uses evolutionary computation techniques for evolving intrusion detection rules of two types of known attacks in routing algorithms. The rules obtained will be executed online and locally. As intrusion detection systems are resource consuming, a multi-objective algorithm, SPEA2, is used for obtaining optimal trade-off between intrusion detection and power consumption. It minimises the energy consumption while maximises the coverage and exposure of the sensor nodes. Both are centralised and offline approaches that use global knowledge.

Localising the intruder (malicious node) in wireless ad hoc networks is achieved using anchor points in [91]. A node detecting an intrusion triggers the localisation estimation algorithm, and sends messages to calculate the distance. The anchor points use a GA for locating the intruder given the signal strength inputs. It is a centralised and online approach that uses global knowledge.

3.5.11 Other applications

A lossy compression algorithm for sensor networks is presented in [234]. As sensors are battery limited and radio communication is, generally, power consuming, the goal is to reduce the data transmitted as much as possible. There is always a tradeoff between the compression rate and the information loss. For that, the well-know multi-objective evolutionary algorithm, NSGA-II, is used to obtain a set of optimal solutions with different tradeoff among the information entropy, the complexity and the Signal to Noise Ratio (SNR). Therefore, the user can decide the most suitable combination depending on the application. It is an offline and centralised approach that uses global knowledge.

In MANETs, packet losses are not only due to congestion as it used to be in wired networks. [285] presents a middleware that allows an adaptive behaviour of the application layer according to the network conditions while still maintaining QoS in adverse situations. It uses a genetic algorithm for finding the best values of the parameters that conform the middleware, as the best moment to trigger the adaptation process. It is a centralised approach that runs off line using global knowledge.

3.5.12 Summary

In Figure 3.10, a classification of the works that were previously described is shown. As we can see, we could not find any work that executes the optimisation algorithm offline in a decentralised manner. That is normal, as there is no need of decentralisation if the algorithm is not run locally on every node. We did not find either any algorithm that is executed online, but using a central unit with local knowledge. As it is none sense to use a central unit that only uses local information.

From the figure, we can see that most of the works are based on a centralised and offline approach that uses global knowledge. However, most of the works dealing with routing or multi-path routing are using a decentralised and online technique that uses local knowledge (generally, ACO). Regarding the optimisation of protocols, they all follow the same approach by optimising offline in a central structure but using local knowledge, as usually the network is not known, and indeed, the topology is changing.

3.6 Open issues

Even though, many different approaches of broadcasting algorithms were reviewed, we consider there are still remaining open issues. Considering the approaches using an underlying topology for disseminating the message, we can see that the tree based topology approach has been extensively studied. However, many of them use global knowledge or consider static or low mobility. Most of the distributed versions require either a preexisting routing architecture, or the the distance to all neighbours and the distance between its neighbouring nodes (mostly using GPS service), or do not properly handle topology changes.

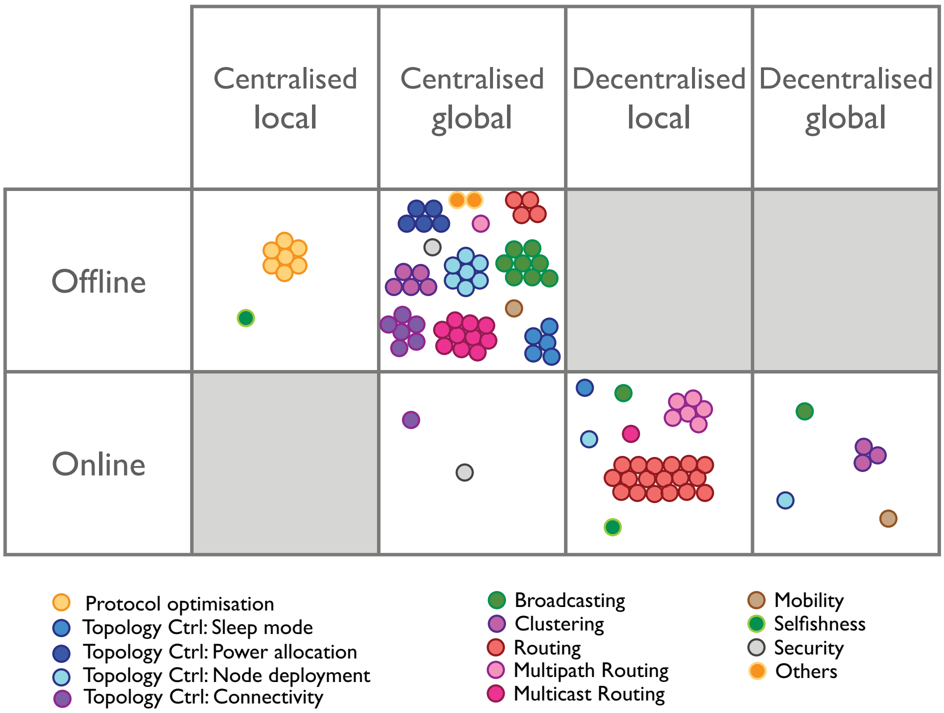


Figure 3.10: Classification of the described works

Most of these algorithms are not able to cope with continuous changing topologies, or the appearance/disappearance of links. Hoever, in reality, the tree should immediately react to all these changes.

Additionally, the existing tree broadcast algorithms use multicast to send the message explicitly to only nodes belonging to the tree. In this situation, the tree topology must cover all the network in a single tree, and have relatively small changes for obtaining high reachability. That might be the case in static networks, but it is not when dealing with highly mobile networks.

Regarding the approaches that reduce the transmission range but do not use any underlying topology, we could see that most of them use 2-hop neighbours information, what makes the algorithm slowly react to changes, as two time intervals are needed to update all neighbours with the changes.

The reviewed works that consider a variable transmission range and also the exclusion of nodes from the 1-hop neighbourhood use the same technique: if the transmission power to reach a neighbour is higher than the sum of the transmission power needed to reach an intermediate node and the power from the intermediate to the intended neighbour, then, this neighbour is excluded from the 1-hop neighbour.

However, this technique supposes that depending on the topology it is likely that the source node reduces the transmission power in order to get the closest node, and highly increases the number of hops needed to relay any message in the network. In Figure 3.11, we can see that the node A checks $P_{AD} > P_{AC} + P_{CD}$ so it decreases the transmission power in order to reach node C. However, $P_{AC} > P_{AB} + P_{BC}$, thus, it will reduce the transmission power in order to reach only node B. When B receives the message the procedure is the

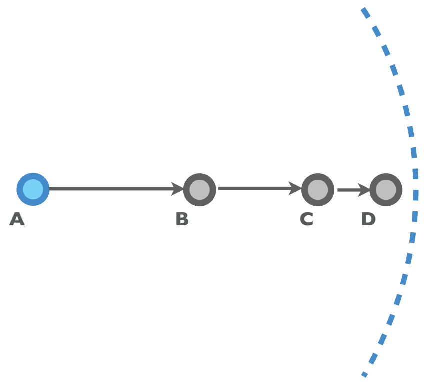


Figure 3.11: Increasing the number of hops

same: checks and $P_{BD} > P_{BC} + P_{CD}$, thus discards node D from the 1-hop neighbourhood. Therefore, finally, the number of hops largely increases. This is not desirable in ad hoc networks, especially when then topology is highly changeable.

Therefore, we consider that an algorithm that rapidly reacts to changes, i.e. uses only 1-hop neighbours information and reduces the transmission power but not to any extent, that is, maintaining a balance between the number of hops and the transmission power used, is desired and missing in the literature.

Part II

Research Objectives and Experimental Framework

Chapter 4

Research Objectives

Contents

Due to the broadcast nature of wireless networks, broadcasting is one of the most suitable techniques for disseminating a message in MANETs. Indeed, broadcasting is one of the main low level operations and many applications and even other protocols rely on its service. Therefore, designing efficient broadcasting algorithms is crucial for the proper functioning of the ad hoc network.

In this thesis, we are focusing on efficiently broadcast a message in ad hoc networks. In these resource constrained networks, approaches for alleviating the broadcast storm problem, i.e. reducing the number of redundant messages, and thus, the use of the network resources are not only desirable but needed. However, many other challenges specific of this kind networks must be also addressed.

As already mentioned in Chapter 3, there are already many different approaches that tackle this problem, but there are still some open issues that need to be solved. In this Chapter, we are describing the key objectives we are targeting in this thesis, as well as the methodology follow to get them.

4.1 Identifying the goals

Taking into account the challenges of mobile ad hoc networks mentioned in Chapter 2, specific features must be considered in the design of broadcasting algorithms. Decentralised algorithms that take decision locally, and where nodes collaborate for routing packets are essential. Moreover, protocols must adapt to the changing topology provoked by the node mobility and the intrinsic dynamism of MANETs. Algorithms not only must have mobility support, but also adaptability and reactiveness to the frequent changes are key. It is important also to know how long the protocol takes to recover from a change. Any valid solution for ad hoc networks must be adaptive, efficient and scalable.

Boukerche showed in [56], that a failure discovering the destination in routing causes fewer problems than the inherent overhead of blind flooding. Therefore, efficient broadcasting algorithms able to cope with these challenges are really needed and are still an open issue.

4.1.1 Broadcasting approaches

After reviewing the literature, we noticed that many of the algorithms that build a tree topology over the network require either global knowledge, e.g. *BIP* [348], or deal with static or low mobility, e.g. LMST [212]. In ad hoc networks, a node is very unlikely to get global information, specially updated one. The propagation of information based on exchanging messages locally, might only work for very small networks, and is not scalable. Most of the distributed versions reviewed require either a preexisting routing architecture, or the the distance to all neighbours and the distance between its neighbouring nodes (mostly using GPS service). Additionally, although the tree should immediately react to topology changes, most of the proposed algorithms do not properly handle them nor the appearance/disappearance of links.

Because usually the broadcast tree algorithm sends the message explicitly only to nodes belonging to the tree (multicast), we consider a new reformulation of these broadcasting algorithms is also needed. Considering the highly changeable topology and the network partitioning, most probably there is not a single tree in the network but a forest. Thus, using multicast, the message will be only disseminated within a tree of the forest. Additionally, due to the broadcast nature of the wireless network, we should not refrain a node from receiving the broadcast message even though the sender does not belong to the same tree.

Therefore, one of the main objectives of this thesis is to design a efficient broadcast algorithm that relies on a tree topology able to locally deal with frequent topology changes, and that is also able to cope with different trees over the network. In Chapter 6 we give all the details of the proposed approach.

However, the underlying topology might not be always present in the network. If we also consider that devices rely on battery, energy consumption is of extreme importance. As stated in [195], wireless communication could be responsible for half of the total energy consumption of a device. Therefore, trying to reduce either the energy consumption and/or the number of nodes rebroadcasting the message is key in the design of dissemination algorithms.

We already mentioned in Chapter 3, that the there still open issues when considering dissemination algorithms that reduce the energy consumption by means of a variable transmission range. They usually rely on 2-hop neighbours information, on estimations of the path loss model or they promote to largely increase the number of hops in order to reduce the transmission range.

Therefore, we are also proposing a dissemination algorithm that rapidly reacts to changes, i.e. uses only 1-hop neighbours information, that does not require any additional help (underlying topology), and where nodes are capable of varying the transmission power but not reducing it to any extent (maintaining a tradeoff between the transmission range and the number of hops). By doing this, nodes not only save energy but also the interferences, increasing the spectrum efficiency. In Chapter 7, we deeply explain this approach.

Summarising, we can say that in this thesis, we are presenting two different approaches for disseminating a message: (1) a tree based algorithm with fixed transmission power, and (2) a context aware protocol able to vary the transmission range.

A protocol that efficiently work in very sparse networks, will have difficulties for providing the desired performance in very dense networks. The adaptability is usually obtained by relying on different thresholds so that the behaviour of the protocol adapts to the current circumstances. The big majority of protocols rely on them for varying their behaviour.

The selected value of those thresholds directly influences on the performance of the algorithms. Finding these optimal values is not trivial, indeed is a complex and critical task, as protocols are usually very sensitive to small changes. Slightly changing the value of one threshold can provoke a considerable change in its behaviour. Thus, much effort must be focused in the process of selecting these values.

4.2 Modelling the optimisation process

As it was highlighted in Chapter 3, metaheuristics [119] are iterative stochastic optimisation tools that are able to provide good solutions in reasonable time for highly complex optimisation problems. They have been widely applied to solve problems in mobile ad hoc networks with great success. In this thesis, we are proposing to use these approximation algorithms for finding the optimal configuration of the algorithms.

As we already mentioned, metaheuristics are used for solving optimisation problems. According to the definition 3.2, any optimisation problem needs a quality criterion $f(s)$ known as the *objective function or fitness function*. It measures the adequacy of every solution defining a total order relation between any pair of solutions. This quality value is known as the *fitness value*. The *fitness function* is problem dependent (directly related to the objectives of the problem) and its definition is crucial as it is responsible for guiding the search to good solutions. Therefore, objectives must be carefully identified.

4.2.1 The multi-objective nature of efficient broadcasting algorithms

In conventional networks where there are no partition, resource limitation nor unpredictable topology, broadcasting is easier. However, this is not the case in MANETs. When thinking on efficiently design a broadcast algorithm, the first goal that appears in mind is coverage. That is, to reach as many devices as possible in the network (ideally all of them). Nonetheless, there are many other aspects, due to the intrinsic multi-objective nature of the algorithm. Some of the most relevant objectives in broadcasting are:

1. coverage: the number of nodes in the network that actually gets the broadcast message;

2. network resources: the number of nodes that rebroadcast the message;

3. energy used: the total energy used by the broadcasting process;

4. broadcast time: the actual time since the source node sends the message until the last node receives it.

Obviously, an efficient broadcast algorithm always tries to maximise the coverage and minimise the other three objectives: network resources, energy used and broadcast time. Satisfying all those objectives is not an easy task as most of the objectives are in conflict, i.e. promoting one objective implies the reduction of the quality of another. We have the *simple flooding* example, where all nodes are reached within one connected network, but at the expenses of the highest possible number of retransmissions. For the same reason, a really low broadcast time can be obtained if there is no retransmission, or really low number of retransmissions, but of course, this implies that the coverage drops. Reducing the energy used might result in less rebroadcast and then less coverage. Smart tradeoffs between the objectives are needed.

4.2.2 Finding the interaction between parameters and objectives

In the early stages, the definition of the objectives of the problem is crucial because all the obtained solutions are oriented to not only satisfy but promote those objectives while possibly worsening others. Additionally, the parameters of the broadcasting algorithm interact and influence the different objectives that have been previously defined. There could be parameters that highly influence the objectives, or do not influence them at all, or only in a specific range of their domain. These interaction and influence between the parameters and the objectives can be obtained performing a *sensetivity analysis*. Next, we define it:

> **Definition 15** (Sensitivity analysis).
>
> *Sensitivity analysis can be (loosely) defined as to ascertain how a given model (numerical or otherwise) depends on its input factors. SA helps to understand the behaviour of a model, the coherence between a model and the world, and how different parts of the model interplay [290].*

This method is based on decomposing the variance of the output, as introduced by Saltelli et al. in [290]. The exact implementation used is an extension to the Fourier Amplitude Sensitivity Test proposed in [289], called Fast99. It is used to compute the first order effects and interactions for each parameter. Parameters interaction occurs when the effect of the parameters on the output is not a sum of their linear effects.

Thus, before going into the optimisation process itself, we propose to:

- first identify all the parameters that influence the behaviour of the protocol;
- define the pursued objectives;
- perform a sensitivity analysis in order to see their interactions and influences;
- tackle the optimisation algorithm.

4.3 Need for a modular framework design

In this thesis we are pursuing different objectives. Our main goal is to efficiently disseminate a message in mobile ad hoc networks. However, in order to achieve this objective, several goals must be reached (e.g. design of the protocol, evaluation, optimisation, etc.). These goals are different, although they are usually associated, and not all of them are always required, and might change from one situation to another. Therefore, a modular framework composed of different modules that can be plugged in or unplugged is needed.

This framework must give modularity and clarity to the process of the development of a communication algorithm. It should be able to separately tackle all the already mentioned goals in a efficient and clear way, so that anyone can easily understand, use or modify the work proposed in this thesis.

Using this framework anyone is able to efficiently evaluate, validate and optimise any algorithm under different circumstances or using any tool, without reformulating all the problem again. It must be defined so that the modules can be used or removed according to the targeted goal. Modules must be tightly integrated, because generally, they all collaborate for solving the problem.

Next chapter introduces the proposed experimental framework used during this thesis and explains the different modules composing it and the interactions between them.

Chapter 5

The Experimental Framework

Contents

As nowadays the development of real MANETs is still an open issue, researchers need to rely on different mechanisms for evaluating, validating and optimising any protocol designed for these networks. The existing mechanisms are: creating *testbeds*, or simulations.

Testbeds are experimental networks that allow researchers to run experiments in real devices. However, they present many drawbacks what usually prevent researchers from their use. Among them, we can highlight the high cost related to their creation, the difficulty of monitoring them, the small number of nodes involved, the limited mobility of devices, the lack of reproducibility, etc.

Due to the already mentioned drawbacks, simulators are the most commonly used technique to emulate the behaviour of the MANET for evaluating, validating, and optimising algorithms. There are also two possibilities, either to use an existing general purpose simulator (e.g. ns-3 simulator [2, 200]), or use or conceive a custom simulator tailored to specific purposes [146].

There are many different issues that must be efficiently addressed for obtaining realistic results when simulating the behaviour of a mobile ad hoc network. In order to ease the early stages of the design and development of an algorithm in mobile ad hoc networks, we are proposing a modular experimental framework based on simulation. This framework will be presented in the next section, and later the methodology needed.

5.1 Overview of the framework

There exist different steps in the development of an algorithm. First, the protocol must be implemented and evaluated over different use cases in order to see if the algorithm actually works. Secondly, it must be validated by comparing to other state of the art protocol. If the algorithm contains different threshold values that have been experimentally chosen, there is another required step: the optimisation of the algorithm.

In order to give clarity and modularity to the development process, we are proposing an experimental framework for efficiently evaluate, validate and optimise any algorithm designed for mobile ad hoc networks under different circumstances or different tools. It is composed of several modules that can be easily plugged in or unplugged during the development step. All the modules composing the framework must be tightly integrated, as in most cases they must collaborate to solve the problems at hands. The proposed framework is shown in Figure 5.1, and explained next.

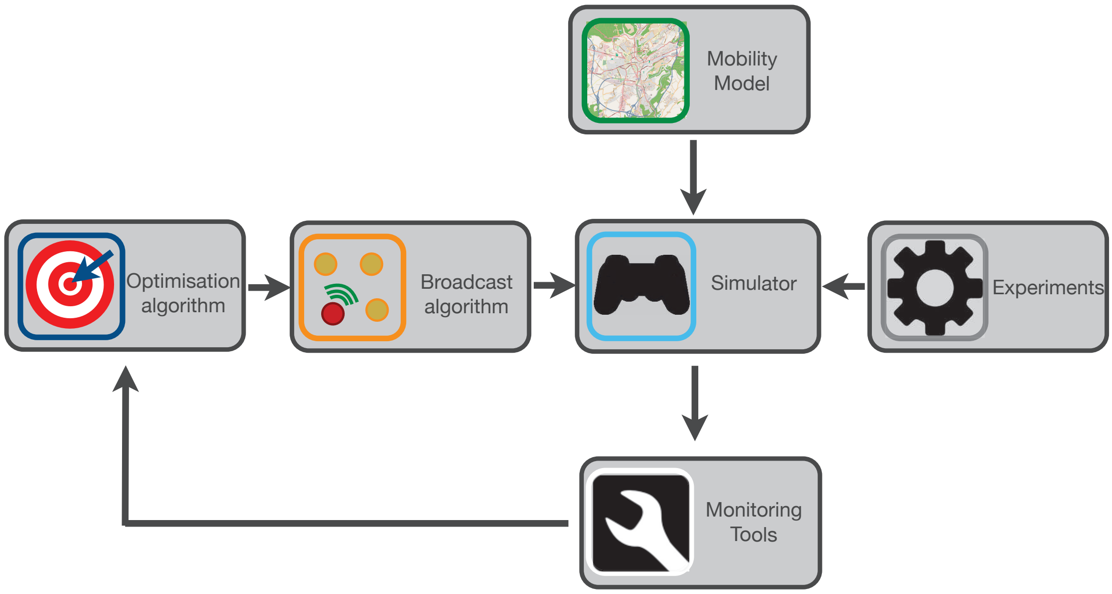


Figure 5.1: The proposed experimental framework

- Broadcast algorithm module: the algorithm we are designing is included in this module. It can be any type of algorithm, not only broadcasting, that needs to be validated and evaluated throughout experiments.
- Simulator module: the network simulator used for analysing the behaviour of the targeted algorithm. In this module, it is possible to specify the technology used, and the features of the communication channel, e.g. the propagation path loss or propagation delay model.
- Mobility model module: depending on the type of network (sensor, vehicular, etc.), and the environment that want to be studied the nodes must follow different mobility patterns. The mobility model must be very accurate in order to perform realistic simulations.
- Experiments module: the number of nodes composing the network or the size of the simulation area influence the behaviour of the algorithm, allowing to study different use cases, like large scale network simulations.
- Monitoring tools module: processes the output given by the simulation. In case we are optimising the algorithm, the output of the simulator is processed by the monitoring

tools and then, used as input for the optimisation algorithm. Indeed, it gives the fitness value of the solutions (that was explained in Chapter 4). In order to have reliable results, not only one execution is performed but several, being the fitness value the average value of all executions.

- Optimisation algorithm module: it is required in case we are optimising the broadcast protocol, and it is not used when evaluating or validating the algorithm. In this module, it is set the optimisation algorithm we want to use for calculating the set of near optimal solutions for the broadcasting parameters. The optimisation algorithm proposes a solution that is given to the broadcast algorithm and is evaluated in the simulator module. The monitoring tools process the output of the simulations and compute the fitness value that will be given to the optimisation algorithm as a metric of the quality of the proposed solution.

The optimisation algorithm module is just used in order to find the best configuration when the protocol has different parameters that highly influence in the performance of the algorithm.

5.2 Methodology

As already explained, we can use the proposed framework for evaluating, validating and optimising any protocol under development for mobile ad hoc networks.

In the early stages of the process, i.e. for evaluating and validating the protocol at hands, the module of the optimisation problem is not needed. Thus, we just unplug this module from the framework and use the others.

In order to get accurate results of the performance of the protocol, the mobility model used should be as realistic as possible. Considering the *random walk* [128] for studying the behaviour of vehicular ad hoc networks is not realistic, as well as devices moving in a line at high speeds is not appropriate for simulating a pedestrian ad hoc network.

Then, the simulator (network simulator in this case) must be chosen. The context where the algorithm will be applied must be considered or studied during the design step, so that the algorithm fits to the specific characteristics, e.g. the high speeds of vehicular ad hoc networks. As it happens with the mobility model, the network simulator is very important for obtaining accurate results of the performance of the protocol. We can find custom simulators tailored for specific purpose and general use simulators. Realism is essential whatever option is chosen. Additionally, the network simulator and mobility model can be tightly related, and this collaboration has to be considered. For example, if we are analysing protocols in VANETs, and we have in mind to use a traffic simulator for creating the mobility traces (like sumo [49] or paramics [12]), we should check that the selected network simulator is able to use the traces provided by that traffic simulator.

The experiments module specifies the scenarios we want to simulate, i.e. the size and shape of the area, the number of nodes, the transmission power that devices use, etc. It is possible to simulate very sparse and partitioned networks or really dense ones. If a deep analysis of the behaviour of the protocol under many different circumstances is needed, the only module that has to be changed is this one. All others modules in the framework remain the same.

Finally, the monitoring tools module processes the data obtained after performing the simulation. Obtaining good performance of the behaviour of an algorithm in one specific network does not imply it is generally a good protocol as it was tested under very specific circumstances. In order to have confident results, usually more than one simulation is required. This module is in charge of calculating the average values of the different executions.

There are many communication protocols, specially in mobile ad hoc networks, that adapt their behaviour to the current circumstances in order obtain a better performance. This adaptability usually relies on different thresholds. The value of those thresholds is very important, and highly influences the performance of the protocol. In order to provide an optimal configuration of the algorithm, this last step of optimisation is required. In this thesis, we are proposing to add an additional module to the already explained framework for optimising those parameters. The mechanism is as follows: the optimisation algorithm proposes the set of values that are validated through simulation. In order to have confident results, the quality of the proposed solution (fitness value) is calculated as the average of different simulations at the monitoring tools module. This fitness value is given to the optimisation algorithm as a metric of the quality of the solution, so that two different solutions can be compared.

The experimental framework has been executed on the HPC facilities of the University of Luxembourg[1], specifically, on the 2.26GHz Intel Xeon L5640 8 cores processor nodes, having 24GB RAM. These nodes run Debian Linux version 6.0.5 (with kernel 2.6.32-5-amd64) and Java version 1.6.0_25.

[1] http://hpc.uni.lu/

Part III

Broadcast Algorithms

Chapter 6

Broadcasting Over a Tree Topology

Contents

One classical way to enable service on mobile ad hoc networks consists in establishing a topology management structure such as a dominant sequence [294] or a tree [64]. This structure is maintained during the whole lifecycle of the ad hoc network using the exchange of beacon messages and minimal information stored at the level of the nodes. Higher level services such as routing then rely on the underlying structure.

As stated in Chapter 3, previous researchers demonstrated the validity of using spanning trees in the networking area. Indeed, establishing a spanning tree in the network is a well known strategy for providing efficient communication and routing algorithms in wired networks [4, 237], as well as in mobile ad hoc networks [138, 264, 282, 322, 377]. Moreover, in mobile networks where devices are moving and the topology is fluctuant, it is possible to create a tree where the nodes chosen to conform the tree are the ones with more stable links [264]. Therefore, for any critical application, using this kind of structures will provide more reliable results.

It might be called into question the suitability of using a tree topology in networks where the topology is very changeable and the speed of devices is high, e.g. VANETs. Therefore, in this work we also validate the use of tree topologies in highly mobile ad hoc networks.

6.1 Tree construction

In order to efficiently construct the underlying topology we are considering the *Dynamicity Aware - Graph Relabeling System* (DA-GRS) [64], an extension of Graph Relabeling System [311]. DAGRS is a high level abstraction model that can improve the development of self-organized systems. It was chosen because of its ability to take into account the locality of the communication, the dynamics of the topology in an innate way, and to provide an abstraction of the communication layer. DA-GRS models topology changes and interaction between devices using a label system. The model supports the design of algorithms in which a computation steps involves only direct neighbours and devices react to the appearance/disappearance of neighbouring nodes [66].

6.1.1 Dynamicity Aware Graph Relabelling Systems, DAGRS

DAGRS is a completely decentralised model for building tree based topologies in dynamic graphs. The model only uses information about direct neighbours for building the topology. Hence, if we represent the network as a graph, where vehicles are the set of vertices *(V)*, and the links between them are the edges of the graph, *(E)*, then the dynamics of the network is represented by the fact that both V and E can change at any time.

The tree construction (or forest, see 3.3.1.2) in DAGRS, is operated by locally applying some simple rules in every node, as shown in Figure 6.1. T represents a node with token, N is a device without token, and *Any* means it can be any of them. The numbers on the edges are labels representing the route to the token. A token does not have any information related to the tree, it just provides a status to a node that will allow the tree merging process. By using a single token per tree mechanism, the formation of cycles is not possible, as only the node possessing the token is able to merge its tree with another. The 4 rules defined in DAGRS are:

- rule 1: A tree link breaks, and the node belongs to the sub-tree which does not possess the token (indicated by the label on the edge). In this case the node must regenerate the token, otherwise there will exist a tree without a token (which is an undesirable situation).
- rule 2: A tree link breaks, and the broken link occurs at a node which currently belongs to the sub-tree which possesses the token. In this case, the node does nothing regarding the maintenance of the token.
- rule 3: When a node with token meets another device possessing a token; both nodes will try to merge their trees in order to obtain a bigger tree from the two existing ones. The trees merging process starts. As result of this rule, a bigger tree and only one

token remain (the merging process discards one token automatically in order to have one and only one token within each tree).

- rule 4: Token traversal in general case: the token visits the nodes of the tree following a given strategy.

Initially, all devices are labelled T, which means they all constitute one-node trees. The algorithm then, performs on the basis of the four rules described above to generate the spanning forest topology in the dynamic network. It is important to emphasise that DAGRS itself does not model services or applications, it simply models the mechanisms to handle with topology changes and interaction between devices. It is only used 1-hop neighbours information, so it is a localised algorithm.

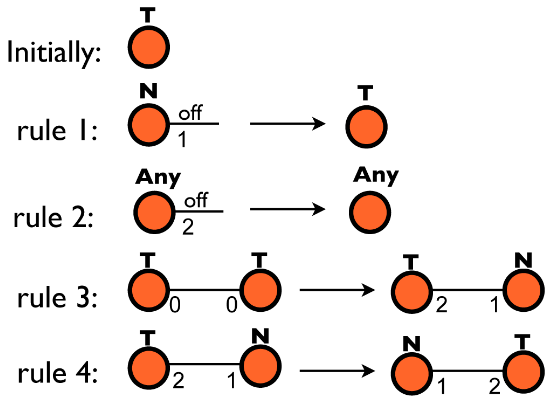


Figure 6.1: DAGRS rules for creating spanning forest topologies.

Due to the lack of central unit, it is necessary to exchange some messages between nodes for merging trees and also for circulating the token. When a device receives the token, it should check if there is any neighbour holding another token around, but this behaviour would lead to a massive interchange of data in the network. In our implementation, as the protocol aims to be specifically designed for highly fluctuating topology, we benefit from the high mobility: only when a change in the neighbourhood is detected (since the last time the device had the token), the node sends a message to check if there is any neighbour with token around. We consider that there is a change in the neighbourhood when a new neighbour arrives or an old one leaves. These changes are detected using the beacons (*hello* messages).

Now, let us explain the message exchange that is produced when two neighbours holding tokens meet. We can imagine node *'A'* receives the token and also detects a change in its neighbourhood. In this situation, it will send a broadcast message to its 1-hop neighbours to notify that it has the token, and it waits for an answer of any neighbouring device saying it also has the token. Due to the mobility and the high speed of the devices, in our implementation the first node answering will be the selected device for merging trees. When the node sends this broadcast message it also sets a timer, if it does not receive any answer during 0.3 seconds, the node will circulate the token. This time was experimentally chosen [282], and it ensures a neighbouring device is able to receive the broadcast message and send an answer (in case it has the token) before the timer expires.

When *'A'* receives the answer of a neighbouring device holding a token, the merging process starts. During this process one of the tokens must be deleted. In DAGRS, any of the two tokens are candidates to be deleted, but in a distributed system a procedure ensuring that only one of token is deleted must be implemented. In our implementation, the node with higher address will be the one deleting the token.

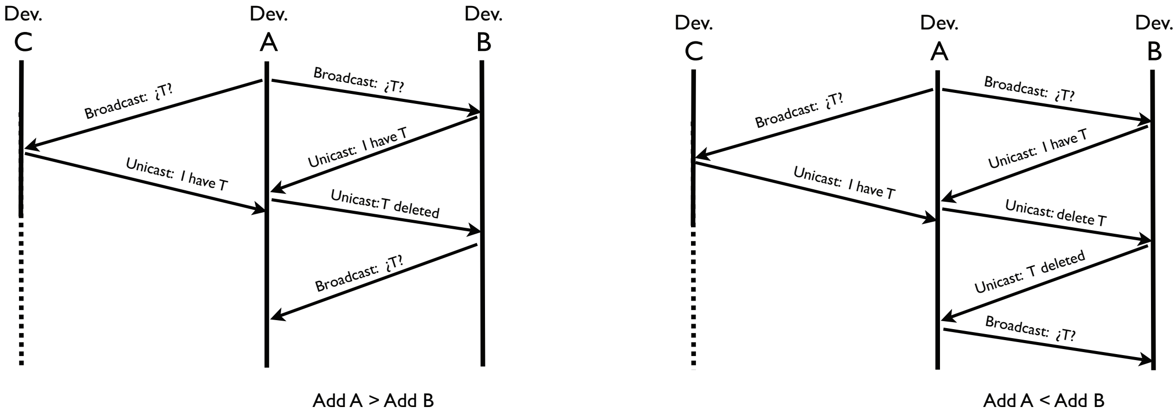


Figure 6.2: Messages exchanged for creating a tree topology in the network.

In Fig. 6.2, we show the messages exchanged between two nodes for merging their corresponding trees. In the left diagram, node *'A'* is deleting the token, meanwhile in the one on the right it will be *'B'*. As it was explained before, the first node answering will be selected for merging their trees (in both diagrams it is node *'B'*). When node *'A'* receives the message from *'B'*, it checks which one has the highest address. In case it is *'A'* (left figure), it will first create the logical link between them, delete the token and send a message to *'B'* informing it about the deletion of its token. In case *'B'* has the highest address, node *'A'* sends a message requesting the token deletion and node *'B'* will create the logical link, delete it, and send a confirmation message. When node *'A'* receives the token deletion confirmation, it creates the logical link between them. In both cases, the node that finally possesses the token sends a broadcast message again asking if there is any neighbour with token around.

In Figure 6.3, we show the construction of a tree in different stages. The first step is reflecting the initial situation where all devices possess the token. Then, to merge their trees, a node (the red one in step 2) broadcasts a message (represented by a circle) asking if there is any node possessing a token in its range. All nodes with token answer to it by unicast (represented by an arrow). The first node answering is the one chosen for merging. In step 5 and 6, the messages for merging trees and token deletion are shown, and finally in step 7 the tree is formed (notice that there is only one token in the new tree, the other was deleted). The same procedure starts again in step 8 in order to merge other nodes into one single tree.

6.1.2 Token Traversal Strategies in a Decentralised System

As explained in the previous section, DA-GRS needs a token for creating and maintaining a tree. Every node, at some moment, must possess the token, since it allows looking for neighbours with token to merge trees. The way this token moves along the tree impacts on the spanning tree construction. In literature, it is formally defined as:

Figure 6.3: Steps in the creation of a tree topology in the network.

Definition 16 (Tree traversal).
Tree traversal refers to the process of visiting each node in a tree data structure in a particular manner [125].

In the context of this study, we want the token to traverse less but to have more chance to meet another token. In other words, we want the fastest rate of the tree construction to cover a connected subgraph, which means less number of trees or in the best case, remaining only one tree over a connected subgraph.

We analyse three possible strategies for traversing the token. It is worth noting that all strategies are working in distributed and decentralised manner suiting to work in mobile ad hoc networks.

Randomness

The Randomness here follows the uniform distribution law. Randomness is the heuristic used by DA-GRS by default. The process is done by selecting a node randomly among the list of neighbours. The description of the 'Randomness' traversal technique is described in Algorithm 1.

Algorithm 1: Using Randomness as token traversal strategy

α is the set of neighbours of node ν ;
node ρ is a node selected randomly from set α ;
move token τ_i from node ν to node ρ ;

TABU

TABU creates a list of forbidden movements in which the most recent nodes possessing the token are stored. This list is called as *tabu list*. The algorithm consults the tabu list before sending the token to a neighbour in order to avoid visiting the same node repeatedly. Tabu list uses a fix size of memory, *memory_size*, to set the number of stored nodes in the list. This list is sent within the token, no node memory is used. In Algorithm 2, a detailed description of this strategy is given.

Algorithm 2: Using TABU as token traversal strategy

α is the set of neighbours of node ν ;
β is the TABU list with *memory_size* size;
Set $availableNode = \alpha - \beta$;
if $availableNode \neq \emptyset$ **then**
    node ρ is a node selected randomly from set $availableNode$;
    token τ_i moves from node ν to node ρ;
    if *the number of item of* β *reach the memory_size* **then**
        remove the first item from list β;
    add ρ to the end of list β;
else
    node ρ is a node selected randomly from set α;
    remove item ρ from list β;
    token τ_i move from node ν to node ρ;
    add ρ to the end of list β;

Applying TABU to DA-GRS and the *memory_size* of the list (its length) was also studied in [260]. It was demonstrated that a tabu list longer than 1 did not provide much better results than using a tabu list with size 1. Therefore, we use in our study TABU with size list equal to 1. For brevity, henceforth we will use 'TABU{1}' to represent the usage of TABU at *'memory_size'* equals to one. This is equivalent to prohibiting sending the token back to the previous holder.

Depth First Search (DFS)

DFS is commonly used as token movement technique [282, 319, 377] when dealing with tree based topologies. It imitates the traversal of the classical Depth First Search algorithm (ordering traversal strategy).

Every node has a list with the 1-hop neighbourhood, so we use this list for circulating the token. As we are dealing with distributed systems, it is necessary to keep track of the node that sent the token to the current device (called *upper neighbour*) and also about the nodes that were already visited. In that way, the device is sending the token to all its neighbours. When the token is received from the neighbour the current device sent it to, it will forward it to the next neighbour in the list. Once the list is finished, meaning that the token was already sent to all neighbours, it is then sent back to the *upper neighbour*.

Notice that in this distributed implementation, there is no notion of *root of the tree* as there is no knowledge of the global tree locally in each device. The only similar concept is the one used for DFS, the *upper neighbour*, where a difference is made between this node and the rest of logical neighbours (that can be considered as children). But this differentiation is only taken into account for traversing the token.

In Figure 6.4, the mechanism of DFS is shown. Considering node A sends the token to node B for the first time, node B considers node A as *upper neighbour*. Because node B does not have any other neighbour but node A, it sends the token back to its *upper neighbour* (A

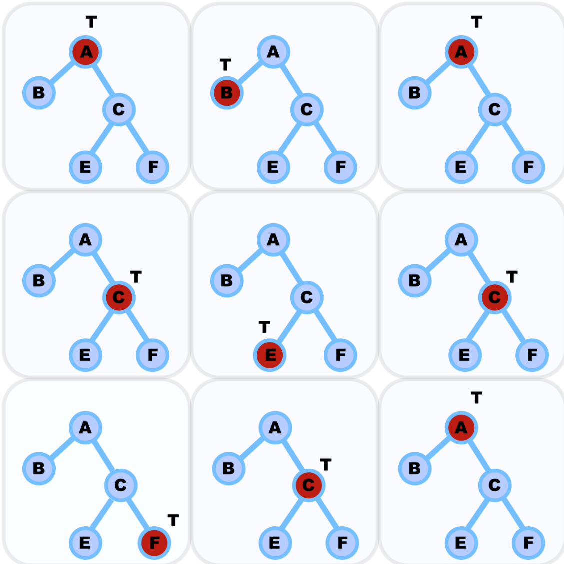


Figure 6.4: The DFS token traversal strategy.

in this situation). Next, after node A receives the token from node B, it sends the token to the next neighbour in the 1-hop neighbour list, that is, node C. Similarly to node B, node C considers node A as *upper neighbour*, and traverses the token among its other neighbours: nodes E and F (for these two nodes C is the *upper neighbour*). When the last neighbour, node F in this case, returns the token, node C sends the token to node A, as it is considered *Cś upper neighbour*. If node A has no more neighbours and no *upper neighbour* (as shown in the example), node A will start the same procedure. This notion of *upper neighbour* is only considered for traversing the token. This implementation is described in Algorithm 3.

As previously done, we model mobile ad hoc networks as a dynamic communication graph (G), where the mobile devices are the set of vertices (V), and the links between them are the edges of the graph, (E). The dynamism of the network is represented by the fact that both V and E can change at any time. Therefore, the graph at a given time t, $G(t)$, is composed of $(V(G(t)), E(G(t)))$.

We want to use the token traversal strategy that achieves the least number of spanning trees in the shortest time. For that, we compared the three token traversal approaches in terms of the two metrics listed now and explained next: *performance ratio* and the *convergence speed*.

1. Performance ratio

   At a given moment t, $G(t)$ may be partitioned into a set of m connected subgraphs. Having Γ as the set of all spanning trees created at moment t of $G(t)$. The quality of the algorithms can be assessed by the number of connected subgraphs (m) over number of trees created (Γ). This quality is determined by the following ratio.

$$performanceRatio(G(t)) = \left(\frac{m}{|\,\Gamma\,|}\right) \tag{6.1}$$

Algorithm 3: Using DFS as token traversal strategy

α is the set of neighbourhood of node ν;
β is the DFS list in node ν;
ϖ is '*upper neighbour*';
δ is the latest node that send τ_i to ν ;
if *ϖ is empty* **then**
  $\varpi = \delta$
Set $availableNode$ = α - β - ϖ;
if $availableNode \neq \emptyset$ **then**
  node ρ is the first node from set $availableNode$;
  move token τ_i from node ν to node ρ;
  add ρ to the end of list β;
else
  clear list β;
  if *ϖ is in the set α* **then**
    move token τ_i from node ν to node ϖ;
    set ϖ to empty;
  else
    $\varpi = \nu$;
    Set $availableNode$ = α - δ;
    node ρ is the first node from set $availableNode$;
    move token τ_i from node ν to node ρ;
    add ρ to the end of list β;

The value of the performance ratio approaching to one means higher quality of the algorithm (less number of trees in a connected subgraph). Having a spanning tree per connected subgraph enables more efficient communication and topology management, since at least, the information can be disseminated systematically via the created spanning tree. This means the algorithm is robust regarding the dynamism of the network because it can construct a tree covering all the nodes conforming the connected subgraph.

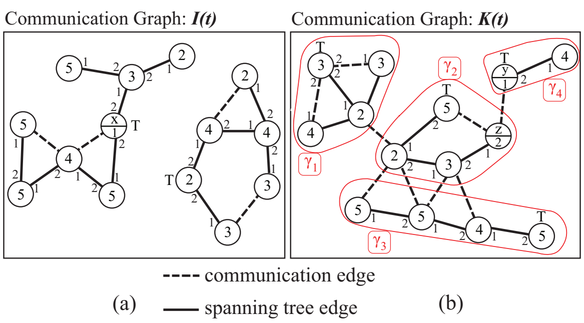


Figure 6.5: An example scenario for illustrating the proposed cost functions for spanning forest

Figures 6.5(a) and (b) illustrate the measurement of all cost functions proposed here. In the figure 6.5(a), the communication graph $I(t)$ has two connected subgraphs, and each connected subgraph has one spanning tree. On the contrary, the communication graph $K(t)$ depicted in figure 6.5(b) has only one connected subgraph but four spanning trees $(\gamma_1, ..., \gamma_4)$. Thus, the $performanceRatio(I(t))$ and $(K(t))$ equal to 1 and 0.25, respectively.

2. Convergence Speed rate

   The convergence speed rate quantifies how fast the tree merges into a spanning tree. The $convergenceSpeedRate()$ is measured based on the number of iterations in simulation. Let Δ be the number of iterations the algorithm required trying to achieve the least $performanceRatio()$ and Δ^* be the number of iterations required per $G(t)$. Having $performanceRatio()$ equal to one within $G(t)$ is an ideal situation. However, having limited merging process causes no guarantee that $performanceRatio()$ will be one, in other words, it is always possible to have multiple trees per connected component at any time t of graph G. In such case, the number of iterations used within that $G(t)$ will be counted into Δ. The lower the value of $convergenceSpeedRate()$ is, the faster the algorithm converges a connected component into a tree. The $convergenceSpeedRate()$ can be written as below.

$$convergenceSpeedRate(G(t)) = \left(\frac{\Delta(G(t))}{\Delta^*(G(t))}\right) * 100 \quad (6.2)$$

Extensive simulations show that the Randomness is the worst strategy for all the studied cases, and DFS is the one that converges faster with the best performance outperforming 'TABU{1}' for the studied cases. For a more detailed information of all the experiments performed and the results obtained, please refer to our preliminary work [265]. Therefore, hereinafter, we always use DFS technique for traversing the token when using DAGRS.

6.2 Validation of the tree topology in VANETs

As we said before, the use of spanning trees is widely studied in the literature for fixed and mobile ad hoc networks, but establishing a tree topology in a VANET may lead us to a contradiction. On the one hand, the restricted mobility of devices, all moving one after the other with quite similar speeds (thus, a low relative speed) and higher coverage range than MANETs make us thinking that is a quite good scenario for a tree topology. On the other hand, the high speed of devices, overtaking each other and meeting devices circulating in different directions provoke that some researchers do not think it is a proper environment for dealing with trees.

As having a tree topology working over MANETs is strongly accepted in the research community, we want to check by making realistic experiments if it is suitable or not to use them in vehicular networks. For that, we are comparing the resources needed for maintaining trees in MANETs and VANETs.

6.2.1 Experimental Setup

Here, we present the instantiation of the framework use in this chapter. As already mentioned in Chapter 5, for the evaluation and validation steps, the optimisation algorithm module is not used. The modules that are used are shown in Figure 6.6.

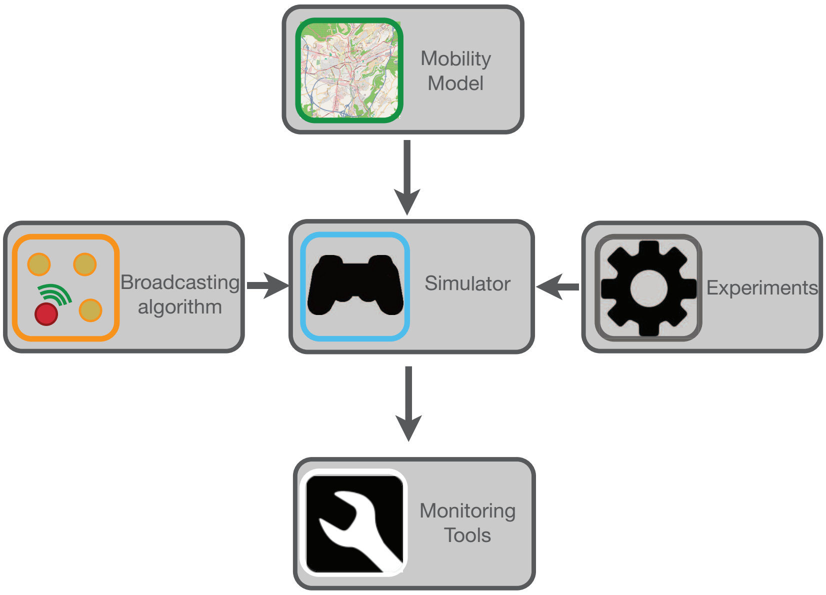


Figure 6.6: The instantiation of the experimental framework.

The instantiation used in this work for the simulator module of our framework (presented in Chapter 5) is the *JANE simulator* [126]. It consists of both a simulation environment and an execution platform. The main feature of JANE is its three steps development; it facilitates the evaluation and minimises the effort needed for software development in MANETs, so that the evaluated code in the simulated environment can be directly executed in real scenarios without modifications. It is implemented in Java and is an event driven simulator. This tool was jointly developed by the universities of Trier (Germany) and Luxembourg.

For the experiments module of the framework, we proposed here three different realistic scenarios for Luxembourg city.

The first scenario is a pedestrian area of a city centre. In this case, the city centre of Luxembourg (see Figure 6.7 is considered. This scenario was first proposed in [32]). The other two proposed scenarios were specifically design in this work for emulating a vehicular ad hoc network in Luxembourg city. Figure 6.8 shows the city centre and the highway. One is a VANET created in the city, and the other is a VANET in a highway (see Figure 6.8).

The main difference between the two scenarios dealing with vehicles is the speed of the devices and, more important, the number of crossings in the path. In the highway there are just a few exits where cars have to reduce the speed, and entrances where devices go slower at the beginning while incorporating into the highway. However, in the city, there are plenty of crossroads where devices reduce their velocity or even stop due to traffic lights or pedestrian crossing and the speed limit is considerably slower than in the highway.

We are considering three different scenarios: (1) the pedestrian city centre, (2) the vehicular city centre and (3) the highway. The density of the networks can not be established as a fixed number of devices for all of them since the environments are completely different. In order to have a balanced density in all environments, we consider the total length of all the paths in each scenario studied, and set the density of the network in nodes per meter. We lengths obtained were: (1) 4473 m in the pedestrian city centre, (2) 13250 m in the

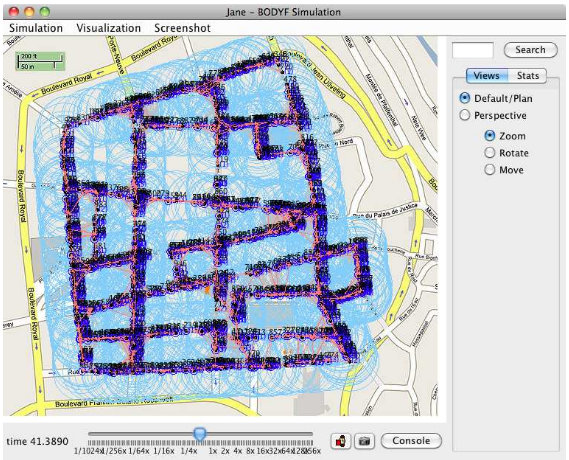


Figure 6.7: Pedestrian area .

(a) Vehicular city centre

(b) Highway

Figure 6.8: Vehicular scenarios.

vehicular city centre and (3) 6150 m in the highway. For our comparison experiments, we are considering three network densities: (1) a device every 25 m, (2) every 15 m and (3) every 10 m. The different configurations obtained for our experiments are presented in Table 6.1.

		MANET	Vehicular centre	Highway
	Length	$4473m$	$13250m$	$6150m$
Densities (devices)	**25 m/dev**	179	530	246
	15 m/dev	298	884	410
	10 m/dev	447	1325	615
	Coverage	$50m$	$250m$	$250m$
	Speed	$0.5-2m/s$	$9-21m/s$	$21-36m/s$

Table 6.1: Configuration parameters.

As shown in Table 6.1, for simulating the pedestrian city centre (MANET) we are considering that the speed can vary between 0.5-2 m/s, that is 1.8-7.2 Km/h and the transmission range of the devices considered is 50 m. For both, the highway and the vehicular city centre scenarios the transmission range is 250 m, meanwhile the speed is different: 9-21 m/s in the city and 21-36 m/s in the highway (30-75 km/h and 75-130km/h, respectively). We are using a collision free network for this study.

Considering the instantiation of the mobility module of the framework is directly related to the scenarios presented above.

We developed a mobility model where vehicles move along streets, stop at crossroads and also overtake each other. In the city centre scenario the real senses of the lanes are kept, showing the most congested areas in reality. In the highway environment all roads have two senses. For the pedestrian model, the devices are also restricted to move only along the streets, but devices move in both senses.

The mobility model is random waypoint with restrictions. Devices can only move from one crossroad to another. The next movement is randomly chosen from the small set of possibles destination considering the current position (the mobility is restricted to the possibles movements along the roads in Luxembourg).

The devices move on a straight line with a constant speed from one position to another. The mobility model uses a directed graph given as XML file for device movement. Vertices are crossroads and contain routing probabilities, all possible destinies have the same probability of being taken; the next route is chosen at random. Edges can have an arbitrary width so that devices move on a lane and not only on a strict line (allowing overtaking).

In these environments, movements are more restricted but, for sure, the simulation is much more realistic. Our main goal is to get results as close as possible to a test on real devices moving along roads in a city centre or a highway.

These mobility model, experiments and simulator modules will be used for the remaining of this Chapter.

In order to show the validity of using tree topologies over VANETs, we want to analyse the stability of the tree in a network by studying the behaviour of its links. As there are different values of the transmission ranges, the resulting topology varies for the different types of ad hoc networks. However, we analyse them from the high level layers where only the performance of the application matters. For that, the monitoring tools module considers three different aspects of the network. The first aspect considered is a measure of the number of tree links created and broken during the simulation. The second aspect is the total number of messages exchanged for creating the tree. And finally, we are also considering the volatility of the tree links, what refers to the number of connections and disconnections a concrete link suffers during its life time. These three aspects are measured during 10 minutes (600 seconds). In order to have reliable results, simulate in 30 different topologies. The results presented in this work are the average values obtained after these simulations.

In the following subsections, we explain and analyse in detail each of the aspects we are considering.

6.2.2 Created and broken links

For obtaining the number of links created and broken, we are running DAGRS in the three scenarios (with three different densities) for 630 seconds. The number of links created/broken obtained is strictly related to the number of devices in the network. Hence, it is necessary to normalise the data obtained by the total number of devices in each network. In Table 6.2, the total number of created or broken links per device during the simulation time (630 seconds) is shown.

	Densities	**MANET**	**Vehicular centre**	**Highway**
Created Links	**25 m/dev**	$10.6423_{\pm 0.2887}$	$12.7742_{\pm 0.2692}$	$16.5217_{\pm 0.4770}$
	15 m/dev	$10.8221_{\pm 0.2110}$	$11.6663_{\pm 0.1654}$	$14.8687_{\pm 0.3862}$
	10 m/dev	$10.5504_{\pm 0.1816}$	$9.6153_{\pm 0.133}$	$13.1442_{\pm 0.2702}$
Broken Links	**25 m/dev**	$9.7572_{\pm 0.2811}$	$11.8890_{\pm 0.2444}$	$15.7003_{\pm 0.4418}$
	15 m/dev	$9.9020_{\pm 0.20691}$	$10.7595_{\pm 0.1449}$	$14.0150_{\pm 0.3570}$
	10 m/dev	$9.6589_{\pm 0.1673}$	$8.7090_{\pm 0.1171}$	$12.2908_{\pm 0.2499}$
Ratio	**25 m/dev**	0.9168	0.9307	0.9502
	15 m/dev	0.9149	0.9222	0.9425
	10 m/dev	0.9155	0.9057	0.9351

Table 6.2: N. of created/broken tree links per device for 630 seconds.

We can observe in Table 6.2, that the behaviour of the tree topology is quite similar in all the scenarios. Both, the number of created and broken links per device in 630 seconds in all scenarios do not differ much. The difference in the number of tree links created in the three scenarios is small. The maximum difference is found with the lowest density between the MANET and the highway, and it implies that in the highway each device creates 6 more links in 630 seconds (1 additional link created every 105 seconds). And the number of broken links differs in the same proportion. As the chance of breaking a tree link is higher when having more created links, we should consider a parameter that gives an idea of the stability of the links, therefore, we compute the ratio between the number of links broken and the number of links created, shown in Table 6.2. The smaller the ratio the more stable the topology (the closer to 0 the more stable). It is also possible to check that the value of this ratio is also quite similar in all the studied cases. The behaviour between the MANET and the other two VANETs is similar, the biggest difference in this ratio is found between the densest network in MANETs and sparsest network in the highway scenario (0.0347).

It is also necessary to remark that when the density of devices is the highest, 1 device every 10 meters, the vehicular environment shows better results than the pedestrian city centre and the highway scenarios.

6.2.3 Number of messages exchanged

As we mentioned before, we also took into account the total number of messages exchanged for creating and maintaining the topology. That is, the number of broadcast messages sent trying to merge trees, and the unicasts sent for the merging process. Therefore, all the messages exchanged for creating and maintaining the topology are considered. In Table 6.3, the bandwidth each device uses per second for that is shown. For estimating these data, we suppose we are dealing with IPv6, and the header of each message exchanged is 40 bytes. In our implementation of DAGRS the packets sent are empty, we just use the identifier. Therefore, we consider each message is 40 bytes. The total bandwidth used per device per second for the complete creation and maintenance of the tree during the 630 seconds of simulation is shown in Table 6.3.

We should also mention that the implementation we made for DAGRS and the configuration of the devices (transmission range) favour that the number of messages exchanged in

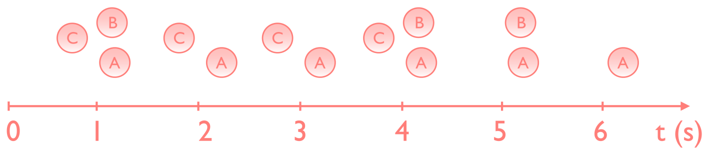
Densities	**MANET**	**Vehicular centre**	**Highway**
25 m/dev	$0.8189_{\pm 0.0119}$	$2.5504_{\pm 0.0914}$	$3.3104_{\pm 0.1380}$
15 m/dev	$0.6869_{\pm 0.0129}$	$4.7078_{\pm 0.1389}$	$5.4067_{\pm 0.1511}$
10 m/dev	$0.6748_{\pm 0.0131}$	$10.0682_{\pm 0.1981}$	$8.9581_{\pm 0.1981}$

Table 6.3: Bandwidth used per device per second (bytes).

VANETs is higher than in MANETs. The reason is that we consider a node will only try to merge its tree with a neighbour (and hence sends a broadcast message), when a change in its neighbourhood is detected or has just merged with another node. But since the transmission range for VANET is 250 m and 50 m for MANET, and also the mobility is higher in VANETs, devices in the vehicular network are much more sensitive to changes in the neighbourhood than devices in MANETs (the neighbourhood is 25 times larger), so nodes send more broadcasting messages looking for a neighbour with token. However, even considering a much larger neighbourhood, the bandwidth each device uses in the worst case is 10.0682 bytes/s. We consider this is an acceptable rate, moreover if we consider all the advantages of having this topology over a network.

6.2.4 Volatility metric

The third aspect studied to compare the feasibility of using tree topologies in VANETs is the volatility of the tree links in all the scenarios studied. We consider the volatility of the link can be measured as the total number of appearances of a link between two devices according to the total time this link is alive. The volatility varies between 0 and 1. The closer to one, the more volatile, while the closer to 0 the more stable. This idea was taken from [260], where the volatility is presented as shown in Figure 6.9.



A-B	1-1	1-1	1-1	2-2	2-3	2-3
A-C	1-1	1-2	1-3	1-4	1-4	1-4
C-B	1-1	1-1	1-1	2-2	2-2	2-2

Figure 6.9: Process for measuring the volatility of a link.

As we can see, the information initially stored for all the links is the same: first time appearing in the network (1), and also the duration of the link (1). In the second time stamp, node B disappears, so node A increases the life time of link A-C, but not the link A-B since it disappeared. In the fourth time stamp, node B appears again, so that, both node A and C increase the appearance of this link and also the life time. When the simulation time is finished, the volatility can be calculated as in Equation 6.3:

$$volatility\ of\ a\ link = \frac{number\ of\ appearances}{life\ time} \qquad (6.3)$$

For measuring the volatility of the links it is necessary to discretise the time. In our experiments, for estimating the volatility we are running DAGRS for 600 seconds and every second each device stores local information about the current links it has with its neighbours: the number of appearances and the life time. We need to store the number of times each link appears and also for how long this specific link is on.

In Table 6.4, the average of the volatility of the whole network is presented. We calculated the volatility of each link during 600 seconds for the three scenarios studied (with each of the three densities) in this work, and computed the average. We can see the values of the volatility are very low, which means the links are quite stable. In the worst case, we have value 1, which means that as soon as the link appears, it disappears. But in fact, most of the links are connected more than 1 second, and as time passes, the volatility of the link decreases quickly (a link that appears only once and is ON for 10 seconds has a volatility of 0.1). The ideal network regarding the volatility is a static one, where all the links appear in the first second and do not disappear anymore. In this case, if we consider the same simulation time, the best value of the volatility we can obtain is 1 / 600, what means 0.0016667. As we can observe in Table 6.4, the volatility of a MANET is smaller than in the vehicular centre, and this one is also smaller than in the highway. This is due to the speed the devices have in each scenario. We can see that the volatility is hardly related to the density of the network (number of devices), since the values obtained for the volatility in each environment are quite similar despite the density.

Densities	**MANET**	**Vehicular centre**	**Highway**
25 m/dev	$0.0467_{\pm 0.0704}$	$0.0679_{\pm 0.0868}$	$0.0989_{\pm 0.1121}$
15 m/dev	$0.0475_{\pm 0.0716}$	$0.0670_{\pm 0.0832}$	$0.0936_{\pm 0.1039}$
10 m/dev	$0.0483_{\pm 0.0720}$	$0.0661_{\pm 0.0802}$	$0.0921_{\pm 0.1015}$

Table 6.4: Average of the volatility of the links.

Considering that the average of the volatility does not provide a complete view of the behaviour of the topology regarding the stability along the time. We also computed the probability distribution of the volatility. We show in Figures 6.10 and 6.11 the probability of having a certain volatility. We calculate the probability distribution of the link volatility for each network. The ordinate axis represents the volatility value. It has been divided in ten intervals. The rank of each interval is 0.1. In the abscises axis the probability of having this volatility during the simulation time is represented.

As we can see in Figure 6.10 and 6.11, the same conclusion as in Table 6.4 arises: the volatility does not depend on the density of the network. The probability distribution obtained behaves similar in every density for each scenario. The probability values obtained for having a volatility lower than 0.1 are around: (1) 0.89 for the pedestrian area, (2) 0.81 for the vehicular city centre, and (3) 0.67 for the highway. If we take into account that the maximum speed in the highway is 18 times bigger than the maximum speed in the MANET, we can consider that the volatility distribution shows that the highway is a stable topology.

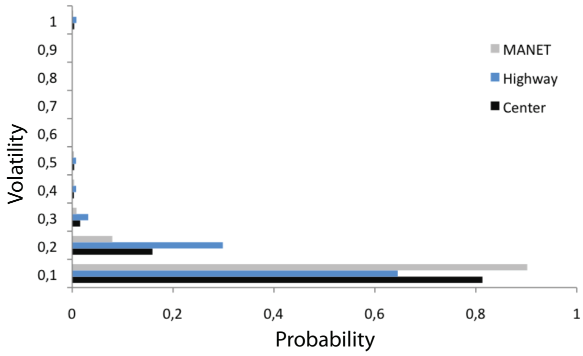
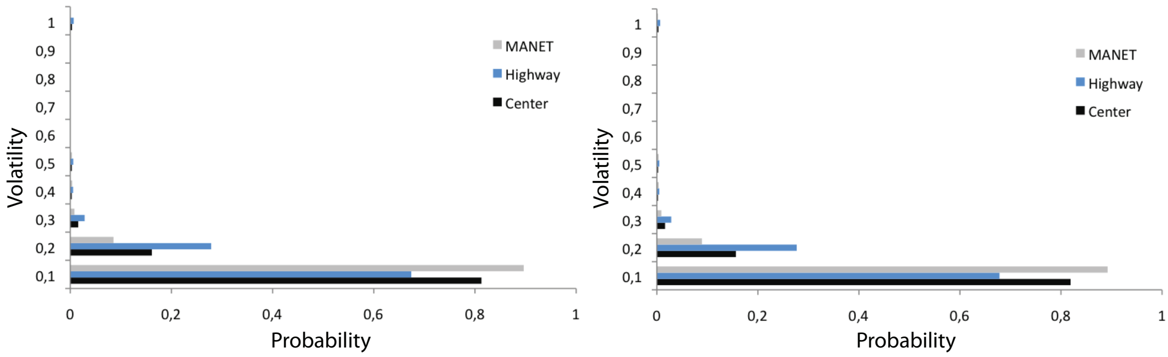


Figure 6.10: Probability distribution of the volatility in a density of 25 m per device



(a) 15 m per device

(b) 10 m per device

Figure 6.11: Probability distribution of the volatility in different densities.

6.3 The broadcasting algorithm over the tree topology, BODYF

BODYF is an efficient broadcasting protocol specifically designed for highly dynamic ad hoc communication networks, that relies on a spanning forest [282]. Our main goal is not the tree itself, but the design of a new broadcasting protocol that achieves the best possible coverage and network use at a minimum cost, using the information of the network provided by the tree-based topology. We assume that it can be already present in the network, since it is commonly used for routing purposes or any other necessity.

One of the main features of BODYF is the absence of parameters in its configuration, such that no tune is needed for the different scenarios in which it could be used. This outstanding characteristic distinguishes it from other well-known broadcasting algorithms that typically have several parameters to tune impacting in its performance [26]. Although it has no parameters, BODYF relies on a tree topology that might have some. For example, in this work we use DAGRS that depends on the *hello* interval and the speed of the token.

Assuming we have a forest topology in our network, we will use it to disseminate information using BODYF. BODYF distinguishes between two kind of neighbours in the network:

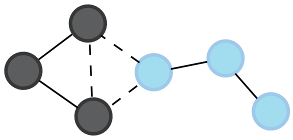


Figure 6.12: Possible types of neighbours in a spanning forest algorithm.

(1) *logical neighbours* are those ones belonging to the same tree, and (2) *potential neighbours* are those that do not belong to the same tree yet but are in communication range, as it is shown in Fig. 6.12. In this picture, devices in same colour belong to the same tree. The links between devices are represented as a continuous line if the neighbours belong to the same tree (logical neighbours), or as a dashed line if they are either in communication range but do not belong to the same tree or belonging to the same tree but not connected in order to avoid cycles (potential neighbours).

Algorithm 4: Pseudocode of BODYF

```
   Input: m: the incoming broadcast message
   Input: d: the node receiving broadcast message
   Input: s: the node which sent m
1  if m is received for the first time then
2      if s and d are logical neighbours then
3          d → forward m
4      else
5          wait until the token is received → d possesses the token
6          if d received m also from its tree then
7              d → discard m
8          else
9              d → forward m
10 else
11     d → drop m
```

The functioning of BODYF is described in Algorithm 4. We suppose that every message has a unique identifier, and when a device receives the same message more than once, it will directly discard it (no processing is done). When a device wants to spread a message, it will broadcast the information to its neighbourhood. We could think on doing a multicast only to its logical neighbours (the neighbours in the same tree), but this operation supposes the same network load as the broadcast in wireless communications (or even less if we consider the list of addresses included in the multicast header message), and the latter is allowing the potential neighbours (not belonging to the tree) to also receive the message. When a device receives the message from a logical neighbour, it will forward it only if it was received for the first time (lines 1-3), otherwise it is dropped (lines 12-14). In the case it was received from a potential neighbour, the device will keep it until it receives the token (line 5). Once the device possesses the token, it will forward the message just in the case that it did not receive the same message from a neighbour belonging to its tree during the time it was waiting for the token (lines 6-10). That is the way the message can spread through different trees, trying to avoid the dissemination of the a message more than once in the same tree. Additionally, the delay introduced when a device receives the message from a potential neighbour helps to spread it to other partitions of the network, thanks to the mobility of devices.

6.4 The Compared Protocols

We compared in this section 5 different broadcasting algorithms present in the literature: (1) DFCN [28] that decides to resend a message in terms of the neighbour knowledge. (2) SAPF [245] that adaptively regulates the rebroadcast probability in terms of the speed of the device, (3) WPB [351] that calculates the forwarding probability in terms of the distance between the source and the receiver nodes, and (4) SF [350], since typically this protocol is not able to reach different partitions to the one wherein the broadcasting process was started, but it reaches instantly all devices in the original partition. Therefore, SF will be used as a reference of the starting partition size. And finally, the already introduced BODYF.

Next, we give a brief introduction to the other protocols we are using for the sake of comparison: simple flooding (SF) [350], delayed flooding with cumulative neighbourhood (DFCN) [28], speed adaptive probabilistic flooding (SAPF) [245], and weighted p-persistent broadcasting (WPB) [351].

6.4.1 Simple flooding, SF

We decide to use SF for the comparison because it reaches instantaneously all devices inside a partition, but it is the least efficient in terms of the network resources. Moreover, it is not able to spread the message outside the partition where the source node is. Hence, it provides a good approximation to the initial partition size, and it will show the validity of the other compared protocols. In connected networks the coverage it achieves can be seen as an upper bound. SF is the most intuitive idea for disseminating a message in a network. It does not try to reduce the number of re-emissions or collisions, so it does not need any knowledge about the neighbourhood. The strategy of this algorithm is quite simple, when a device receives a message, it will forward it only once.

6.4.2 Delayed flooding with cumulative neighbourhood, DFCN

DFCN [26, 28, 147] was chosen because it was demonstrated to outperform some other well known broadcasting protocols in the literature [147, 149] as Simple Flooding, Flooding with Self-Pruning [220], the Scalable Broadcast Algorithm [256], Multipoint Relaying [270] and AHBP-EX [258], and because it was also dealing with high mobility (DFCN was used in three different scenarios shopping mall, metropolitan area and highway). DFCN was designed with the aim of minimising the network overload (by suppressing some rebroadcast). For doing that, the node sending the message embeds a list with its 1-hop neighbourhood. The receiver node will decide whether forwarding the message or not according to the *benefit* of an operation, measured in terms of the number of its 1-hop neighbours that are not included in the list (i.e., the number of nodes that most probably did not receive the message). For avoiding collisions, every device receiving a message sets a random delay (RAD). This way, two nodes in the same neighbourhood forwarding the broadcast message will probably not resend it at the same time. However, in order to promote the dissemination of the message in sparse networks, every time a new neighbour is met, *RAD* is fixed to zero. The ability to know when a network is sparse or not is directly related to the number of neighbours of a given node, and defined by *densityThreshold*. As we mentioned before, DFCN has 3 parameters to set:

- *Benefit*: is a fixed threshold in order to decide if a message should be forwarded or not.
- The *RAD*: the period of time a node waits before forwarding the message. It is a random value chosen from a fixed interval.
- The *densityThreshold*: is a threshold for considering if the network is dense or sparse. It is directly related to the 1-hop neighbourhood.

6.4.3 Speed Adaptive probabilistic flooding, SAPF

SAPF [245] is a probabilistic broadcasting algorithm that adapts the rebroadcasting probability to the speed of the current device. One of the main challenges of this dissemination algorithm is the selection of a rebroadcast probability that achieves low delay and high reachability. A high forwarding probability for emergency messages will reach more devices but it will also increase both the network resources and the delay due to the higher amount of rebroadcasting vehicles. In this work, the critical forwarding probability is related to the network density which in turn depends on the vehicle speed (low speed implies high density). For obtaining optimal performance of the broadcasting protocol at all network densities, the forwarding probability dynamically adapts to the speed of the device. Whether the speed is either too high or too slow, the probability chosen is fixed. In case low speed is detected, the fixed rebroadcast probability value is low, as there might be a traffic jam. On the contrary, if network density is low (high speed), the vehicles always rebroadcast in order to reach as many devices as possible. The pseudocode for the algorithm is shown in 5.

Algorithm 5: Pseudocode of SAPF

```
Input: v: vehicle speed
Input: m: broadcast message
if (v >= 15km )&&( v<= 100km) then
    broadcast m with probability p = 0.00557v-0.0033
else if v< 15km then
    broadcast m with probability p = 0.05
else
    broadcast m with probability p = 1
```

On one hand if the speed of the vehicle is very low, probably means there is a traffic jam, and therefore, a low broadcasting probability is used as the network is almost static and a high probability would lead to the broadcast storm problem [249]. On the other hand, when the speed is very high, the network is very fluctuant and the connectivity between devices is difficult, therefore the forwarding probability is set to 1 in order to promote a high reachability.

6.4.4 Weighed p-persistent broadcasting, WPB

WPB [351] is also a probabilistic scheme that calculates the forwarding probability in terms of the distance of the current (*node i*) to the source node (s), see Equation. 6.4.

The first time a node receives a packet sets a *WAIT_TIME*. During this *WAIT_TIME* the distance to the new sources of all the repeated messages received is calculated, and thus, the probability of resending. The smallest probability is the one selected as its forwarding probability.

$$probability_i = \frac{distance_{is}}{transmission\ range_i} \tag{6.4}$$

If the message is not rebroadcast, the node stores the message for an additional *WAIT_TIME* + δ ms, where δ is the 1-hop transmission and propagation delay, which is typically less than *WAIT_TIME*. If the node does not hear the retransmission of the message from any of its neighbours the node should rebroadcast the message with probability 1 after *WAIT_TIME* + δ ms in order to prevent message die out and guarantee 100 percent reachability. In weighted p-persistence the further the location the higher the probability of resending.

6.4.5 Experimental Setup

The experimental setup used is the one proposed in Sect. 6.2.1. Each of the presented algorithms along with BODYF are the instantiation of the broadcast module of the framework.

As we mentioned before in Sect. 6.4, it is necessary to set some parameters for the correct functioning of the some of the protocols. For DFCN, after a tuning process, these threshold values were set as follows: *benefit* = 0.4 and *RAD* $\in$ [0,7] seconds. The *densityThreshold* is different in each density:

1. a device each 25 m $\rightarrow$ *densityThreshold* = 10;
2. a device each 15 m $\rightarrow$ *densityThreshold* = 12;
3. a device each 10 m $\rightarrow$ *densityThreshold* = 15.

WPB also has some parameters that must be configured. Those are *WAIT_TIME* and δ. The former is the time set when a broadcasting message is received to allow duplicated messages to arrive to the same node. The latter is a timer that is established in case of suppressing in order to check if any neighbour sends the broadcast. In case the message of its neighbour is not heard, the node will forward it. Just as recommended in [351], both values are set to 5ms.

The experiments module remains the same as in Sect. 6.2.1. The only differences are that before broadcasting the first message, we let the network evolve for 30 seconds, and then we disseminate a message every 30 seconds during a period of 10 minutes (that means we made 20 broadcasting processes starting from the same device, but from different positions since it is moving). Therefore, the total simulation time is 630 seconds.

The monitoring tools module is in charge of measuring the number of devices in the network that received the broadcasted message (hereinafter called broadcast coverage), and also the message complexity what means the number of broadcast and unicast needed to spread the message in the network. It simulates 30 different topologies to make sure our results are reliable. The results presented in this work are the average values obtained after these simulations. Additionally, statistical analysis are also made to all the results presented in this chapter. This study has been done using the *boxplot* function from *Matlab*. In the displayed *boxplots*, the bottom and top of the boxes represent the lower and upper quartiles of the data distribution, respectively, while the line between them is the median. The whiskers are the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile. The crosses are data not included between the whiskers. Finally, the notches in the boxes display the variability of the median between samples. If the notches of two boxes are not overlapped, then it means that there is statistical significant difference in the data with 95% confidence.

6.5 Simulation Results

In this work, we evaluate and compare BODYF, DFCN, SAPF, WPB and SF in three different scenarios specially designed for Luxembourg city, but applicable to any city centre and therefore, to any mobile ad hoc network.

Two of the proposed protocols incurred in overhead during the broadcast process. DFCN inserts all the neighbours of the device which is forwarding the message, and the tree creation and maintenance of BODYF. In Sect 6.2.3, the expenses of creating and maintaining the tree were already presented.

In case of DFCN, preliminaries studies were made considering the list DFCN attaches to the broadcast message is composed by IPv6 addresses, so it is 16 bytes. We obtained that the extra bandwidth used per device during the complete broadcast process varies from 0.2783 to 0.713 kbits/device from the sparsest to the densest scenarios (please refer to [282]).

The percentage of devices reached by every protocol is also shown in Table 6.5, where it can be seen that BODYF nearly covers the whole network in the two cases dealing with vehicles using less than 32% of forwarding nodes to achieve such a good coverage. SAPF shows a very bad performance for both MANET and Highway environments. On one hand, for the former as devices move very slow, the probability of forwarding does not varies and it is always 0.05, which is a very low value to disseminate the message as it can be seen, since it just covers 6.54% of devices at maximum. On the other hand, when dealing with the highway environment the number of forwarding nodes is highly increased as all devices whose speed is higher that 100 Km/h always resend the message (behaves as SF). But for the vehicular city centre the performance shown is very competitive. WPB shows a good behaviour for both the coverage achieved and the network resources, similar to the one obtained with BODYF. DFCN is always below any protocol for both parameters measured.

In Table 6.6, we show the results obtained by the broadcasting protocols in terms of the message complexity, i.e. the number of messages sent by the protocols (network use). As these five protocols just use the local information to decide whether to resend the message or not, and the message is forwarded using one simple broadcasting message, the complexity of the message can be also seen as the number of forwarding nodes. It is possible to check how BODYF and WPB are forwarding less messages when the density increased for all the environments, but at the same time the coverage achieved is also increasing. This is a very good performance since, as the network grows, the use of network resources decreased for

% Dev. Reached	Densities	MANET	Highway	Vehicular centre
BODYF	**25 m/dev**	50.68	94.52	99.50
	15 m/dev	80.84	98.97	99.87
	10 m/dev	94.41	99.71	100
DFCN	**25 m/dev**	45.85	73.55	78.85
	15 m/dev	71.08	86.18	90.42
	10 m/dev	87.16	95.26	92.26
SF	**25 m/dev**	89.87	100	100
	15 m/dev	99.30	100	100
	10 m/dev	99.99	100	100
SAPF	**25 m/dev**	5.59	97.98	96.85
	15 m/dev	6.14	99.11	99.82
	10 m/dev	6.54	99.28	100
WPB	**25 m/dev**	55.04	92.50	98.16
	15 m/dev	83.59	98.51	99.64
	10 m/dev	95.15	99.61	99.81

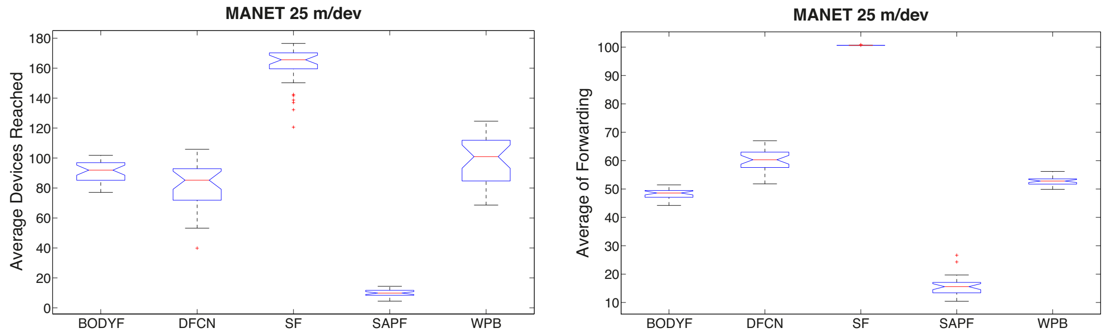
Table 6.5: Percentage of devices reached.

achieving better coverage. The values shown in this table are the percentage of forwarding nodes that after receiving the message decides to rebroadcast it. Explaining why in DFCN, in the sparsest network, the percentage of rebroadcast is 60.17% while the coverage achieved is 45.85% (more than sixty percent of the 45.85% nodes receiving the message resend). As it can be seen, the worst protocol in terms of this ratio is SF, which always sends one message per device. SAPF presents a low number of nodes rebroadcasting in the pedestrian environment but it is due to the small reachability obtained. WPB, BODYF and DFCN show a similar performance for both vehicular scenarios. This is a very important result since DFCN that was specifically designed to reduce the number of forwarding messages, and needs a long process to be tuned for being competitive, meanwhile BODYF or WPB have no tune process.

% Dev. Forwarding	Densities	MANET	Highway	Vehicular centre
BODYF	**25 m/dev**	48.20	31.57	30.04
	15 m/dev	37.95	27.75	25.88
	10 m/dev	34.73	24.28	22.04
DFCN	**25 m/dev**	60.17	26.73	27.81
	15 m/dev	56.24	24.59	26.59
	10 m/dev	52.41	23.23	26.63
SF	**25 m/dev**	100	100	100
	15 m/dev	100	100	100
	10 m/dev	100	100	100
SAPF	**25 m/dev**	15.08	76.22	30.20
	15 m/dev	10.72	75.87	30.01
	10 m/dev	8.62	75.88	29.94
WPB	**25 m/dev**	52.67	27.89	23.82
	15 m/dev	40.66	22.47	21.13
	10 m/dev	33.93	20.19	19.43

Table 6.6: Percentage of forwarding nodes.

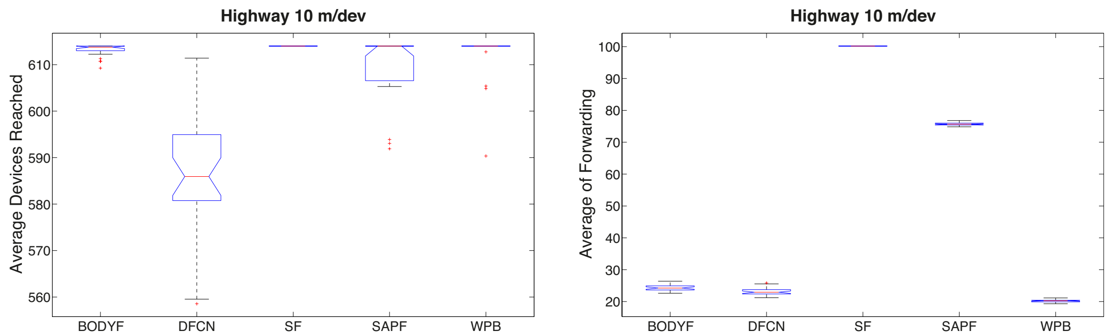
Concerning the statistical tests performed, Figure 6.13a shows that even when in average WPB obtained a higher number of devices reached than BODYF (see Table 6.5), the statistical analysis indicates that there are no significant differences between them, neither



(a) Average of devices reached

(b) Average of forwarding nodes

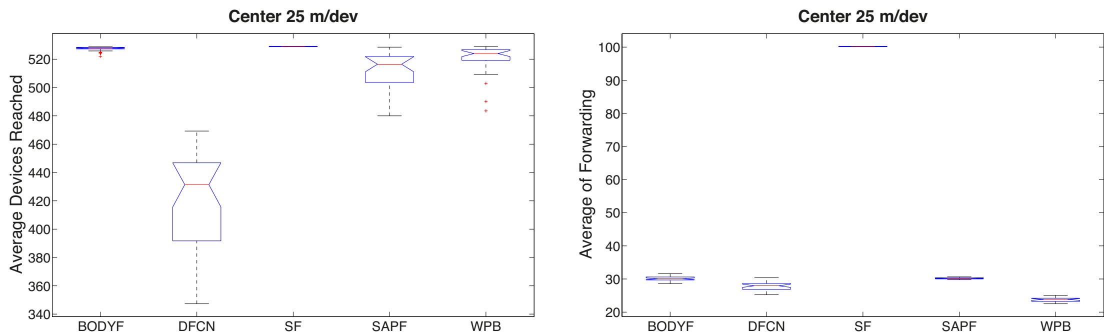
Figure 6.13: Statistical test for the MANET environment with 25 m/device.



(a) Average of devices reached

(b) Average of forwarding nodes

Figure 6.14: Statistical test for the Highway environment with 10 m/device.



(a) Average of devices reached

(b) Average of forwarding nodes

Figure 6.15: Statistical test for the vehicular city centre environment with 25 m/device

with DFCN. SF is the one with the best coverage achieved with statistical differences over all the rest. Meanwhile in Figure 6.13b, SAPF presents the best behaviour over all the other protocols, followed by BODYF that also has significant differences with WPB, DFCN and SF.

For the highway environment shown in Figure 6.14 we can see in 6.14a that, there are no statistical differences between SF, BODYF, SAPF or WPB regarding the coverage achieved. In terms of the number of nodes forwarding, in Figure 6.14b WPB shows the best behaviour with statistical significance.

In Figure 6.15, the results for the sparsest density in the vehicular city centre environment are presented. On one hand, it is possible to see how in Figure 6.15a, BODYF and SF has no statistical differences, but they two have with all the rest of protocols in terms of the number of devices reached. On the other hand, in terms of the average percentage of forwarding performs for the same network, Figure 6.15b shows that WPB behaves better than the rest with statistical difference.

We did not include more statistical *boxplots*, but we will comment the more remarkable results obtained. For the number of devices reached, DFCN presents the worst behaviour with significance differences in all the scenarios, except the MANET, were SAPF does not work at all and obtained worst statistical results. Contrary, as it was expected due to the high connectivity of the network, SF shows the best behaviour. Regarding the number of retransmissions performed, it is clear that SF is statistically the worst algorithm in all cases, followed by SAPF in both VANETs environment, but due to the bad performance in MANETs, the number of forwarding is also low, and therefore, statistically is the one using less network resources in the MANET scenario.

	Coverage	**Forwarding**
BODYF	2	2
WPB	3	1
SF	1	5
DFCN	5	3
SAPF	4	4

Table 6.7: Ranking of algorithms.

We did a ranking between each proposal in every density and scenario numbering each algorithm from 1 (the best) to 5 (the worst) according to the results obtained in the statistical tests. If there is not significance differences between two algorithms they are ranked in the same position. Summing up the values obtained we get an overall ranking for the proposal that is shown in Table 6.7. The proposal obtaining the best position in the ranking for the coverage achieved is SF as it was expected because it is an upper bound (network very connected), but the second one was BODYF. In terms of the network resources or number of forwarding performed, WPB has the first position, but again, BODYF is ranked the second one. So overall, BODYF shows a stable and reasonable behaviour for all the scenarios studied. However, among the other algorithms that do not use any underlying topology, the WPB is the best one according to our experiments.

Chapter 7

Energy Efficient Broadcasting Algorithm

Contents

Energy efficiency has attracted a lot of attention during the recent years, especially in mobile ad hoc networks where devices rely on batteries. In this kind of networks, the cooperation between devices forwarding packets or acting as a router is crucial for the network performance. Thus, reducing the energy consumption of each sending is a key feature that has been widely studied. In [195], was stated that the wireless communication could be responsible for half of the total energy consumption of a device. Therefore, considering the joint approaches of reducing the number of rebroadcast and using a variable transmission range will lead to big energy savings.

As it was already identified, our goal is to design a broadcast algorithm that uses a variable transmission range using 1-hop neighbours information. In the previous chapter, we compared different protocols: one using a tree topology and several ones not relying on any underlying topology. Among those last ones, the WPB based on the distance showed the best behaviour.

In this chapter, we are proposing an energy aware broadcasting algorithm, that using a cross-layer design decides to forward the message or not in terms of the signal strength received from the source node. It belongs to this family of protocols that uses a variable transmission range in order to save energy. Results showed that reducing the transmission power does not cause any detriment in the network connectivity, but it can also increase the performance of the dissemination process.

7.1 Cross-layer approach

Typical communication architectures are divided into layers, clearly defining the functionality of each layer and the interaction between them. Only neighbouring layers can communicate.

Wireless networks borrowed many architectures and designs from wire networks. In the current wireless networks, there exists a layered architecture with transparency between every layer, where non adjacent layers cannot communicate. For example, in wired networks transmission errors only occur when the network is congested. Therefore, reducing the bit rate as soon as transmission errors are detected and slowly recover when no more error exist, is a reasonable solution. However, this solution is no longer valid in wireless networks were packet loss can be caused by interferences, fading, collisions, etc. Reducing the bit rate might not solve the problem, but reduces the throughput and takes time to recover.

We can not just avoid the layered structure because it would lead to a *spaghetti-like* code. The solution to this problem is a cross-layer design that understands and exploits the interactions between different layers. It aims at exploiting a richer interaction between layers to achieve better performance. Using this technique it is possible to adapt node behaviours to local conditions. Cross-layer design is not only related to wireless but also to wire networks. Next, we provide a formal definition.

Definition 17 (Cross-layer design)**.**

Protocol design by the violation of a reference layered communication architecture is cross-layer design with respect to the particular layered architecture.

Definition 17 was proposed by Srivastava and Motani [313]. It only considers protocol design, whilst literature reveals cross-layer design related to applications and algorithms. For more detailed information about cross-layer design, please refer to [169].

7.2 Distance Based Broadcasting Algorithm, DB

DB is one of the different schemes proposed by [249] for minimising the effects of the broadcast storm problem when disseminating information in wireless networks. The protocol needs to know the distance between the source node and the receiver (they do not specify how to obtain this metric). The idea is that a node receiving a broadcast message for the first time will compute the distance to the source node. If this distance is short, forwarding the message does not significantly add to the coverage, thus, the message is not rebroadcast. This is shown in Figure 7.1, where the extra coverage of a close node does not worth the cost of resending.

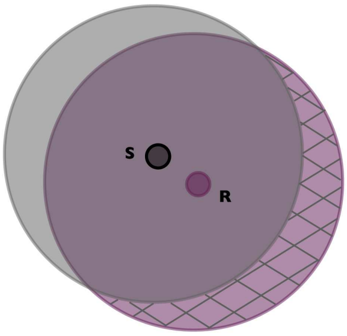


Figure 7.1: Extra Coverage

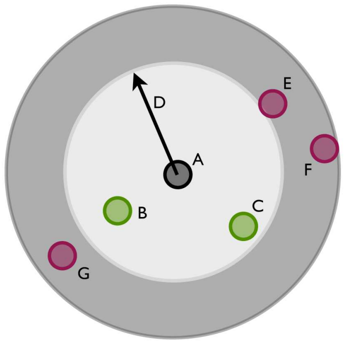


Figure 7.2: DB mechanism

As stated in [249], the distance from source node to the receiver is clearly related to the additional coverage obtained in case of forwarding, so it can be used as a metric. Only nodes that are separated at least a minimum distance from the source node resend the message. This minimum distance is a predefined threshold, D. Moreover, the protocol includes a delay before forwarding a received message, and if the same message is heard more than once (during this waiting time), the delay is cancelled, the distance between the current node and the new sender is calculated and if it is larger than D, the message is resent.

Algorithm 6: Pseudocode of DB.

```
Data: m: the incoming broadcast message.
Data: r: the node receiving broadcast message.
Data: s: the node that sent m.
Data: d: the distance between r and s.
Data: dmin: the minimum distance between r and any s.
Data: D: the distance threshold.
if m is received for the first time then
    calculate d;
    update dmin;
    if dmin < D then
        r → drop message m;
    else
        waiting = true;
        wait random number of slots;
else if waiting then
    calculate d;
    resume waiting;
    waiting = false;
    if d < dmin then
        update dmin;
else
    r → drop message m;
if dmin < D then
    r → drop message m;
else
    transmit m;
```

The behaviour of the algorithm is represented in Figure 7.2. Considering node A broadcasts a message m, nodes B and C do not resend m because the distance from those nodes to A is smaller than D. Nodes E, F and G wait for a random number of slots. If node F finishes the waiting time first, it forwards the message and, thus, node E hears it and calculates the distance to node F. As the distance between E and F is smaller than D, node E drops the packet. The pseudocode of the protocol is shown in Algorithm 6.

7.3 Enhanced distance based broadcasting algorithm, EDB

In the minimum energy broadcast problem, every node is able to adjust the transmission range in order to reduce the power consumption of the dissemination process while still guaranteeing full coverage in the network. In a real scenario with obstacles, devices moving, fading, path loss, packet loss, etc. guaranteeing the full coverage might be very ambitious or impossible, and in some cases even unnecessary. In safety control or important messages, it might be worth the overhead needed for having a reliable broadcast algorithm. But for all the other messages (info or ads), it would rather be more efficient to consider the possibility of not guaranteeing full coverage, and thus, saving all the overhead derived from acknowledgements, retransmissions, etc. In this work, we consider this second family of protocols, where full coverage is not required.

When it is necessary to calculate the distance between a source and a destination, the most common technique is either assuming a GPS service or considering the signal strength. In this work, we do not assume that all devices in our network must provide a GPS service. Therefore, we use the signal strength of the received packets.

There are some approaches where a propagation path loss model for the channel is selected and the distance between the nodes is estimated according to this model. [59] considered free space propagation model, and used that equation to calculate the distance thanks to the reception power.

In this work, we propose the *enhanced distance based broadcasting algorithm*, EDB hereinafter, that does not assume any specific propagation path loss model. The received power is related to the distance, although we are not interested in the distance itself but in the energy lost during the transmission. Considering this, a device close to the source node but with some buildings around (weakening the signal), will forward the message contributing to the process in an area where the dissemination of the message is not easy. Using this implementation we are also aware of the non perfect shape (generally considered as a disk) of the transmission range, that generally depends on the environment. So for us, the threshold D is not in terms of distance (meters) but power (dBm). This threshold D is called the *borders_Threshold* as it defines the nodes that are considered to be far from the source and therefore, close to the border of the transmission range.

Every device sends a beacon to alert devices within range about their presence. A device receiving these beacons is able to keep track of all neighbours around.

We are considering here a cross-layer design where the physical layer informs the upper layers about the received signal strength of each beacon and message received. In this situation, the algorithm is able to take decisions depending on these values. When a broadcast message is sent, the receiver will check the reception power, if it is below the *borders_Threshold*, it will consider itself as a border node (located in the forwarding area and thus, candidate for resending the message), and set the delay.

7.3.1 Enhancements

We are adding the variable transmission power capability to the devices, as well as studying different delay techniques to find out the most accurate for the algorithm.

Reducing transmission power

In any wireless transmission, as the electromagnetic wave propagates through the space, the power of the signal suffers from path loss attenuation causing a reduction in the signal strength. The relation between the transmitted power and the power finally received at the destination directly depends on the loss suffered during the transmission. Equation 7.1 represents the relation in terms of dB.

$$receivedPower = transmittedPower - loss \tag{7.1}$$

If we assume that all nodes send beacons with the same transmission power, a node receiving a beacon will be able to estimate the loss that packet suffered during its transmission, using the reception power detected at the physical layer.

Every node keeps and updates the reception power of each of its neighbours in a list, so that, when a device wants to send a broadcast message, it will be able to estimate the loss the packet will suffer (we assume a packet traversing in a direction will experiment the same loss as another traversing in the opposite direction).

If a node can estimate the loss the packet is going to suffer, it will be able to reduce its transmission power and use only the necessary one to get the furthest one hop neighbour. Thus, reducing the transmission power for sending the broadcast message provides some reduction in the energy consumption of the device, without degrading the performance of the broadcasting process (we do not consider loosing the connection with any 1-hop neighbour).

When the loss the packet suffered due to the propagation is calculated, the node can estimate the transmission power needed to reach the furthest neighbour in the one hop neighbourhood. If we are reducing the transmission power, the furthest node is receiving the packet with the minimum possible reception power allowed to correctly decode the message. This is the sensitivity of the device and we call it the *end_Threshold*. The transmission power needed can be calculated as shown in Equation 7.2.

$$transmissionPower = loss + end\_Threshold \tag{7.2}$$

Once the needed transmission power is estimated, in terms of the reception energy stored using the beacons, it is necessary to consider that the devices do move and the information can be out of date since beacons are sent every 1 second. Therefore, we are considering a margin of error (*margin_Forwarding*) that is added to the estimated transmission power. This value was experimentally chosen and was estimated considering the loss a packet at the border of the transmission range might suffer when the node moves for 1 second (as beacons are sent every second). Its value is 0.5 dBm. Therefore the new transmission power is shown in Equation 7.3.

$$newTransmissionPower = transmissionPower + margin\_Threshold \tag{7.3}$$

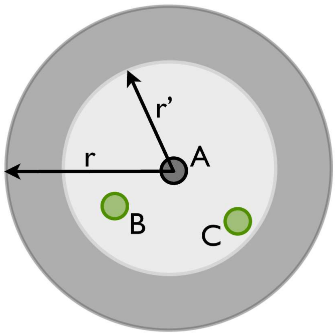


Figure 7.3: Reducing the transmission power (thus, the transmission range) of the node.

If it is less than the default transmission power, we reduce it in order to save energy, otherwise we use the default transmission power. This is shown in Figure 7.3, where it is possible to see that all the one hop neighbours are close to the source. Therefore, reducing the transmission range from r to r' decreases the energy consumption with no detriment of the network connectivity or the broadcast performance. It not only improves the energy consumption in wireless networks but also reduces the interference level of devices in a close area.

We can observe on the left side of Figure 7.4 that both nodes A and B use the default transmission range r. In this situation node C is not in range either with node A or B but it suffers from their interference. However, if both nodes A and B reduce their transmission range to the maximum needed (r') to reach the furthest node in the one hop neighbourhood, as it is shown on the right side of Figure 7.4, node C will not receive anything from A or B, and thus, the interference level will not affect C or at least it will be reduced.

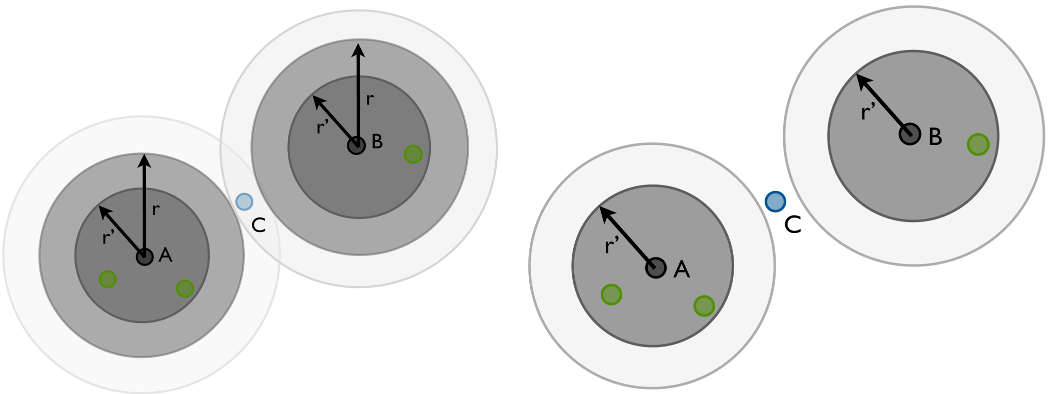


Figure 7.4: Reducing the interference level.

According to [180], the transmission power affects many aspects of the network, e.g. the transmission range, and thus, the connectivity of the network (the lower the power, the smaller the transmission range), the performance of the medium access, since it depends on the number of nodes within range, the capacity of the network, etc. According to them, the capacity is increased as the transmission power level is reduced by decreasing the interference area (it is proportional to the square of the transmission range). The network connectivity in this case is not decreased, since nodes try to reduce the transmission power but considering all neighbours in range. Neither the contention for the medium access is, exactly for the same reason explained before.

Reducing the transmission power not only reduces the energy consumption of the device but also helps in the dissemination of the message when dealing with distance based broadcasting algorithms. In DB, if the source node only has one neighbour and it is not located in the forwarding area the message will not be rebroadcast, even if that node has many other neighbours around, thus, the dissemination process is stopped. A graphical explanation is provided next. On the one hand, we can see in Figure 7.5 that node B is not in the forwarding area of node A, therefore, the message will not be forwarded and the dissemination is finished even if many nodes did not receive the message. On the other hand, if node A is reducing the transmission power so that the furthest neighbour (B) is reached, node B will be in the forwarding area, thus, it will resend the message and the rest of nodes will eventually receive it.

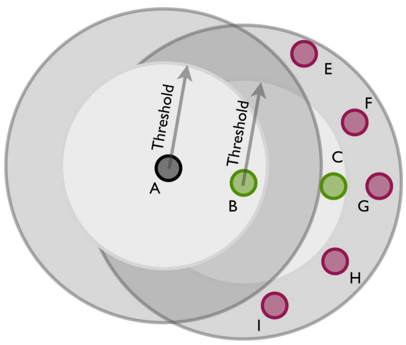


Figure 7.5: Advantage of reducing the transmission range.

Using different delay techniques

In the original implementation, DB stops the random delay when a repeated message is heard. Then, if the distance from the new source node is smaller than the threshold D, the message is discarded and no retransmission is performed. Otherwise, the forwarding starts.

In our implementation, we consider keeping track of the received energy and continue the delay, instead of stopping the delay when a repeated message is heard. Once it is finished, the forwarding decision is taken according to the received signal strength of each of the copies heard of the same message.

In DB, the delay is randomly chosen from a predefined interval. We also propose to adopt a similar scheme as the one presented by [50], where the delay is fixed and inversely proportional to the distance between the receiver and the source node. In this situation, a node closer to the source will rebroadcast later than a node far from it. We propose two approaches and compared them to random, originally proposed by DB:

1. the first proposal considers a fixed delay inversely proportional to the received power.

Algorithm 7: Pseudocode of EDB.

```
Data: m: the incoming broadcast message.
Data: r: the node receiving broadcast message.
Data: s: the node that sent m.
Data: p: the received signal strength of m sent by s.
Data: pmin: the minimum signal strength received for m from any s.
Data: borders_Threshold: the signal strength threshold.
if m is received for the first time then
    calculate p;
    update pmin;
    if pmin > borders_Threshold then
        r → drop message m;
    else
        waiting = true;
        wait time RandomDelay/FixedDelay/PowerDelay;
else if waiting then
    calculate p;
    if p > pmin then
        update pmin;
else
    r → drop message m;
if pmin > borders_Threshold then
    r → drop message m;
else
    estimate power to reach furthest neighbour;
    transmit m;
waiting = false;
```

The process for calculating the delay is shown in Equation 7.4:

$$powerDelay = \frac{-1}{rxPower - borders\_Threshold - 1} \tag{7.4}$$

If a node is setting a delay, it means, the node is a *border node*, otherwise the node is not considered as candidate to forward the message, and therefore, no delay is set. All *border nodes* receive the message with a reception power that can vary between the *borders_Threshold* and the *end_Threshold*. Therefore, considering Equation 7.4, we can check that the delay varies between 0.167 and 1 second. The higher reception power (closer neighbours), the longer the delay, and vice versa;

2. the second proposal considers a random delay chosen from an interval whose size also varies with the reception power. That is, the waiting time will be chosen between [0, *powerDelay*] calculated as in Equation 7.4.

We are comparing different techniques: (1) *RandomDelay*: the delay is chosen randomly from the interval [0 1] s; (2) *FixedDelay*: the delay is fixed with the value *powerDelay*; and finally (3) *PowerDelay*: the delay is chosen randomly from the interval [0 *powerDelay*] s. The pseudocode of the *enhanced distance based* broadcasting protocol proposed in this work is shown in Algorithm 6.

We can clearly differentiate different parameters in the proposed broadcasting algorithm:

- The value of the *borders_Threshold* sets the size of the forwarding area. The higher the threshold, the higher the number of potential forwarders, the coverage, the network resources used, and the number of collisions.

- The *margin_Forwarding* is related to both the energy saved and the coverage achieved. It is the extra amount of energy added to the estimated transmission power. The higher the margin value, the higher the coverage reached as well as the energy used.

- The value of the *delay* interval sets the waiting time and also affects the behaviour of the protocol. We split this threshold into two different variables: *minimum delay* and *maximum delay* referring to the lower and upper value of the interval, respectively. If the delay is very high, the time used to spread the message will be high, but if it is very small, the number of collisions will probably increase.

7.3.2 Experimental setup

As already mentioned in Chapter 5 and as it was done in Chapter 6, for the evaluation and validation steps in the development process of the communication algorithm, the optimisation algorithm module is not used. The modules that are used are the ones shown in Figure 7.6.

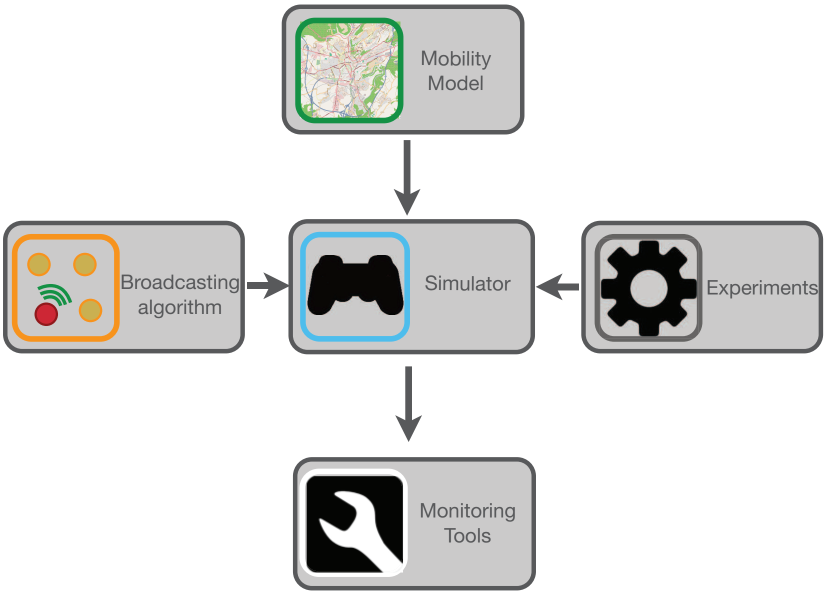


Figure 7.6: The instantiation of the experimental framework

The instantiation used in this chapter for the simulator module of our framework is the *ns-3 simulator*. The ns-3 simulator was proposed in [200]. The principal concern of the project is the realism of the simulator. It only relies on C++, for the library and the simulation scenarios (although python bindings exist). This helps producing code easier to debug. The network layers are very realistic, specially the wireless layers: real IP addresses are used, sockets resemble real BSD sockets, packet are real network big endian byte arrays. This permits both to easily execute native code and to do emulation. Moreover, ns-3 is very accurate in emulating the wireless technology.

For the mobility model module of the framework, we are using the *random walk* also known as *brownian motion mobility model* presented in [128]. Nodes move with a speed and direction randomly chosen during a fixed amount of time (we are considering 20 seconds, known as *time moving* hereinafter). After, new random speed and direction are chosen. If a node hits one of the boundaries of the area, it rebounds on the boundary with a reflexive angle and speed. The speed of the nodes that can vary from 0 to 2 m/s (between 0 and 7.2 Km/h).

The above mention modules (simulator and mobility model) will be used hereinafter for the reminder of this thesis.

The experiments module is composed of the simulation environment used that is a square area of 2000 m size (4 Km^2). We measure our experiments with different network densities: the number of nodes varies from 100 up to 1000 in steps of 100 devices. We do not consider a naive or ideal communication channel but consider collisions and a propagation path loss model: *log distance path loss model* [275]. In the simulations, the network evolves for 30 seconds in order to have the nodes uniformly distributed in the area. Then, after these 30 seconds, a node starts the broadcasting process. The simulation stops after 40 seconds. In Table 7.1, we present a summary with the configuration we are using for the simulations.

The value for the *borders_Threshold* parameter is *-90 dBm*. This value was experimentally chosen and represents one third of the total transmission range. In Figure 7.7, we can see how the value of the *borders_Threshold* influences both the coverage reached and the number of forwarding nodes. We can see that setting a higher value of the *borders_Threshold*, gives a high value of coverage but at the expenses of a high number of nodes rebroadcasting the message. The highest difference between the two graphs is found at -90 dBm (82.64% of coverage and 34.24% forwarding nodes).

We also consider that the sensitivity of a node is *-95 dBm*, this is called the *end_Threshold*. Therefore, all nodes receiving a message whose reception energy varies from [-95, -90] dBm (forwarding area) are candidates of forwarding the broadcast message.

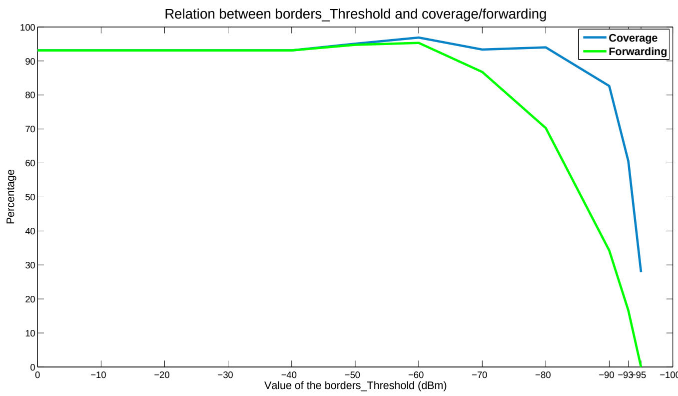


Figure 7.7: Influence of the borders_Threshold over the reachability and the network resources

As we explained before, we are comparing different variants of EDB to DB. In Table 7.2, all the different proposals are explained.

In order to evaluate the performance of EDB and validate the proposed enhancements, we are comparing the original DB to EDB in terms of different parameters. The quality

Number of devices	100-1000	*borders_Threshold*	-90 dBm
Speed	[0, 2] m/s	*margin_Forwarding*	0.5 dBm
Size of the area	2000 m × 2000 m	*end_Threshold*	-95 dBm
Transmission power	16.02 dBm	*delay interval*	[0, 1] s
Direction change	every 20 s		

Table 7.1: Parameterisation used

DB	original DB with random delay ∈ [0, 1]
StopRandomDelayEDB	EDB with random delay ∈ [0, 1]
StopFixedDelayEDB	EDB with fixed delay = *powerDelay*
StopPowerDelayEDB	EDB with delay ∈ [0, *powerDelay*]
NonStopRandomDelayEDB	EDB with random delay ∈ [0, 1] & not stop delay
NonStopFixedDelayEDB	EDB with fixed delay = *powerDelay* & not stop delay
NonStopPowerDelayEDB	EDB delay ∈ [0, *powerDelay*] & not stop delay

Table 7.2: Different variants of DB and EDB

of the performance of a broadcasting algorithm in ad hoc networks is usually related to some standard measurements. The aspects considered at the monitoring tools module of the framework are the most common ones in this kind of protocols:

1. the **coverage** obtained, i.e., the number of devices that, after the dissemination process, receive the broadcast message;

2. the **energy used** by the broadcast process, measured as the sum of the energy every device consumes to forward the message;

3. the number of **forwardings**, considered as the number of nodes that decide to resend the broadcast message;

4. and the **broadcast time**, considered as the time needed to spread a message in the network, since the source node sends the message until the last node receives it.

Additionally, we are also analysing the **the number of collisions** due to the broadcast process, in order to see the effect of the different techniques for the delay.

For obtaining reliable results the monitoring tools module executes 100 different independent topologies for each of the different densities we are using. Moreover, It makes statistical analysis to all the results presented in this chapter. This statistical study was done using either the Anova or the Kruskal-Wallis tests, depending on whether the data follow a normal distribution or not. In order to know if the data follow it, we apply the Kolmogorov-Smirnov test.

7.3.3 Simulation results

In order to highlight the statistical results obtained, we are including a background colour to show significant differences in the results obtained. In light grey colour, the proposal that shows the worst performance with significant differences all over the others is marked, and in dark grey colour, the one with, statistically, the best behaviour. In bold font, the highest average value is shown. In tables, the last row shows whether there are statistical differences between any variant in each density. A + for the p-value means there are significant differences at least between two algorithms.

Collisions

We consider a collision can be produced due to the reception of a packet when the device is already synchronised or transmitting. In Table 7.3, the number of collisions produced only due to the broadcasting process is shown for each density and the two variants of algorithms (DB and EDB). The values presented there represent the number of devices detecting the collisions, that is, for example, in a neighbourhood composed of 10 devices (all in range), if a message is sent while another node is still transmitting, all devices will detect the collision (it is not one collision but 10 devices detecting a collision).

Variants of Protocols	100	200	300	400	500	600	700	800	900	1000
DB	0.09 ±0.73	1.01 ±4.31	16.60 ±25.11	101.98 ±85.64	391.55 ±183.51	810.55 ±275.51	1428.18 ±416.31	2384.31 ±640.28	3857.89 ±959.43	5768.80 ±1401.34
NonStopRandomDelayEDB	**0.00** ±0.00	0.22 ±1.41	1.55 ±3.98	**5.30** ±7.79	**19.03** ±19.73	**38.92** ±28.06	**69.96** ±48.85	**114.59** ±57.46	193.84 ±74.82	**238.81** ±99.41
StopRandomDelayEDB	0.08 ±0.56	1.02 ±4.62	23.54 ±32.88	128.26 ±96.66	367.17 ±147.08	812.67 ±289.21	1603.83 ±434.92	2673.45 ±652.44	4126.23 ±909.85	6117.96 ±1804.76
NonStopFixedDelayEDB	**0.00** ±0.00	0.09 ±0.64	1.53 ±4.18	9.62 ±11.57	27.86 ±23.97	55.02 ±34.73	100.17 ±52.64	143.55 ±58.73	238.63 ±109.92	307.59 ±126.12
StopFixedDelayEDB	0.04 ±0.40	2.67 ±7.59	16.85 ±23.33	116.72 ±90.84	421.88 ±179.17	932.97 ±275.46	1644.51 ±395.78	2697.93 ±752.37	4483.52 ±1254.70	6563.00 ±1818.84
NonStopPowerDelayEDB	**0.00** ±0.00	**0.07** ±0.50	1.33 ±3.55	17.92 ±22.78	46.76 ±36.22	144.35 ±72.01	230.86 ±107.36	365.85 ±148.45	503.95 ±177.88	661.81 ±224.86
StopPowerDelayEDB	0.14 ±1.00	1.64 ±5.25	17.43 ±28.63	151.34 ±113.40	426.24 ±211.19	960.23 ±318.92	1769.64 ±444.99	3121.64 ±868.90	5069.38 ±1450.58	8159.67 ±2443.67
p-value	-	+	+	+	+	+	+	+	+	+

Table 7.3: Average number of collisions in the dissemination process

It is possible to check that in the sparse densities the number of collisions is very low, i.e. with a network composed of 100 devices the maximum average number of collisions is 0.14 for any protocol. We must remark the big difference between stoping the waiting time or not (no matter which value of the delay is set).

In all protocols that do not stop the delay when a copy of the message is heard, the number of collisions is much lower than in the ones stopping. That was an expected result as devices in a close area receive a duplicate copy of the message almost at the same time, and therefore, they stop the delay and forward the message (if this is the case) at the same moment. For example, the original DB (*DB*) in a network of 1000 devices has an average number of collisions of 5768.80 while the same configuration just finishing the delay and reducing the transmission power (*NonStopRandomDelayEDB*) detects less number of collisions 238.81 (95.68%).

The variant that chooses a random delay is the one detecting less number of collisions. That is logical because the value that each device sets for the delay is randomly chosen, meanwhile in the other cases, it depends on the distance between the source and the current node. Meaning that, devices nearby will have a similar value. Moreover, in the *NonStopPowerDelayEDB* variant the number of collisions is higher than in the others because the value of the delay is randomly chosen between [0, *powerDelay*]. In this case, the maximum possible value is *powerDelay*, which is smaller as the neighbours are closer to the limit transmission range (0.167 seconds). So that, for dense networks where the probability of nodes close to the limit transmission range is high, nodes will have a small value for the delay, increasing therefore, the probability of collisions.

The statistical tests have been applied to all the variants of the protocols and all the densities for comparing the average number of collisions. Results showed that for 100 devices there is no statistical differences in any case. For 200 devices, both variants of *StopFixedDelay* are worse than all the *NonStopDelay* ones. For a network of 300 devices, all the different proposals of *StopDelay* are statistically worse than the *NonStopDelay*, and this behaviour is also presented for all the others densities. For 500 up to 1000 devices, not only the *StopDelay* variants are worse than the *NonStopDelay* ones, but also *NonStopRandomDelay* is statistically better than *NonStopPowerDelay*.

From these first results showing the difference in the performance between stopping the delay or not when a duplicate copy of the message is heard we can consider the possibility of discarding all the stopping variants from the comparison. For a more detailed study considering these variants and the same variants for DB, please refer to [281].

Energy

The average energy used by each node to rebroadcast a message is shown in 7.4

Variants of Protocols	100	200	300	400	500	600	700	800	900	1000
DB	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0	40.00 ±0.0
NonStopRandomDelayEDB	24.12 ±12.87	**31.35** ±7.47	**34.04** ±4.32	**36.30** ±2.33	**37.39** ±0.46	**38.03** ±0.42	**38.46** ±0.29	**38.81** ±0.29	**39.07** ±0.22	**39.27** ±0.18
NonStopPowerDelayEDB	**21.11** ±13.03	31.98 ±6.26	34.38 ±4.57	36.50 ±0.93	37.48 ±0.69	38.09 ±0.40	38.57 ±0.30	38.96 ±0.25	39.15 ±0.21	39.35 ±0.16
NonStopFixedDelayEDB	23.30 ±12.81	31.36 ±7.74	34.99 ±4.22	36.66 ±1.69	37.60 ±1.93	38.39 ±0.32	38.81 ±0.28	39.13 ±0.22	39.28 ±0.48	39.46 ±0.16
p-value	+	+	+	+	+	+	+	+	+	+

Table 7.4: Average of the energy used per forwarding message in mWatts

The original DB protocol consumes 40 mW (16.02 dBm), the default transmission power. According to the results obtained from the study of the number of collisions, we are only considering the variants of EDB that do not stop the delay. In general, the best performance is found in the *NonStopRandomDelayEDB*. It reduces the energy from 1.35% for the densest network up to 60.3% for the 100 devices network. In bold font, the proposal with the lowest average value of the energy used is marked. We can see that, generally, EDB consumes less.

For all densities *DB* showed the worst performance with significant differences (marked with light grey background). Considering the behaviour between the other three variants no statistical differences where shown for the sparsest networks (up to 300 devices). From 400 up to 1000 devices *NonStopFixedDelayEDB* always showed statistically worse results than either *NonStopRandomDelayEDB* or *NonStopPowerDelayEDB*.

Coverage

In terms of the coverage achieved by the broadcasting process, Table 7.5 shows the average number of devices reached for each density after 100 runs for the different variants. As we mention before, the results shown correspond to the variants that do not stop the random delay. We compare them with the original DB (*DB*).

Variants of Protocols	100	200	300	400	500	600	700	800	900	1000
DB	3.35 ±3.39	17.40 ±16.44	89.50 ±70.48	265.56 ±122.31	436.96 ±114.12	574.36 ±84.40	693.86 ±6.01	795.14 ±5.60	896.63 ±3.42	997.32 ±2.91
NonStopRandomDelayEDB	3.94 ±3.27	**23.88** ±22.61	92.46 ±69.00	297.35 ±108.89	**472.19** ±30.71	584.07 ±57.76	694.30 ±6.02	**796.28** ±3.96	**897.75** ±2.77	997.74 ±3.03
NonStopPowerDelayEDB	**4.08** ±3.91	23.70 ±22.06	**111.28** ±86.13	**299.64** ±109.39	443.03 ±123.87	**590.82** ±8.69	688.46 ±63.17	796.03 ±4.96	897.42 ±2.69	998.01 ±1.99
NonStopFixedDelayEDB	3.74 ±3.37	21.34 ±21.64	107.13 ±82.35	282.19 ±118.41	457.67 ±96.79	586.21 ±20.86	**694.60** ±7.31	796.04 ±4.55	888.40 ±88.97	**998.12** ±1.87
p-value	-	-	-	+	+	-	-	-	+	+

Table 7.5: Average of devices reached in the dissemination process

The coverage achieved by the broadcasting process is very good with reasonable use of the network resources. For the 100, 200 and 300 densities, the network is very sparse and the coverage achieved is very low (the minimum for 100 devices is 3.35%), but as the density grows, the coverage also increases. In networks with 600 and 700 devices, the minimum coverage achieved between all protocols is 95.72% and 98.35%, respectively. For networks with density equal or higher to 800 the minimum coverage achieved is higher than 98.5%, being 99.39%, 98.71% and 99.73% for 800, 900, 1000 devices respectively.

Statistical tests have been applied to the results. There are only statistical differences for 4 network densities 400, 500, 900 and 1000 devices. In 400 devices density, *DB* behaves statistically worse than *NonStopPowerDelayEDB*. For 500 devices, *DB* is significantly worse than both *NonStopPowerDelayEDB* and *NonStopFixedDelayEDB*, meanwhile it is worse than *NonStopRandomDelayEDB* and *NonStopPowerDelayEDB* for 900 network density. For the last configuration, with 1000 nodes, *DB* reaches significantly less coverage than both, *NonStopRandomDelayEDB* and *NonStopFixedDelayEDB*.

These results lead to an important conclusion: reducing the transmission power for disseminating a message does not decrease the number of devices that finally receives the dissemination message. Moreover, as explained in Section 7.3.1, there are some cases in which reducing the transmission power promotes the dissemination process and increases the coverage. This happens when the source node does not have neighbours in the forwarding area, thus, no rebroadcasting is performed. However, when reducing the transmission power to reach the furthest neighbour, there is at least one node (the furthest), that is in the forwarding area and thus, the message is forwarded.

Network Usage

The percentage of nodes that after receiving the dissemination message forward it is presented in Table 7.6. There, it is possible to see that from 500 devices and on, the number of forwarding nodes is decreasing as the network density increases. The protocol that uses the lowest average number of forwarding node is in bold font.

Variants of Protocols	100	200	300	400	500	600	700	800	900	1000
DB	**1.59** ±2.15	**3.82** ±4.05	**11.98** ±10.02	**24.36** ±11.65	28.87 ±7.86	28.98 ±4.55	28.26 ±1.30	27.19 ±1.43	26.33 ±1.26	25.56 ±1.12
NonStopRandomDelayEDB	2.37 ±2.30	5.88 ±5.97	12.78 ±9.91	26.80 ±10.18	29.48 ±2.56	26.79 ±2.85	24.31 ±0.70	22.04 ±0.59	20.03 ±0.43	18.43 ±0.46
NonStopPowerDelayEDB	2.43 ±2.39	5.92 ±5.71	15.64 ±12.40	26.99 ±10.31	**27.38** ±7.85	26.93 ±0.95	23.97 ±2.37	21.69 ±0.61	19.80 ±0.48	18.17 ±0.41
NonStopFixedDelayEDB	2.32 ±2.01	5.32 ±5.82	14.88 ±12.03	25.09 ±10.69	28.21 ±6.18	**26.28** ±1.38	**23.91** ±0.71	**21.52** ±0.61	**19.45** ±2.02	**18.05** ±0.44
p-value	+	+	-	-	+	+	+	+	+	+

Table 7.6: Average of forwarding nodes

On the one hand, when the network is sparse (up to 500 devices), around 30% of forwarding nodes are needed. On the other hand, in the densest network less than 19% is necessary for the energy aware proposals, and 25.56% for the original DB. The reduction in the percentage of the number of nodes rebroadcasting in the densest network is at least 7.13%.

Statistical tests showed that for 100 devices, *DB* is the best one, and that in 200 devices the original DB outperforms *NonStopRandomDelayEDB*. In 300 and 400 device networks, no statistical differences were found between any protocol. From 500 devices on, (*DB*) is always worse in terms of the percentage of forwarding nodes with statistical differences over all the other variants. Moreover, for 600 devices *NonStopFixedDelayEDB* is the best one with significant difference. In 700 and 900 devices network *NonStopFixedDelayEDB* outperforms with statistical confidence *NonStopRandomDelayEDB*, meanwhile for 800 and 1000 is worse than both *NonStopFixedDelayEDB* and *NonStopPowerDelayEDB*.

Summarising all the results obtained, we can say that all the variants that do not stop the delay upon the reception of a duplicate message have a better performance in terms of number of collisions than the ones that do stop the delay. Moreover, between the variants that do not stop the delay, *NonStopRandomDelayEDB* outperforms with statistical confidence *NonStopPowerDelayEDB*. For the coverage achieved there is not an algorithm that generally behaves better. Indeed, *DB* is never the best algorithm, what means that reducing the transmission power does not decrease the performance of the broadcasting process. Regarding the energy used per forwarded message, *NonStopRandomDelayEDB* has the lowest average values and along with the *NonStopFixedDelayEDB* are statistically the best ones. In the case of the number of the percentage of rebroadcast messages, *NonStopFixedDelayEDB* is the one that, for dense networks, generally behaves better in average.

Analysing different values of the thresholds

Broadly speaking, as a result of all these studies, we consider that *NonStopRandomDelayEDB* is the variant that, generally, provides better results. Only when considering the percentage of retransmitted messages, *NonStopFixedDelayEDB* performs better results with statistical differences for 5 network densities than *NonStopRandomDelayEDB*, and also *NonStopPowerDelayEDB* for two of them. But, we must remark this difference is less than 1% in the case of *NonStopRandomDelayEDB*.

Therefore, we extend the experiments for this protocol variant: *NonStopRandomDe-*

layEDB, studying different values for the thresholds. The different configurations of the protocol are shown in Table 7.7.

Protocol variants	*Delay interval*	*borders_Threshold*	*margin_Forwarding*
NonStopRandomDelayEDB 01_90_0.5	[0, 1]	-90	0.5
NonStopRandomDelayEDB 01_90_1	[0, 1]	-90	1
NonStopRandomDelayEDB 01_92_0.5	[0, 1]	-92	0.5
NonStopRandomDelayEDB 02_90_0.5	[0, 2]	-90	0.5
NonStopRandomDelayEDB 02_92_1	[0, 2]	-92	1

Table 7.7: Variants of *NonStopRandomDelayEDB* with different values of the thresholds

In Figure 7.8, the average energy consumption (in mWatts) of *NonStopRandomDelayEDB* per forwarded message is shown for all the different densities using different values of the thresholds. The amount of energy used is increasing up to a moment where it remains almost constant (little reduction is performed) with a value close to the default transmission power. This is due to the high density of nodes in the network, as the possibility of having a neighbour close to the limit transmission range is high, and therefore, no energy reduction is performed. As expected, in the cases where *margin_Forwarding* is set to 0.5 dBm, the energy consumed per forwarded message is lower than the others approaches.

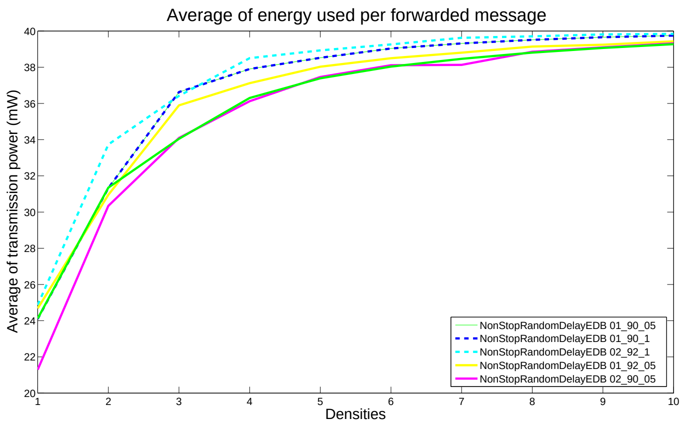


Figure 7.8: Average of the energy used in each forwarding of the broadcasting process.

The coverage achieved is shown, in Figure 7.9. It is clearly lower in the case of the *NonStopRandomDelayEDB 02_92_1* and *NonStopRandomDelayEDB 01_92_0.5*, both configurations that set the *borders_Threshold* to -92 dBm. That is normal, as the lower this threshold, the smaller the forwarding area, and thus, the lower the number of potential forwarding nodes. From these results, we realised that the coverage achieved does not depend on the value of the *margin_Forwarding*, as there is almost no difference between *NonStopRandomDelayEDB 01_90_05* and *NonStopRandomDelayEDB 01_90_1*. In fact, this threshold was calculated to prevent the possible movements of nodes during 1 second (the periodicity of the beacons), so increasing it, should not provide extra coverage (as we can observe from Figure 7.9).

In Figure 7.10, the time necessary to broadcast a message in the network is shown. The behaviour of the different configurations is similar: in very sparse networks the broadcast

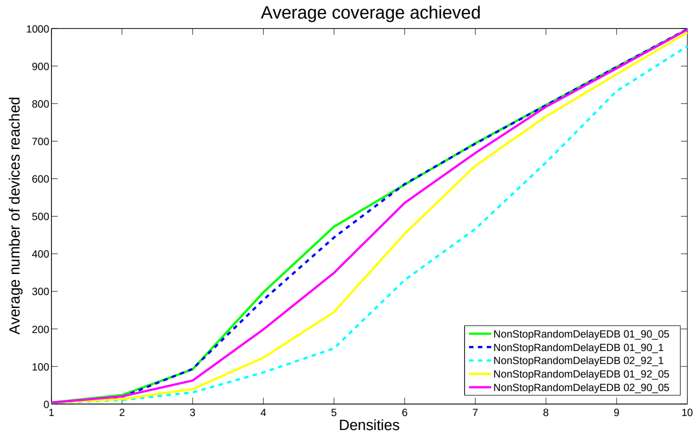


Figure 7.9: Coverage of the broadcasting process.

process is very fast since it reaches only a few nodes of the network, but as density increases the broadcast time needed to spread the message increases too. This behaviour is shown until the network becomes denser (around 600 or 700 devices for the configurations with maximum value of 2 seconds delay, and 400 devices for the 1 second delay). After that, the process becomes faster as in denser networks the dissemination of a message is easier.

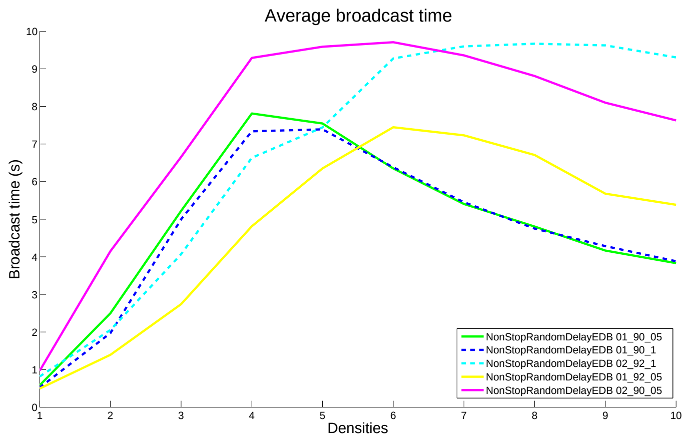


Figure 7.10: Broadcast time of the broadcasting process.

NonStopRandomDelayEDB 02_92_1 and *NonStopRandomDelayEDB 02_90_0.5* need a higher network density to start the drop of the broadcast time. These are the two variants that have higher value for the *delay interval* ([0 2]). As expected, the broadcasting process takes longer for those two variants.

For both *NonStopRandomDelayEDB 01_90_05* and *NonStopRandomDelayEDB 01_90_1* the behaviour is quite similar. But for *NonStopRandomDelayEDB 01_92_05* it takes shorter to disseminate the message in sparse networks, while longer in denser networks. This can be explained as the effect of having less forwarding nodes (due to a smaller forwarding area), what makes the dissemination shorter as is reaching less devices.

After studying the behaviour of the protocol under different settings we can consider that *NonStopRandomDelayEDB 01_90_05* is the one that, generally, behaves better. It is one among all the variants consuming less energy, and achieving more coverage in less time. But it is not clearly always the best option among all the different variants, what means, it is not easy to fine tune so many parameters for any network density.

7.4 The adaptive enhanced distance based broadcasting algorithm, AEDB

EDB tries to save energy by reducing the transmission power when forwarding the broadcast message so that it reaches the furthest neighbour. In denser networks, the probability of having a node close to the limit transmission range is high, therefore, the probability of EDB reducing the transmission power is low, as it was shown in Figure 7.8. Indeed, when the network is very dense the connectivity is usually very high. Thus, reducing the transmission power allowing the loss of some one hop neighbours will save energy without any detriment in the performance of the broadcasting process. Contrary, when the network is sparse, the node must maintain the network connectivity, as not doing so would make more difficult to spread a message through the whole network.

Therefore, we are presenting an extension of EDB, that is called the *adaptive enhanced distance based* broadcasting algorithm, AEDB hereinafter. AEDB considers the possibility of discarding some neighbours from the one hop neighbourhood in dense networks. In fact, the algorithm is able to adapt its behaviour to the network density. Potential forwarders set a random delay before resending. If the density is considered high the transmission range is reduced and some one hop neighbours are discarded.

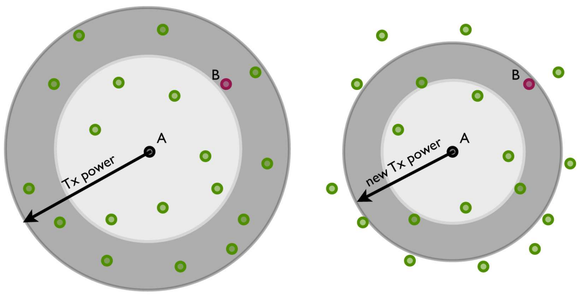


Figure 7.11: Selecting the new transmission power of the node.

As the density of the network is related to the number of one hop neighbours, the node is able to dynamically manage the transmission power used in terms of the node density. In order to decide how dense the network is, we consider the number of nodes located in the forwarding area. In this case, we are not considering all the nodes but only the potential forwarders, as a high number of them in that zone will make easier the dissemination of the message.

If this number is higher than a predefined threshold called *neighbours_Threshold*, then the network is considered dense, and the node decreases its transmission power for disseminating the message. In Figure 7.11, the idea is represented graphically. The number of neighbours

Algorithm 8: Pseudocode of AEDB.

```
Data: m: the incoming broadcast message.
Data: r: the node receiving broadcast message.
Data: s: the node that sent m.
Data: p: the received signal strength of m sent by s.
Data: pmin: the minimum signal strength received for m from any s.
Data: borders_Threshold: the signal strength threshold.
Data: neighbours_Threshold: # neighbours in the forwarding area.
if m is received for the first time then
    calculate p;
    update pmin;
    if pmin > borders_Threshold then
        r → drop message m;
    else
        waiting = true;
        wait time rand;
else if waiting then
    calculate p;
    if p > pmin then
        update pmin;
if pmin > borders_Threshold then
    r → drop message m;
else
    if # border_neighbours > neighbours_Threshold then
        estimate power to reach closest neighbour to borders_Threshold;
    else
        estimate power to reach furthest neighbour;
    transmit m;
waiting = false;
```

in the forwarding area (dark grey zone) is higher than *neighbours_Threshold*, thus, the transmission power is adapted to reach the new furthest node, in this case node *'B'* (in red). The pseudocode of the protocol is shown in Algorithm 7.

7.4.1 Experimental setup

The instantiation of the framework, as it was said, is the one presented in Figure 7.6. The simulator module of the framework and the mobility model are the same as the ones used for validating EDB in Sect. 7.3.2. There are minor changes in the experiments module for studying more dense networks.

Number of devices	500-1200	*borders_Threshold*	-90 dBm
Speed	[0, 2] m/s	*margin_Forwarding*	0.5 dBm
Size of the area	2000 m × 2000 m	*neighbours_Threshold*	8/12
Transmission power	16.02 dBm	*delay interval*	[0, 1] s
Direction change	every 20 s	*end_Threshold*	-95 dBm

Table 7.8: Parameterisation used

In Figure 7.8, we can see that the energy used increases up to 700 (7.55% savings), where devices use almost all the available energy (default), thus, no savings are performed. Therefore, in this work we are studying networks whose number of devices range from 500 to 1200 for the same simulation area. Different values for the *neighbour_Threshold* have been studied.

In Figure 7.12, the evolution of the percentage of the average number of neighbours in the forwarding area according to its size is presented. The behaviour for three network densities is studied. The selected network densities is 100, 200 and 300 devices/km^2. But actually, the behaviour is exactly the same. We can see in the figure that the number of 1-hop neighbours is the same, around 8%. The percentage of the number of neighbours in the forwarding area decreases as the value of the *borders_Threshold* also does (i.e. the size of the forwarding area decreases). For the value that was experimentally chosen of the *borders_Threshold*, *-90dBm* we can see the percentage is over 4%. We want to promote the energy savings in dense networks, thus, we get the the values of the 200 and 300 devices configuration and analyse them. Those values are: 8 and 12.

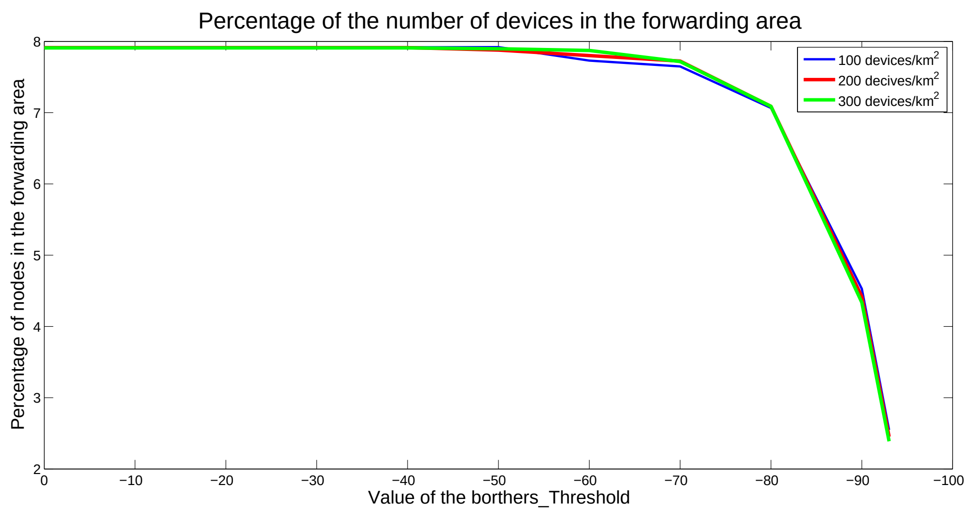


Figure 7.12: Percentage of the average number of neighbours in the forwarding area

7.4.2 Simulation Results

In order to obtain reliable results in our experiments, we are doing 100 independent simulation runs for each proposal of the algorithm. Table 7.9 presents the average of the values obtained after 100 runs for each of the network densities studied. These results are shown for each of the proposed settings of the *neighbour_Threshold* (8 and 12). As before, the parameters showed are: (1) the total energy used, (2) the coverage achieved, (3) the number of rebroadcasts performed, and finally (4) the broadcast time.

The first set of results in Table 7.9 correspond to EDB. It can be seen that from 800 devices on, the network is very dense as almost all the nodes receive the message. The lowest coverage is found in the sparsest network with 69,13% of devices reached, while the densest achieved 99.86%. The percentage of the number of forwarding nodes decreases as the density increases. The highest value for the number of rebroadcasts is 26.61%, obtained in the

		500	600	700	800	900	1000	1100	1200
EDB	Coverage	345.69 ±132.84	538.18 ±111.59	666.78 ±116.36	792.46 ±8.19	879.12 ±122.69	997.25 ±2.87	1097.88 ±2.02	1198.28 ±1.44
	Forwarding	114.02 ±45.75	159.67 ±34.71	178.61 ±31.89	191.35 ±4.88	193.37 ±27.73	200.73 ±4.52	204.27 ±5.07	206.45 ±4.57
	Bc time	7.89 ±2.66	7.81 ±2.02	6.47 ±1.62	5.55 ±1.11	5.04 ±1.19	4.58 ±0.88	4.12 ±0.75	3.78 ±0.70
AEDB	Coverage	223.84 ±145.70	382.55 ±178.74	527.59 ±179.58	632.22 ±189.67	779.18 ±166.05	857.68 ±227.74	1016.76 ±184.67	1134.21 ±164.64
	Forwarding	75.84 ±51.60	120.02 ±57.71	155.75 ±54.90	179.15 ±55.62	214.51 ±48.09	230.88 ±63.39	268.45 ±50.24	291.00 ±43.80
	Bc time	6.84 ±3.44	8.04 ±3.04	8.83 ±2.26	9.05 ±1.99	9.27 ±1.56	8.96 ±1.82	9.06 ±1.66	9.06 ±1.41
AEDB	Coverage	320.21 ±157.81	525.70 ±145.60	649.32 ±150.21	776.77 ±88.16	892.55 ±8.25	992.43 ±8.66	1094.71 ±5.37	1183.66 ±118.23
	Forwarding	106.17 ±54.26	157.45 ±44.61	176.71 ±41.58	195.33 ±23.31	211.20 ±5.68	227.25 ±5.78	244.16 ±5.75	258.51 ±26.62
	Bc time	7.61 ±3.16	7.81 ±2.25	6.88 ±1.95	6.62 ±1.33	6.67 ±1.24	6.59 ±1.16	6.51 ±1.14	6.42 ±1.23

Table 7.9: Results of the experiments.

network composed of 600 devices, and the lowest 17.20% for 1200 nodes. The broadcasting process takes never more than 7.89 seconds, and it was checked that the denser the network, the lower the duration.

We first observe the coverage achieved by each possible solution, as the main idea is to reduce the transmission power, but maintaing the good of performance EDB. For the results obtained when the *neighbours_Threshold* was set to 12, we observe that the percentage of the coverage achieved ranged from 64.04% to 99.51%, values that are similar to the ones obtained in the original EDB. Moreover, we did statistical tests comparing the coverage achieved by each proposal (EDB and AEDB with value 12) and EDB was better with statistical differences only for the networks with 1000, 1100 y 1200 devices, but the coverage obtained with AEDB is already satisfactory, around 99%.

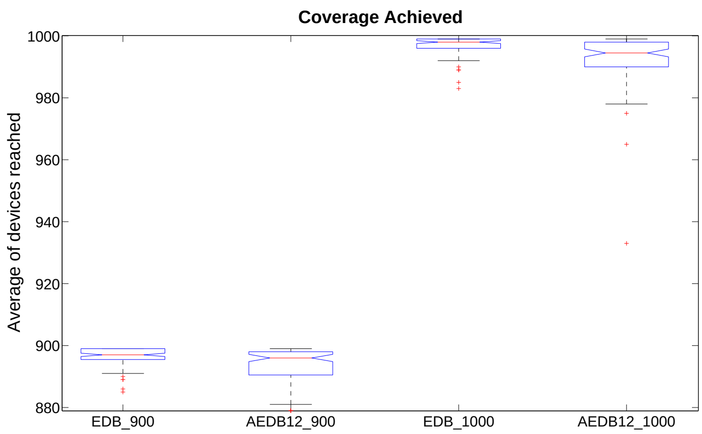


Figure 7.13: Statistical study of the coverage for 900 and 1000.

We show an example of one of the graphs obtained. In Figure 7.13, we can see the *boxplots* related to the coverage achieved by the original EDB for 900 and 1000 densities and AEDB (with the value 12 for the threshold) for the same densities. As explained before, AEDB and EDB do not have significant difference for 900, but EDB performs better for 1000 with statistical difference.

Regarding the time needed to broadcast the message, Table 7.9 shows that it is higher for the denser networks, but it must be noticed that never more than 2.64 seconds. The number of forwarding nodes is, as expected, increased. But the maximum percentage obtained for this configuration is smaller than the maximum for the original 26.24%. For the 3 first densities this configuration uses less nodes for forwarding than the original, but for the densest network (1200 devices), the number of rebroadcast is increased at maximum in 4.33% (total percentage for EDB is 21.54% in the same environment).

The values obtained with *neighbours_Threshold* equal to 8 gives lower values for the coverage, varying from 44.77% to 94.52%. The percentage of forwarding nodes is also smaller compared to the original EDB for the first four densities but it is a 5.83% higher for 1100 devices (up to 24.40%). The time needed for covering the network is also higher, exactly 5.28 seconds longer.

In order to make a statement about the benefits of reducing the transmission power when the density is high, we need to calculate the average of the transmission power each device uses in the broadcasting process. When reducing the transmission power, more rebroadcast are generally needed to cover the same area. The number of hops increases so does the broadcast time (those values are shown in Table 7.9). It is not only interesting to see the total energy used for the dissemination process in the whole network, but in reducing the energy each device spends in order to make longer its battery life. So, it is necessary to provide the average energy used per device. This is presented in Table 7.10, and also the percentage of the savings obtained in every density compared to the original EDB.

	500	**600**	**700**	**800**	**900**	**1000**	**1100**	**1200**
EDB	15.01	15.08	15.15	15.21	15.26	15.30	15.33	15.36
AEDB_8	14.48	14.06	13.59	13.05	12.61	12.23	11.91	11.68
% Eg. saved AEDB_8	11.49	20.93	30.18	39.19	45.67	50.68	54.50	57.15
AEDB_12	14.98	14.95	14.87	14.64	14.28	13.88	13.38	12.96
% Eg. saved AEDB_12	0.69	2.95	6.24	12.30	20.20	27.89	36.17	42.46

Table 7.10: Average of energy used per forwarding.

The values presented in Table 7.10 are in logarithmic scale, so after converting them to the linear one, we can calculate the percentage. We can see that, when setting the *neighbours_Threshold* to 8, the energy used is at minimum reduced 11.49% up to 57.15%. In the case of establishing the threshold to 12 neighbours, the saving is lower but still very considerable, from 0.69% to 42.46%. We must highlight that EDB already saves energy. If we consider the default transmission power (not performing any reduction as EDB does) the percentage of energy saved goes as far as 63.19% and 50.57% for the AEDB_8 and AEDB_12 settings respectively.

On one hand, the energy saved when using 12 as *neighbours_Threshold* is smaller but the coverage achieved is similar to the one obtained with EDB, and the time is only increased in 2.64 seconds for covering a network of 1200 devices. On the other hand, setting the threshold to 8 gives less power consumption, decreasing 5.34% the coverage and increasing in 5.28 seconds the total time of the broadcast process.

After analysing the behaviour of both EDB and AEDB in terms of the different scenarios and according to different parameters, we must highlight the difficulty of selecting the values for the threshold so that the performance of the algorithm is optimum.

As we already mentioned in Chapter 4, we consider essential the optimisation process before proposing any configuration setup of the protocol.

Part IV

Optimisation Process

Chapter 8

Optimisation of the EADB Protocol Using a Local Search

Contents

The design of communication protocols for MANETs is a complex and critical task that directly impacts on the network performance. As already mentioned in Chapter 4, in order to cope with the challenges of mobile ad hoc networks most of the algorithms rely on different thresholds for making its behaviour adaptive to the network conditions and/or requirements. As result of the unpredictable and highly changing topology, the behaviour of the protocol is highly sensitive to small changes in the set of configuration parameters. The performance of the algorithm highly depends on the setting used, that is typically experimentally chosen [27]. Therefore, fine tuning them for optimally configuring a communication protocol is required.

Moreover, in these self-organised networks not a single goal has to be satisfied but several (usually in conflict) like network resources, QoS, energy used, etc. Therefore, multi-objective optimisation is needed for fine tuning the protocol.

In this work, we propose a novel parallel multi-objective local search that optimises AEDB (see Chapter 7) in terms of coverage, energy used, broadcasting time, and network resources. The proposed method looks for appropriate values of the set of 5 variables that markedly influence the behaviour of the protocol to provide accurate tradeoff configurations in a reasonable short execution time.

8.1 Modelling the Optimisation Problem

In this thesis, we proposed two different approaches of broadcasting algorithm: *BODYF*, that uses a tree topology (see Chapter 6), and another with variable transmission range, *EDB/AEDB* (see Chapter 7). In the former, the protocol itself does not have any threshold or parameters, therefore, there is no need of this final optimisation step. However, in the latter different parameters are used. Next, as already proposed in Chapter 4, before tackling the optimisation algorithm we need to identify the parameters, define the objectives we pursue and perform the sensitivity analysis of our broadcast algorithm.

8.1.1 Identifying parameters and objectives

As mentioned in Chapter 7, both EDB and AEDB have a set of fixed parameters whose values determine the behaviour of the protocol. Although during the progress of this thesis, EDB was also optimised, we will show here only the optimisation of AEDB as it is an extension of EDB, and additionally, it has one more parameter to tune, the *neighbours_Threshold*. The methodology used is very similar, however, for a more detailed explanation of the EDB optimisation process, please refer to [284].

The already mentioned thresholds of AEDB are explained after and listed here: *borders_Threshold*, *margin_Forwarding*, the *delay* interval, and *neighbours_Threshold*.

- The value of the *borders_Threshold* sets the size of the forwarding area. The higher the threshold, the higher the number of potential forwarders, the coverage, the network resources used, and the number of collisions.
- The *margin_Forwarding* is the extra amount of energy added to the estimated transmission power in order to cope with possible outdated information. It is is related to both the energy used and the coverage achieved. The higher the margin value, the higher the energy used.
- The value of the *delay* interval sets the waiting time and also affects the behaviour of the protocol. We split this threshold into two different variables: *minimum delay* and *maximum delay* referring to the lower and upper value of the interval, respectively. If the delay is very high, the time used to spread the message will be high, but if it is very small, the number of collisions will probably increase.
- Finally, the *neighbours_Threshold* fixes the minimum number of neighbours needed in the forwarding area to consider the network is dense enough to discard some nodes. It affects the use of the network and the energy used. The lower the value, the lower the energy used and the higher the number of forwarding.

The quality of the performance of a broadcasting algorithm in ad hoc networks is usually related to some standard measurements. The aspects we are considering, and that are the most common ones in this kind of protocols are:

1. the **coverage** obtained, i.e., the number of devices that, after the dissemination process, receive the broadcast message,

2. the **energy used** by the broadcast process, measured as the sum of the energy every device consumes to forward the message,

3. the number of **forwarding**, considered as the amount of nodes that after receiving the broadcasting message decide to resend it,

4. and the **broadcast time**, considered as the time needed to spread a message in the network, since the source node sends the message until the last node receives it.

In preliminary work [283], we analyse the protocol in terms of the coverage achieved, the energy used and the broadcast time. Although the number of rebroadcasting is directly related to the total energy used, some of the solutions obtained were promoting to increase the value of the *borders_Threshold* (thus, the number of forwarding nodes and the number of hops), in order to decrease the energy used. The new furthest neighbour when discarding 1-hop nodes is the one closest to the *borders_Threshold*, if this value is high, the number of hops also increases. Thus, we included the number of retransmission in the broadcasting process as another objective.

However, from the point of view of the designer of the broadcasting algorithm, the higher the number of objectives the more complex the decision making and the optimisation process. Thus, in this work instead of optimising the protocol in terms of the previously mentioned four different objectives, we consider three objectives and a constraint. In [283], we could observe that the best solutions found do not take longer than 2 seconds for disseminating the broadcasting message. Therefore, in the evaluation process of the optimisation, we consider a solution is no longer valid if the broadcasting time is higher than 2 seconds, and analyse the following three objectives: (1) energy used, (2) coverage achieved, and (3) number of forwarding used.

In this work, we are proposing a technique to find the most suitable values of the parameters subject to a specific restriction. In our case, we only consider the time constraint, but more constraints could be easily added to the optimisation algorithms.

The purpose of this work is, therefore, to tune all these parameters so that the defined objectives are satisfied. However, those objectives are in conflict and promoting one incurs in a decrease of another. Therefore, we need to use a multi-objective technique (based on Pareto dominance) in order to obtain the best possible behaviour of the protocol, considering the three objectives and the constraint explained above.

The targeted Optimisation Problem

$s:$ instance of the ns3 simulator
$dmin = d_1 \in \mathbb{R} | d_1 \in minimum\ delay$
$dmax = d_2 \in \mathbb{R} | d_2 \in maximum\ delay$
$b = b_1 \in \mathbb{R} | b_1 \in border\_Threshold$
$m = m_1 \in \mathbb{R} | m_1 \in margin\_Threshold$
$n = n_1 \in \mathbb{R} | n_1 \in neighbour\_Threshold$
$z = (e, c, nb, t) = s(dmin, dmax, d, m, n)$

$$f(dmin, dmax, b, m, n) = \begin{cases} min\{e\} \\ max\{c\} \\ min\{nb\} \end{cases} ; \text{ s. t. } t < 2 \tag{8.1}$$

where z is the set of objectives: e stands for energy saved, c for coverage, nb for number of broadcastings and t is the broadcasting time. The domains of the variables *minimum delay*, *maximum delay*, *border_Threshold*, *margin_Threshold*, and *neighbour_Threshold* will be analysed next according to the results of the sensitivity analysis.

8.2 Sensitivity Analysis

In order to better understand the relationship between the AEDB parameters and the objectives, we carried out the sensitivity analysis that was introduced in 15.

We considered a wide range of values for every parameter in the sensitivity analysis: *min_delay* $\in [0, 5]$, *max_delay* $\in [0, 5]$, *border_threshold* $\in [0.0, 95.0]$, *margin_threshold* $\in [0.0, 16.2]$, and *neighbour_threshold* $\in [0, 100]$.

An example of the results obtained from the sensitivity analysis for the 300 devices network is shown in Figure 8.1. The influences of the variables on the different objectives are presented. We can see that the broadcast time is mainly influenced by *max_delay* and *min_delay*. The coverage achieved is markedly affected by the *neighbour_threshold*. While both *neighbour_threshold* and *border_threshold* are the parameters that influence most in the energy used and the number of forwarding.

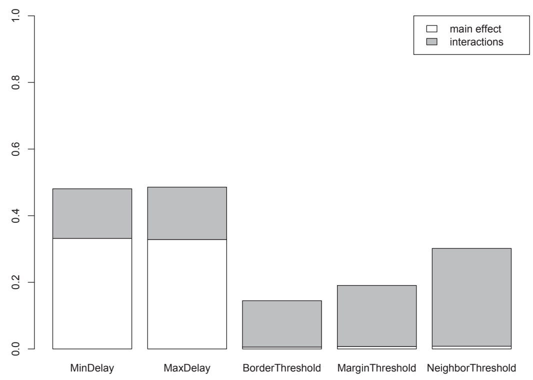
Generally speaking, we notice that the *margin_threshold* has the lowest direct influence on any objective or density. However, it interacts with all of them to some extent. The *delay* interval strongly affects the broadcast time in any density. For the number of forwarding, the *border_threshold* and the *neighbour_threshold* show the highest direct influence. The energy used is affected mainly by the *border_threshold* and the *neighbour_threshold* and then by the *delay*, in that order, but the importance of the *border_threshold* decreases with density, while *neighbour_threshold* becomes more prominent. The same behaviour is shown in case of the coverage, but for the densest network it is mainly affected by the *neighbour_threshold*.

Table 8.1 summarises our main findings. The influences of the parameters in the objectives are indicated in the table with yes, few, very few and no, in case there exist, there is just a few, very few or not influence is found. Symbols △ and ▽ indicate whether the variable should be increased or decreased, respectively, to optimise the corresponding objective. □ stands for no interaction found. Additionally, the interactions that are later used in the proposed iterated local search have a grey background colour.

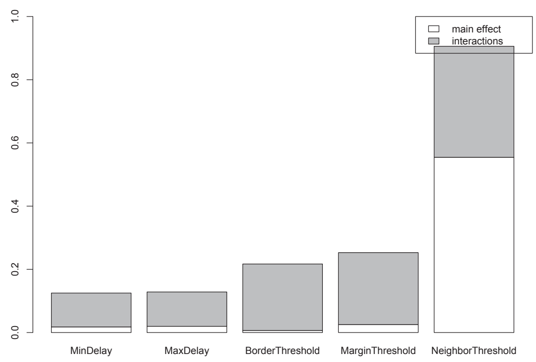
The domains of the variables *minimum delay*, *maximum delay*, *border_Threshold*, *margin_Threshold*, and *neighbour_Threshold* will be analysed next according to the results of the sensitivity analysis.

8.2.1 Limiting the search space

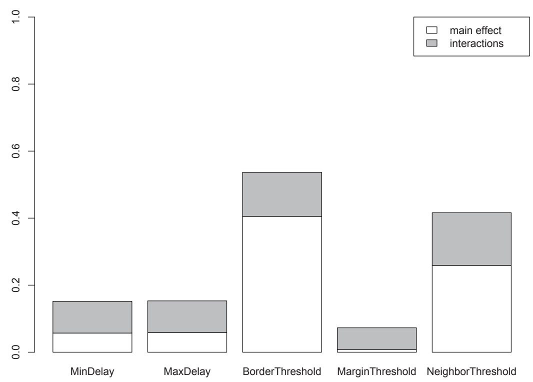
In order to find reasonable solutions and limit the search space, we choose an interval for each parameter. The optimisation algorithm only uses values included in those intervals for



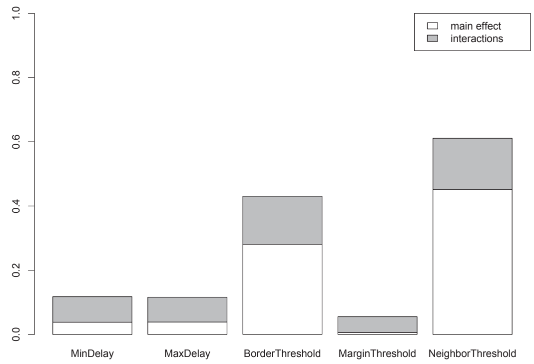
(a) Influence on the Broadcast Time



(b) Influence on the coverage



(c) Influence on the forwarding



(d) Influence on the energy

Figure 8.1: Influence of the parameters on the different objectives for the 300 devices network

	objective			
parameter	coverage *maximise*	forwardings *minimise*	energy used *minimise*	broadcast time *constraint*
border threshold	△ yes	△ **yes**	△ **yes**	□ few
delay	▽ few	△ few	▽ few	△▽ **yes**
margin threshold	△ very few	△ very few	△ very few	□ no
neighbors threshold	△ **yes**	△ **yes**	△ **yes**	▽ few

Table 8.1: Summary of the parameter sensitivity analysis

each of the parameters contained in every solution. These intervals are shown in Table 8.2, and they are large enough to cover the most suitable solutions.

Considering that the constraint included in the optimisation problem implies that the broadcast process must finish before 2 seconds, the interval for the *random_delay* that is [*minimum delay*, *maximum delay*] can not be very large.

minimum delay	[0, 1] s
maximum delay	[0, 5] s
border_Threshold	[-95, -70] dBm
margin_Threshold	[0, 3] dBm
neighbors_Threshold	[0, 50]

Table 8.2: Domain of the variables to optimise.

The border_Threshold considered covers around 86% of the transmission range. Only solutions that hardly use the distance based approach are discarded.

The *margin_Threshold* was included in the algorithm in order to cope with outdated information due to the mobility of the devices. Thus, if we want to **theoretically** study the benefit provided by this threshold by means of the additional distance obtained we do:

$$P_r = P_t - PL \tag{8.2}$$

We are considering in this work the *log distance path loss model* [275], thus the loss can be calculated in terms of the distance as follows. The general formula computing the path loss PL expressed in dB is:

$$PL = P_t - P_r = PL_0 + 10\gamma\ \log_{10}\frac{d}{d_0} + X \tag{8.3}$$

where:

- P_t is the transmission power;
- P_r is the reception power;
- d is the distance between transmitter and receiver;
- d_0 is the reference distance between transmitter and a reference point;
- PL_0 is the path loss in dB of the reference point;
- γ is the path loss exponent;
- X is a random variable that takes into account unpredictable fading.

We want to study the additional distance incurred when adding the *margin_Threshold*, thus we are considering that the received transmission power is the same. From equations 8.2 and 8.3, we can say that:

$$P_{r1} = P_t - PL1 = P_{r2} = P_t + margin\_Threshold - PL2 \tag{8.4}$$

$$-(PL_0 + 10\gamma\ \log_{10}\frac{d1}{d_0} + X) = margin\_Threshold - (PL_0 + 10\gamma\ \log_{10}\frac{d2}{d_0} + X) \tag{8.5}$$

$$-10\gamma\ \log_{10}\frac{d1}{d_0} = margin\_Threshold - 10\gamma\ \log_{10}\frac{d2}{d_0} \tag{8.6}$$

$$10\gamma\ \log_{10}\frac{d2}{d1} = margin\_Threshold \tag{8.7}$$

$$d2 = d1 * 10^{\frac{margin\_Threshold}{10\gamma}} \tag{8.8}$$

If we consider the value of γ (path loss exponent) is 3, according to Eq. 8.8, the percentage of the extra distance obtained ranges according to the *margin_Threshold*, from 0 to 25.89%.

Finally, the value of the *neighbour_Threshold* varies from 0 to 50 nodes. This value does not refer to the 1-hop neighbours but to the 1-hop neighbours in the forwarding area. That is a big interval considering that the number of nodes in the forwarding area was evaluated, and for a dense network of 300 devices/km^2 there are in average 12 nodes (and 24 1-hop neighbours).

We are first using a specific heuristic (iterated local search) to solve our optimisation problem, and then in next chapter more complex metaheuristics to validate the results obtained with the iterated local search.

Next, we present the proposed iterated local search.

8.3 The proposed iterated Local Search

The AEDB multi-objective local search (AEDB-MLS hereinafter), is a multi-start population-based local search algorithm that maintains several distributed populations. It is a massively parallel algorithm in which every solution in every population is simultaneously improved by the parallel application of an iterative local search procedure. When improving a solution in a given population, each local search procedure makes use of the other solutions in the same population in order to guide the search. The best solutions found by all the local search procedures are stored in a distributed external archive of non-dominated solutions (see Section 8.3.1). After each iteration, the best solutions in every population are stored into the external archive; once in the archive, each solution will be preserved only if it is a non-dominated solution.

The solutions in each population are improved in an isolated fashion for a fixed number of iterations, without collaborating with other populations. After a given number of iterations, each population is reinitialised by using randomly selected solutions from the external archive. This mechanism generates population diversity for the local search procedure, provides a collaboration mechanism between the distributed populations, and helps to avoid stagnation conditions.

The algorithm follows a non-hierarchical schema in which all parallel local search procedures are peers and no procedure performs a master role. Algorithm 9 presents the logic of the parallel local search procedure that iteratively improves each solution in the AEDB-MLS algorithm.

Initial feasible solutions are randomly initialised in each population. Each local search initialises its assigned starting solution s (line 1), evaluates it, and stores it to the external archive (lines 2–3). Then, all the local search procedures working on the same population are synchronised with each other in order to wait until the local population is fully initialised. Once this happens, the main loop is repeated until the *stopping condition* is met. In it, the local search operator is iteratively applied to the currently assigned solution s. In order to apply the operator to the solution s, another solution t is randomly selected from the local population and used as reference to quantify the perturbation applied during the local search (lines 6–8), as explained in Sect. 8.3.2. If the perturbed solution $\hat{s}$ is feasible (i.e., it complies with the time constraint), then s is replaced with $\hat{s}$ and the new solution is stored in the external archive (lines 9–12). When the *reinitialise condition* is met, the whole

Algorithm 9: Pseudocode of the local search procedure in the AEDB-MLS algorithm

```
1  s ← initialise_solution()
2  evaluate(s)
3  store_in_archive(s)
4  synchronise_threads()
5  while stopping condition is not met do
6      t ← random_solution(population)
7      ŝ ← local_search_operator(s,t)
8      evaluate(ŝ)
9      if ŝ is feasible then
10         store_in_archive(ŝ)
11         s = ŝ
12     if reinitialise condition is met then
13         s ← receive_from_archive()
14         synchronise_threads()
```

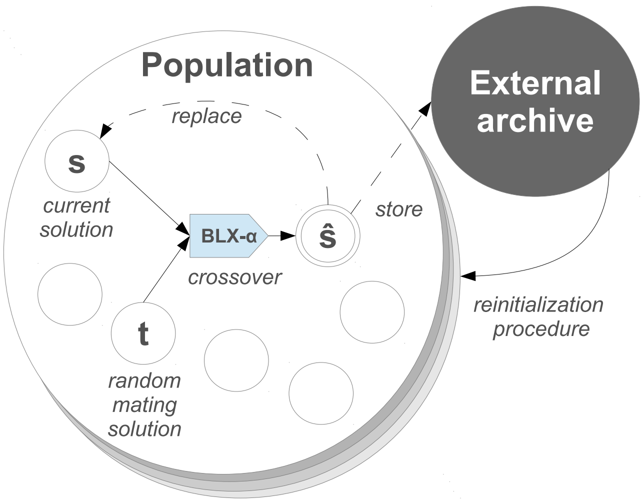


Figure 8.2: General overview of the AEDB-MLS algorithm

local population is discarded and replaced with randomly selected solutions form the external archive, restarting the local search procedures from a new location in the search space (lines 13–15). Figure 8.2 shows a general overview of the AEDB-MLS algorithm.

The AEDB-MLS algorithm follows a hybrid parallel model: message-passing is used for the collaboration between the distributed populations and the external archive, and shared-memory is used in the collaboration between solutions in the same population.

8.3.1 External archive

In this work, we use the Adaptive Grid Archiving (AGA) algorithm as the archiving method. The elite population in AEDB-MLS is limited in size, so an archiving technique must be applied to discard solutions when the maximum size of the archive is exceeded. The AGA algorithm was initially proposed as the density estimator for the Pareto Archived Evolution Strategy ($PAES$) algorithm [187]. It consists in dividing up the objective space into hypercubes with the goal of balancing the density of non-dominated solutions in the hypercubes. Then, when inserting a non-dominated solution in the Pareto front, its grid location in the solution space is determined. If the Pareto front is already full and the grid location of the new solution does not match with the most crowded hypercube, a solution belonging to that most crowded hypercube is removed before inserting the new one. The AGA strategy guarantees three very desirable properties for multi-objective optimisation algorithms: i) it maintains solutions at the extremes of all objectives; ii) it maintains solutions in all of the Pareto occupied regions, and iii) it distributes the remaining solutions evenly among the Pareto regions.

8.3.2 Local search operators

The local search operators were designed based on the sensitivity analysis study presented in Section 8.2. From this study we conclude that there are three different search criteria that can be applied when modifying a solution, depending on the objective to be improved: i) if the *energy used* objective or *forwardings* objective are targeted for improvement, then the *border_threshold* and *neighbors_threshold* parameters should be modified; ii) if the *coverage* objective is to be improved, then the *neighbors_threshold* parameter should be tuned; and iii) if the *broadcast time* constraint is to be improved, then the *min_delay* and *max_delay* parameters should be adjusted.

The local search operator uses the BLX-α operator, a recombination operator for real-coded EAs [103], which has been successfully used on a wide range of problems [145, 327]. Each iteration, one of the three aforementioned search criteria is randomly selected and it is applied to the current solution s. The BLX-α operator is independently applied to each of the corresponding parameters P defined by the selected search criterion. The value of the parameter $p \in P$ in solution s is modified by BLX-α as shown in Equation 8.9.

$$\begin{array}{c} \hat{s_p} = s_p + \phi \times [(3 \times \rho) - 2] \\ \phi = \alpha \times |s_p - t_p| \end{array} \tag{8.9}$$

where t is an auxiliary solution randomly selected from the current population, s_p is the value of the parameter p in the solution s, $\hat{s_p}$ is the value of the parameter p in the solution $\hat{s}$, and $\rho \in [0, 1)$ is a randomly selected number. The additional parameter $\alpha \in (0, 1)$ represents the perturbation magnitude of the BLX-α operator, the higher the α-value the more perturbed is the s_p-value during the crossover. The parameter α was empirically tuned during the experimental analysis.

8.4 Experimental setup

In this case, we are using the optimisation module of the framework, as well as the connection between the monitoring tools module and this one. We rely on the ns3 simulator for evaluating every solution of the optimisation algorithm. As an attempt to obtain concluding results in the evaluation of solutions, the monitoring tools module simulate every protocol configuration (i.e., every solution) on 10 different networks. The fitness value for every objective is defined as the average of the values obtained for the 10 networks in every objective. We always used the same 10 different seeds in our ns3 simulations to evaluate the solutions, thus, we analyse the behaviour of AEDB over 10 different networks. The monitoring tools gives this fitness value to the optimisation algorithm as input.

This instantiation of the framework is the one that will be used for the two remaining optimisation chapters, and is shown in Figure 8.3.

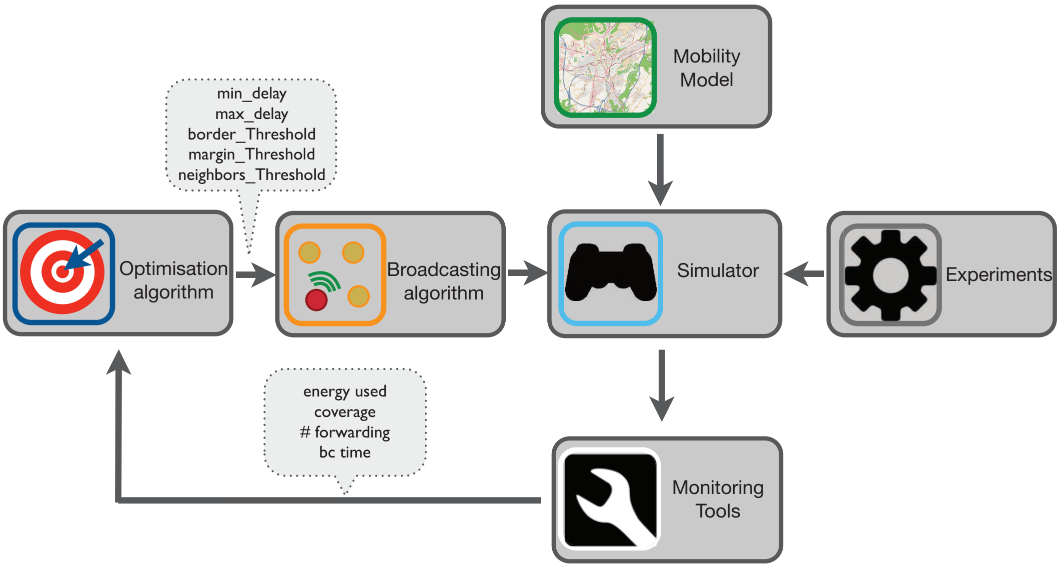


Figure 8.3: Illustration of the interactions of the optimisation algorithm module

Regarding the configuration of the simulator and the mobility model modules of the framework, we are using the same configuration as the one presented in Chapter 7. That is, ns3 simulator and random walk [128]. Please, refer to Chapter 7 for a more detailed explanation of the instances used.

Devices/km^2	100, 200, 300
Speed	[0, 2] m/s
Size of the area	500 m × 500 m
Default trans. power	16.02 dBm
Dir. & speed change	every 20 s

Table 8.3: Parameterisation used

In the experiments module, the simulation environment used is a square area of 500 m side. We study three different network densities in the optimisation process, with 100, 200, and 300 devices/km^2. All the parameters are summarised in Table 8.3. As done in Chapter 7, the network evolves for 30 seconds. After these 30 seconds, a node starts the broadcasting process. The simulation stops after 40 seconds

The optimisation algorithm module is instantiated with the proposed AEDB-MLS algorithm. Each execution is performed using 8 distributed populations with 12 individuals per population and 1 threads per individual (i.e. 12 threads per population), the maximum number of cores per computing node available in the computing platform. A limit of 250 solution evaluations per thread is used as a stopping criterion for the AEDB-MLS algorithm. This gives a total of 24000 evaluations per algorithm execution.

The size of the external archive explained in Sect. 8.3.1 is limited to 100 solutions. A configuration analysis was performed using the less dense network in order to find out the best values for the α parameter used in the BLX-α operator, and the number of iterations needed to meet the *reset condition* in the AEDB-MLS algorithm. The candidate values for the parameter settings study were: $\alpha \in \{0.1, 0.2, 0.3\}$, and *reset condition* $\in \{15, 25, 50\}$. The best results were obtained using $\alpha = 0.2$, and *reset condition* $= 50$. These are the values we adopted for AEDB-MLS.

In order to have confident results, in the monitoring tools module the quality of the solution is not tested in one single network but in 10 different networks, and the fitness value of each objective is defined as the average value of theses 10 runs. These 10 networks are always the same for evaluating every solution. Additionally, for the same reason, the optimisation algorithm is not executed only once, but for every density, 30 independent runs are performed.

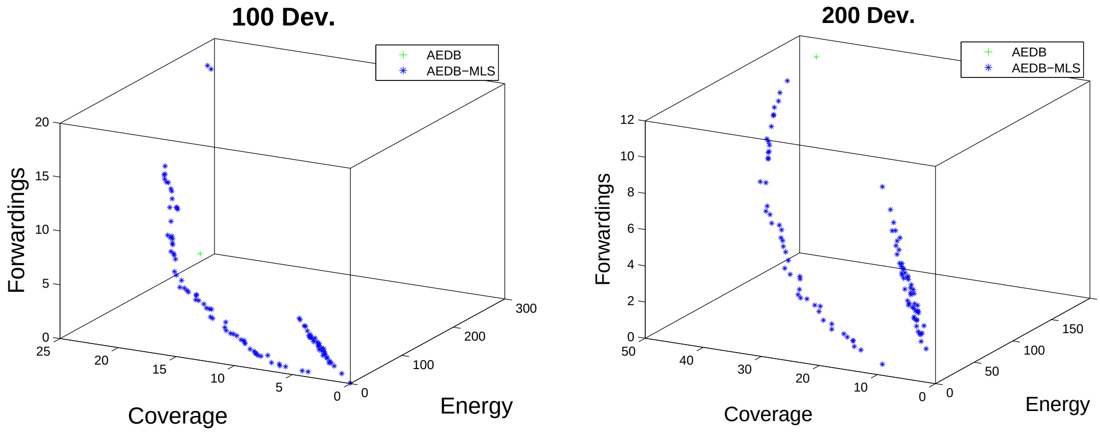
As in previous chapters, measuring the quality of a given parameter configuration (i.e., a tentative solution to the problem) is a complex task that must evaluate the solution in terms of the coverage, the energy used, the number of forwarded messages, and the broadcasting time obtained by the optimised protocol in any network configuration.

8.5 Simulation results

The results obtained in our experiments are summarised in this section. We evaluate he quality of the AEDB configurations found by AEDB-MLS. As we are working with multi-objective optimisation, the result is not a single solution but a set. This set of solutions is called the Pareto front, and it is composed of non-dominated solutions, i.e. there is not a solution that is better than another for all the objectives (please refer to Sect. 3.4).

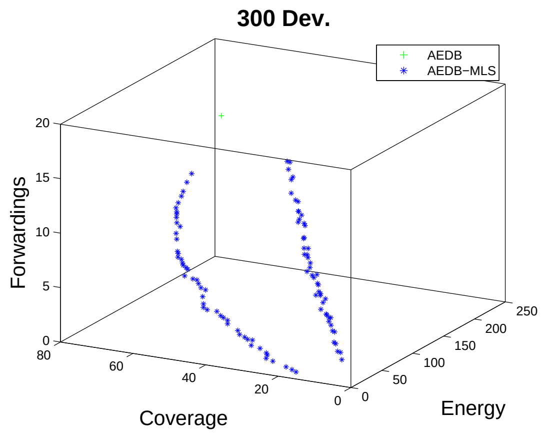
We show in figures 8.4a, 8.4b, and 8.5a the results obtained for each of the studied densities, that is, 100, 200, and 300 nodes per square km, respectively. These graphs are obtained by merging all the Pareto solutions found in every independent run (for the same density) into one single Pareto front. We call these merged Pareto fronts the AEDB-MLS Pareto front, and the maximum possible number of solutions it can contain is fixed to 100. Therefore, the AEDB-MLS Pareto fronts plotted in these figures contain the best 100 solutions found by the algorithm for every density in the 30 independent runs performed. Moreover, the original configuration of AEDB is also included in the figure as a green cross.

In the Pareto front approximations shown in Figure 8.4 and Figure 8.5, it stands out that the fronts have two clear sets of solutions in the three scenarios. For the lowest energy values in the approximated range $[-20, 20]$ *dBm*, solutions provide very low coverage and high number of forwarding, following a linear relationship between these two objectives in which the coverage value is similar to the number of forwarding. These are typically solutions in which devices are only broadcasting the message to their closest one, and therefore the number of forwarding is very close to the number of devices receiving the message (i.e., the coverage). However, for higher energy values over 20 *dBm*, the shape of the Pareto front changes, and we can see a clearly defined front of solutions in which coverage values are growing much faster than the number of forwarding. This region of the front is the one in which we are more interested, since it is providing high coverage at a reasonable number of



(a) 100 Devices/km^2 (b) 200 Devices/km^2

Figure 8.4: The Pareto fronts obtained with MLS for the studied densities and the original AEDB



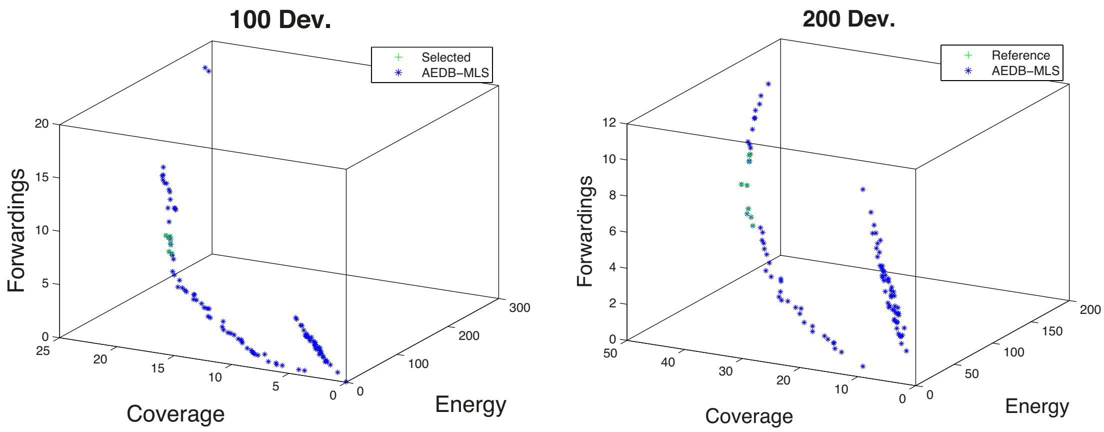
(a) 300 Devices/km^2

Figure 8.5: The Pareto fronts obtained with MLS for the studied densities and the original AEDB

forwarding and energy requirements.

We compared these Pareto front approximations to the solution obtained with the original configurations of AEDB for the three network densities. Looking for fair comparisons, AEDB with the initial settings was executed on 10 different networks using the same seeds as in the optimisation algorithms. The average values for each of the objectives is compared to the solutions of the Pareto front. In case, at least one objective of the solution is better than AEDB and better or equal for the rest of the objectives, the solution is said to be dominant.

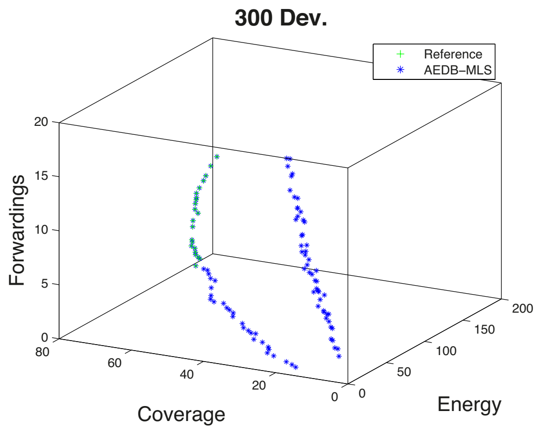
We calculated the number of solutions contained in the Pareto front that dominates the original configuration of AEDB. We found that for the 100, 200 and 300 devices configuration, 8, 15 and 1 solutions dominate AEDB, respectively. However, as explained above, the Pareto



(a) 100 Devices/km^2 (b) 200 Devices/km^2

Figure 8.6: The Pareto fronts obtained with MLS for the studied densities and the solutions selected

front used is the merge of the 30 Pareto fronts obtained. From this 30 independent executions of the optimisation algorithm, 100 solutions are selected so that they are uniformly distributed over the front. Therefore, it is possible to calculate the number of dominant solutions over all the 30 Pareto fronts. We obtained that for 100 devices 187 solutions dominate AEDB, however only 13 are non dominated. For the 200 devices configuration 291 dominate AEDB but only 17 are non dominated. Finally, 31 solutions were found on 300 devices configuration but there is a single solution that dominates the rest.



(a) 300 Devices/km^2

Figure 8.7: The Pareto fronts obtained with MLS for the studied densities and the solutions selected

However, from the point of view of the protocol designer not all solutions included in the Pareto front are valid solutions. In order to facilitate the decision making, it is possible to apply constraints to the optimisation process or later to the solutions obtained.

We can apply restrictions to the Pareto front solutions in order focus on one part of the front, i.e., targeting a minimum coverage or network usage. It is also possible to apply these restrictions during the optimisation process so that all the solutions found by the optimisation algorithm fit in this part of the front (as we did with the broadcast time). However, including the restriction during the optimisation will prevent the designer from the complete view of the protocol behaviour mentioned before. That is what happened with the broadcast time, in this work, we do not know the performance of the protocol for solutions taking longer than 2 seconds.

In this work, we want to have a comprehensive idea of the protocol for all the different objectives (that is why no more constraint were added), and then focus on one small area for better understanding the behaviour of the protocol. For that, we selected some solutions from the Pareto front and applied some restriction considering we are using a broadcasting algorithm in an ad hoc network

As we are dealing with a broadcasting algorithm, we consider that achieving less than 80% coverage or more than 30% number of forwarding is not acceptable. Therefore, we apply those restrictions to the solutions found. However, for the 200 and 300 network configurations the number of solutions accomplishing these restrictions is high. Thus, a more restrictive constrain is applied, 15% of forwarding nodes. The solutions obtained are sorted in terms of the energy saved per forwarding and the 10 best ones are selected. For the 100 devices configuration only 8 solutions are found that fulfil the constraints, thus all of them are included. The solutions obtained are shown in figures 8.6 and 8.7 along with the respective Pareto fronts for each density.

We can see in figures 8.6 and 8.7, that only for the 300 devices configuration the solutions with higher coverage also fit the constrain regarding number of forwarding nodes. In both 100 and 200 configurations, there are solutions obtaining higher coverage that are not in our selected subset.

Before trying to analyse the effectiveness of the solutions found by the local search, we want to validate the results obtained. Although solutions that outperformed the original configuration were found, we decide to validate the results obtained using more advance optimisation algorithms.

In the next chapter we are using more complex metaheuristic, two evolutionary algorithms, for solving the same problem. The results obtained give a reference of the accuracy of the solutions found by the local search. Moreover, it is possible to include the proposed local search in the evolutionary algorithm or in any other metaheuristic, so that the solutions found are more accurate.

Chapter 9

Optimisation of the AEDB Protocol Using Evolutionary Algorithms

Contents

In previous Chapter, we tackled the optimisation process using a massively parallel Iterated Local Search algorithm specifically design for our broadcasting algorithm, AEDB-MLS. However, we wanted to validate the results obtained with more advanced metaheuristics.

In this chapter, we optimise the AEDB protocol using advanced state-of-the-art evolutionary algorithms in order to find optimal parameter settings for the protocol. The obtained results will allow us validating our local search.

9.1 Description of the optimisation algorithms used

Evolutionary Algorithms (EAs) [42] are a popular family of metaheuristics. One important feature of EAs is that they work with several candidate solutions at the same time, therefore simultaneously exploring several different regions of the search space. This allows EAs to better explore the search space and reducing the probabilities of getting stuck in local optima with respect to other metaheuristics families.

Our problem was defined in Chapter 8 as a three-objectives one, we need to maximise its coverage, and minimise the energy used and the number of forwarded messages by the devices. Thus, we require approximation algorithms that face multi-objective optimisation problems. We choose the well-known Non-dominated Sorting Genetic Algorithm II (NSGAII) algorithm, and CellDE, a cellular differential evolution algorithm for performing the experiments in this chapter.

Specifically, we use NSGAII [90], which is probably the most referenced multi-objective algorithm in the literature, and CellDE, because it is a highly competitive evolutionary algorithm for multi-objective optimisation that has proven to perform specially well for three-objectives problems with continuous variables [99], as it is our case. We use the implementations of the algorithms available in jMetal framework [98], with the configurations provided in their original papers (set by default in jMetal).

Next, we are explaining the two evolutionary algorithms.

Algorithm 10: Pseudocode for NSGAII

```
1  //Algorithm parameters in 'nsga'
2  InitialisePopulation(nsga.pop);
3  EvaluatePopulation(nsga.pop);
4  while ! StopCondition() do
5      for index ← 1 to cga.popSize/2 do
6          parents←SelectParents(nsga.pop);
7          children←Crossover(nsga.Pc,parents);
8          children←Mutate(nsga.Pm,children);
9          offspringPop←Add(children);
10     EvaluatePopulation(offspringPop);
11     union←Merge(nsga.pop, offspringPop);
12     fronts←SortFronts(union);
13     (Pop', lastFront)←GetBestCompleteFronts(fronts);
14     if size(nextPop) < nsga.popsize then
15         Pop'←BestAccToCrowding(lastFront,nsga.popsize-size(Pop'));
```

9.1.1 Non-dominated Sorting Genetic Algorithm, NSGAII

The NSGAII [90] algorithm is, undoubtedly, the reference algorithm in multi-objective optimisation (MO). Even when nowadays better algorithms exist for MO, NSGAII is still the most referenced one, probably because of its simplicity and good operators and components, adopted by a large number of other MO algorithms.

A pseudocode of NSGAII is given below. NSGAII does not implement an external archive of non-dominated solutions, but the population itself keeps the best non-dominated solutions found so far. The algorithm starts by generating an initial random population and evaluating it (lines 2 and 3). Then, it enters in the main loop that evolves the population. It starts by generating a second population of the same size as the main one. It is done by selecting two parents (line 6) by binary tournament based on dominance and crowding distance (in the case the two selected solutions are non-dominated), applying the recombination operator (typically SBX [90], standing for simulated binary crossover) to generate two new solutions (line 7), which are mutated in line 8 (typically using polynomial mutation [90]) and added to the offspring population (line 9). The number of times this cycle (lines 5 to 10) is repeated is the population size divided by two, thus generating the new population with the same size as the main one. This new population is then evaluated (line 11), and merged with the main population (line 12). Now, the algorithm must discard half of the solutions from the merged population to generate the population for the next generation. This is done by selecting the best solutions according to ranking and crowding, in that order. Concretely, ranking consists on ordering solutions according to the dominance level into different fronts (line 13). The first front is composed by the non-dominated solutions in the merged population. Then, these solutions in the first front are removed from the merged population, and the non-dominated ones of the remaining solutions compose the second front. The algorithm proceeds to iterate like this until all solutions are classified. To build the new population for the next generation, the algorithm adds those solutions in the first fronts until the population is full or adding a front would suppose exceeding the population size (line 14). In the latter case (lines 15 to 17), the best solutions from the latter front according to crowding distance (i.e., those solutions that are more isolated in the front) are selected from the front to complete the population. The process is repeated until the termination condition is met (lines 4 to 18).

9.1.2 Cellular differential evolutionary algorithm, CellDE

Cellular differential evolution algorithm, CellDE [99] is a muti-objective evolutionary algorithm (MOEA) for three objectives problems. It is a hybrid of MOCell [246], a cellular MO algorithm, and Differential Evolution [268]. Therefore, it implements a cellular population topology. The pseudocode of CellDE is given below. As it can be seen, it starts by creating and evaluating a random initial solution (line 2) and building an initial empty archive of non-dominated solutions (line 3). Then, the algorithm iterates to evolve the population until a given termination condition is met (lines 4 to 13). In every iteration, all individuals are evolved by selecting two different parents (line 7) from the neighbourhood of the current individual being evolved and applying a standard differential evolution operator later described (line 8). The resulting offspring individual is then evaluated, and it replaces the current individual in the population if the current one does not dominate it (all individuals are updated at the same time, in a synchronous way). Then, the offspring is inserted into the archive following the SPEA2 density estimator. Finally, some randomly selected solutions in the archive are moved back to random positions in the population.

The differential evolution operator used in CellDE is defined by Eq. 9.1, where u is the offspring, i is the individual index, j is the variable position in the representation, x is the current individual, and x_{r_0}, x_{r_1} and x_{r_2} are the three randomly selected parents. Variables F and CR are two control parameters specific of DE, and they represent the scaling factor for mutation and the control of the recombination operator, respectively.

$$u_{i,j} = \begin{cases} x_{r_0,j} + F \cdot (x_{r_1,j} - x_{r_2,j}) & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} . \tag{9.1}$$

Algorithm 11: Pseudocode for CellDE

```
//Algorithm parameters in 'cellde'
InitialisePopulation(cellde.pop)
CreateFront(cellde.front)
while ! StopCondition() do
    for individual ← 1 to cellde.popSize do
        n_list←Get_Neighbourhood(cellde,position(individual));
        parents←SelectDifferentParents(n_list);
        offspring←DifferentialEvolution(parents, individual);
        Evaluation(offspring);
        Add(position(individual),offspring,cellde);
        AddToArchive(individual);
    PopFeedback();
```

9.2 Experimental setup

The overview of the experimental framework used in this chapter is the same that was presented in Chapter 8 (see Figure 9.1).

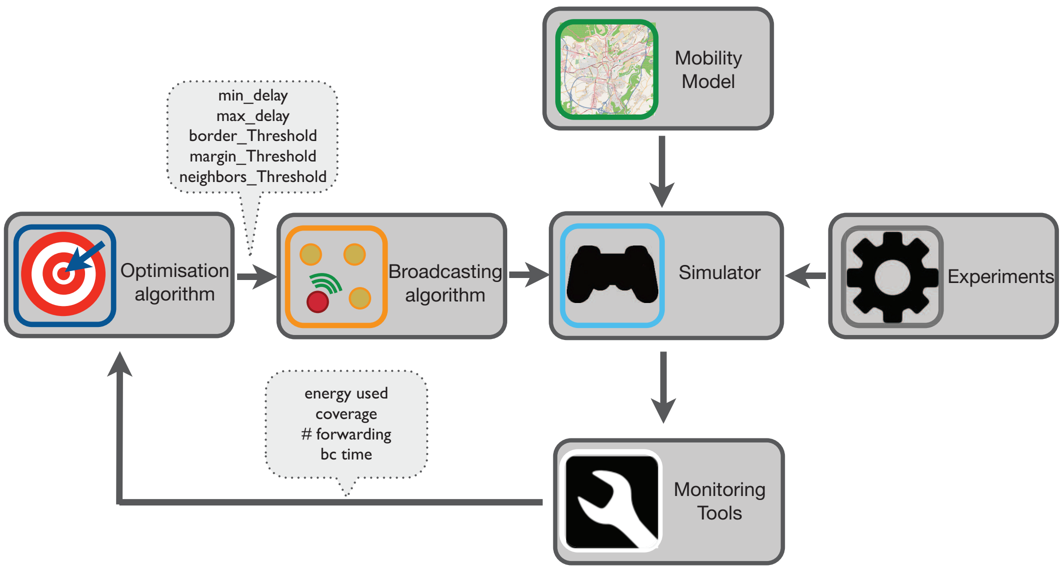


Figure 9.1: Illustration of the interactions in the optimisation algorithm module of frameworks

The only change introduced in the framework concerns the optimisation algorithm module. In this chapter, we are using two different optimisation algorithm from the state of the art that were presented in Sect. 9.1. We are using the implementations of CellDE and NSGAII provided in jMetal framework [98], and we used the default configurations, which match the ones proposed by the authors of the algorithms.

Table 9.1 summarises the configuration proposed by the authors. The termination condition of the algorithms was set to 10000 evaluations performed. Individuals are encoded as an array of 5 real values, and the value used for evaluation for the integer variable (*neighbour_Threshold*) is the integer part of the real value. Therefore, the real-coded variables are *minimum delay, maximum delay, border_Threshold*, and *margin_Threshold*. The problem is

Population size	100 (NSGAII)
	10×10 (CellDE)
Termination Condition	10,000 function evaluations
Selection	Binary tournament (NSGAII)
	Three random parents (CellDE)
Neighbourhood	C9 (CellDE)
Crossover probability	$p_c = 1.0$
Mutation probability	$p_m = 1/\text{chrom\_length}$

Table 9.1: Generic parameters used for the studied algorithms

decomposed into eight islands, each of them being in charge of the optimisation of 32 bits of the chromosome of each individual.

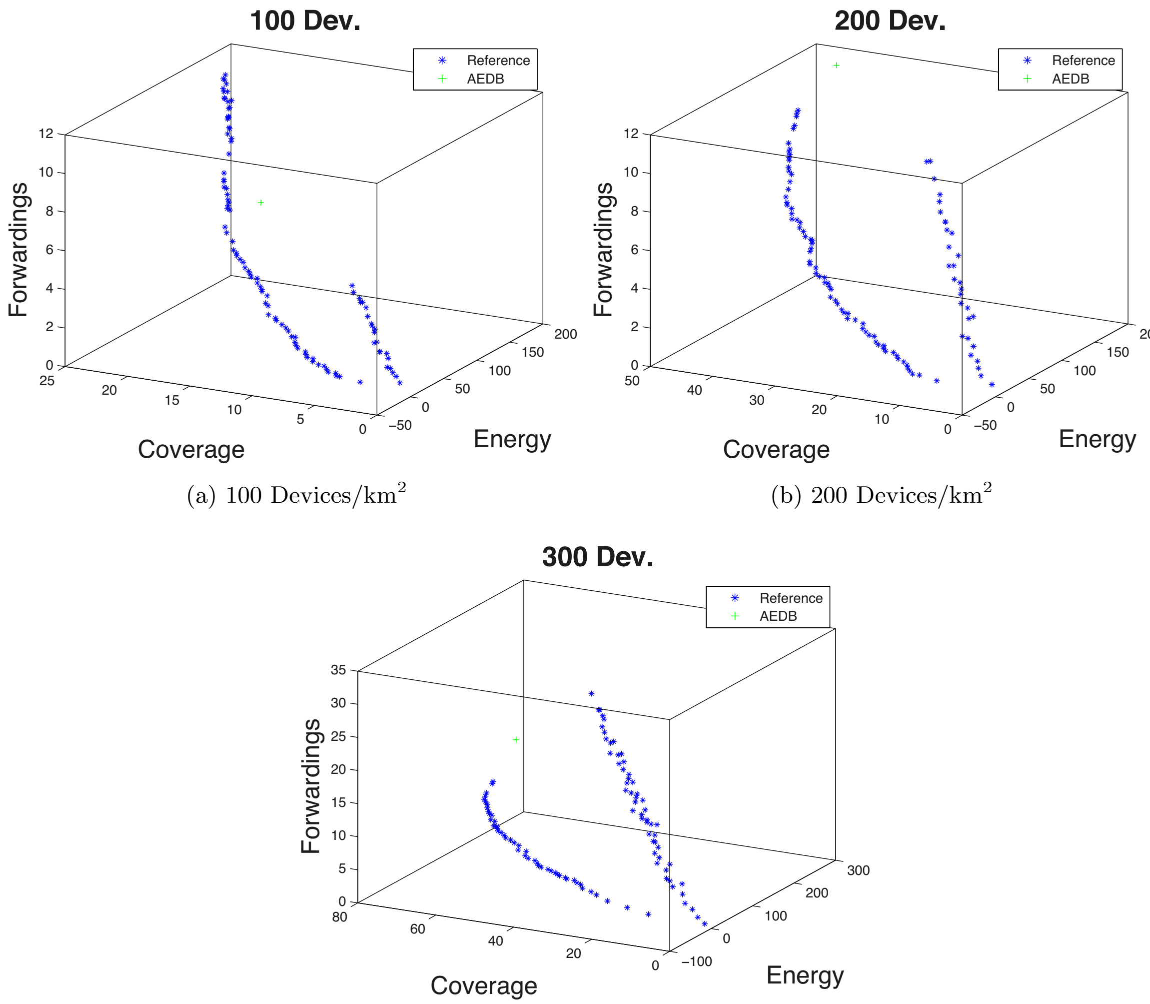
Finally, in order to provide concluding results, in the monitoring tool module the Wilcoxon unpaired signed-ranks test [116, 304, 375] is used in some experiments to look for significant differences on the results provided by the algorithms and protocols, compared pairwise. In the case of MO optimisation algorithms, the test is applied on the results obtained by the quality indicators. This test is a non-parametric alternative to the student t-test. This method is used to check whether two data samplings belong to different populations or not. Therefore, we can use it to compute if there are statistically significant differences between the data reported by two different algorithms after the independent runs. The null hypothesis for this test is that the median difference between pairs of observations in the underlying populations represented by the samples of results provided by the algorithms is zero.

We use symbols ▲, ▽, and – to show existing significant differences in the pairwise comparison of the algorithms according to the Wilcoxon test. These symbols will be arranged in tables. Symbol ▲ means that the algorithm in that row is statistically better than the algorithm in the corresponding column with 95% confidence level. On the contrary, ▽ stands for significantly worse result of the algorithm in that row compared to the algorithm in the column. Finally, – indicates that no statistical differences were found between the corresponding algorithms.

9.3 Simulation results

As already explained in Chapter 3, multi-objective optimisation algorithms do not give a single solution but a set of them, non better than another that conforms the Pareto front. In this chapter, in order to find the best possible results we are using two optimisation algorithms. Therefore, we have as results two different sets of solutions, i.e. two different Pareto front. We merge this Pareto fronts, in order to have only one *Reference* Pareto front with the best solutions obtained from both algorithms. As we did in the previous chapter, we limit the size of the Pareto front to 100 non-dominated solutions. The resulting *Reference* Pareto front as well as the original configuration of AEDB are shown in Figure 9.2, for every density.

As we already mentioned in previous chapter, it this case it the two different sets in the Pareto fronts also appeared. As it was stated before, we are only interested in this part of the fronts were for energy values over 20 *dBm*, the shape changes, and we can see a clearly defined



(a) 100 Devices/km^2

(b) 200 Devices/km^2

(c) 300 Devices/km^2

Figure 9.2: The *Reference* Pareto fronts obtained for the studied densities and the original AEDB configuration

solutions in which coverage values are growing much faster than the number of forwarding.

We also compared in this case these Pareto front approximations to the solution obtained with the original configurations of AEDB for the three network densities. We found 11, 17 and 1 solutions dominating the original configuration of AEDB (i.e., providing better results for the three objectives) for the three different configurations: 100, 200 and 300 devices/km^2, respectively.

We considered all the solutions found by each algorithm in the 30 executions and we found out that CellDE outperforms AEDB in 201 solutions, NSGAII in 349. From these solutions, 17 were non dominated with the solutions in the Pareto front. For the 200 devices configuration, CellDE outperforms AEDB in 343 solutions, and NSGAII in 527. From this set of solutions, 37 were non dominated. In case of the 300 devices network, CellDE dominates

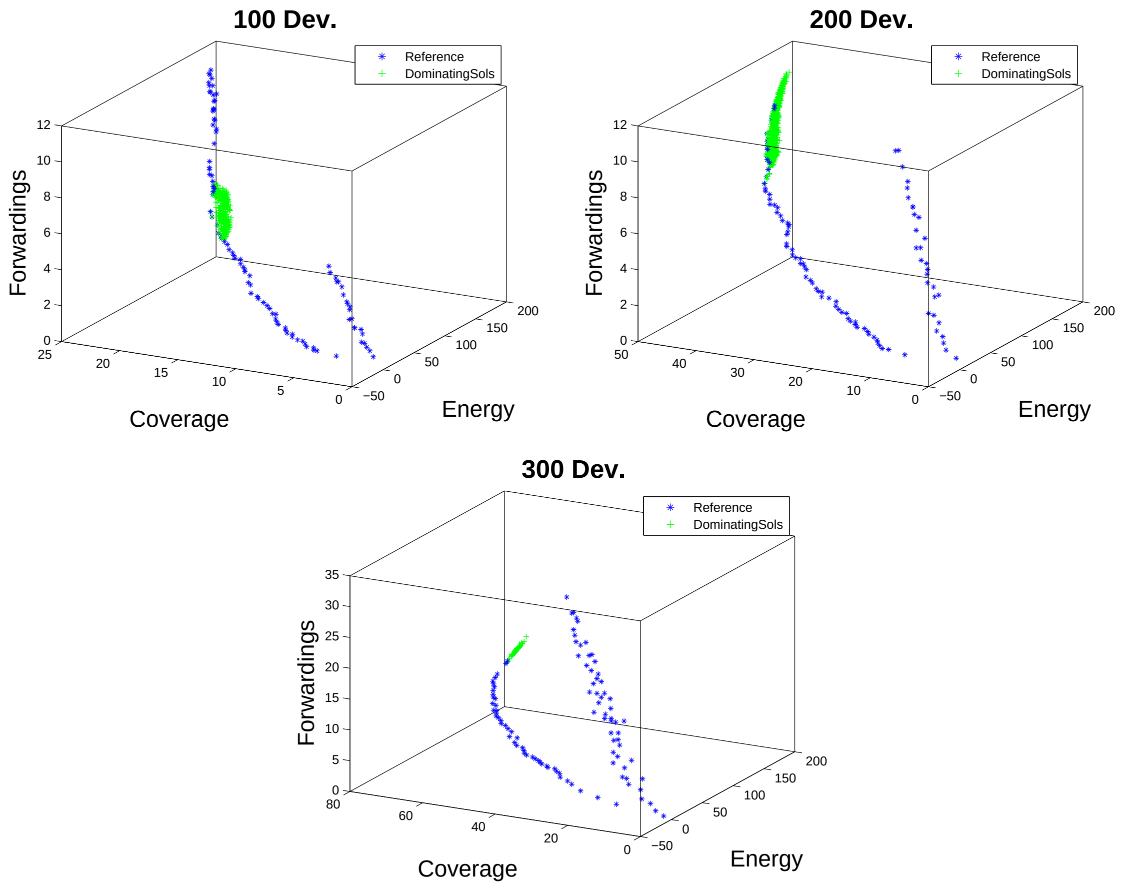


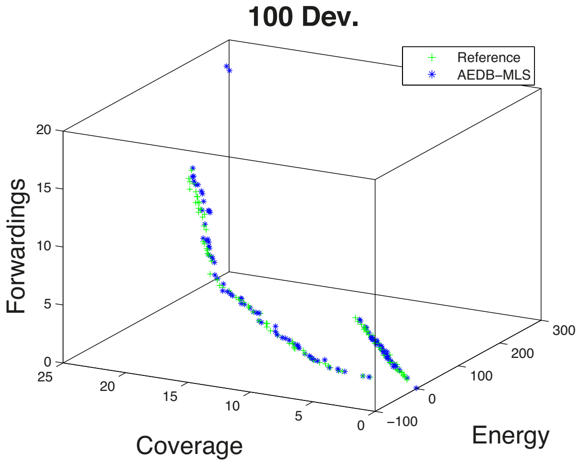
Figure 9.3: The reference Pareto fronts obtained after merging all the Pareto front approximations ob tained with all dominating solutions. Green crosses are the solutions that dominate the original AEDB configuration.

AEDB in 29 solutions, and NSGAII found 53 better solutions. From those 82 solutions, 2 were non dominated solutions with those in the Pareto front. In Figure 9.3, the solutions found that dominate AEDB are plotted as green dots.

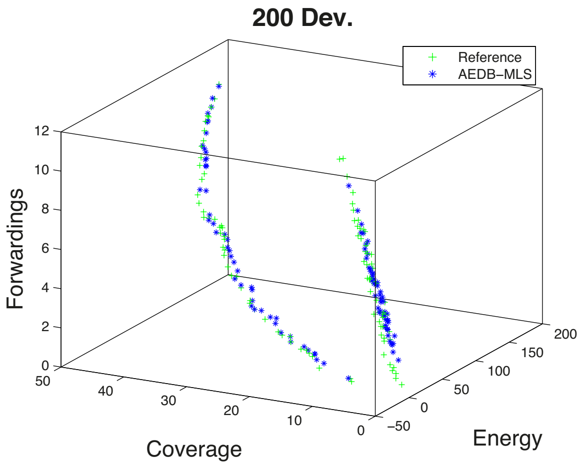
From Figure 9.3, we can see that in both the Pareto front of 100 and 300 devices networks, the dominating solutions added are localised in a small area. However, in the 200 devices case we can see that the solutions are more spread along the Pareto front. The reason is that the original configuration of AEDB is not as fine tuned as in the case of the other two network densities.

9.4 Comparison between the iterated local search and evolutionary algorithms

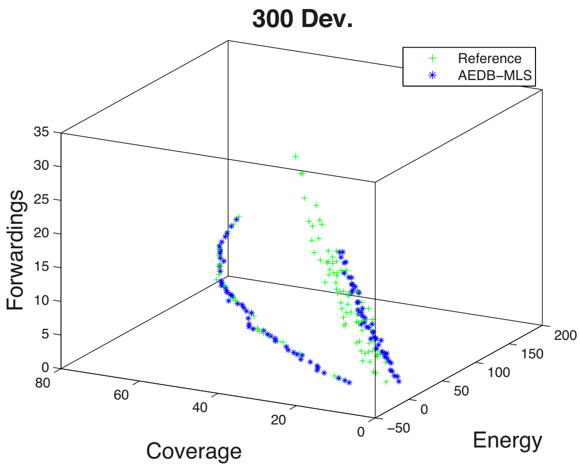
In order to evaluate the solutions obtained with the proposed iterated local search (AEDB-MLS) in Chapter 8, we compare them to those obtained by the MOEAs, NSGAII and CellDE. We build the Pareto front approximation of AEDB-MLS with the best non-dominated solu-



(a) 100 Devices/km^2



(b) 200 Devices/km^2



(c) 300 Devices/km^2

Figure 9.4: The *Reference* Pareto front and the Pareto fronts obtained with AEDB-MLS algorithm.

tions found in 30 independent executions (according to the AGA method) for every network density. They are displayed in Figure 9.4, and compared versus a *Reference* Pareto front approximation built from the best results found by the two MOEAs in 30 independent runs for every density (AGA was used in this case too).

Results show that the MOEAs and AEDB-MLS obtain similar Pareto front shapes. As stated before, there exist two clearly differentiated set of solutions. One with very low values of energy approximately between $[-20, 20]$ dBm and similar values for the coverage and the number of forwarding, and another with higher energy values and coverage growing much faster than the number of forwarding. The latter region of the front is the one in which we are more interested, since it is providing high coverage at a reasonable number of forwarding and energy requirements.

We can see that AEDB-MLS solutions are very close to the best solutions found by the MOEAs. Additionally, it can be seen that AEDB-MLS provides a set of diverse solutions, well spread along the Pareto front approximation. We compared the Pareto fronts obtained by the three algorithms in terms of *spread* (to quantify the diversity of solutions), *inverted generational distance* (that measures the accuracy of solutions), and *hypervolume* (accounting for both, accuracy and diversity). They are defined next:

- *Inverted generational distance.* It measures the average euclidean distance from the found solutions to the Pareto-front. It was presented in [338] and defined in Equation **??**:

$$IGD = \frac{1}{|P^*|} \sum_{v \in P^*} d(v, P) \tag{9.2}$$

  where d_i is the Euclidean distance from point i in the Pareto front approximation found to the closest one in the optimal Pareto front, and n is the number of solutions in the front.

  Fronts with small *inverted generational distance* values are desirable. It takes value 0 when all solutions are actually on the Pareto front.

- *Spread.* It quantifies the diversity of solutions in the front by means of how well they are spread along the front. It is defined as:

$$I_{\Delta} = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}} \quad , \tag{9.3}$$

  where d_i is the Euclidean distance between consecutive solutions, $\bar{d}$ is the mean of these distances, and d_f and d_l are the Euclidean distances to the *extreme* solutions of the optimal Pareto front in the objective space. This indicator takes value zero for an ideal distribution, which has a perfect spread of the solutions in the Pareto front.

- *Hypervolume.* This indicator calculates the volume, in the objective space, covered by members of a non-dominated set of solutions Q, for problems where all objectives are to be minimised [383]. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated as:

$$I_{HV} = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right) . \tag{9.4}$$

  The higher the value of *hypervolume*, the better the approximated Pareto front is.

Before applying these metrics, all fronts were normalised because these indicators are not free from arbitrary scaling of the objectives. An approximation of the true Pareto front built from the best solutions found by the three considered algorithms (after 30 independent executions) was used in the normalisation process.

The results of the pairwise comparison of all the three algorithms according to the three metrics is summarised in Table 9.2. The three symbols in every column represent the result of the comparison of the corresponding algorithms for 100, 200, and 300 devices/km^2 instances, in that order.

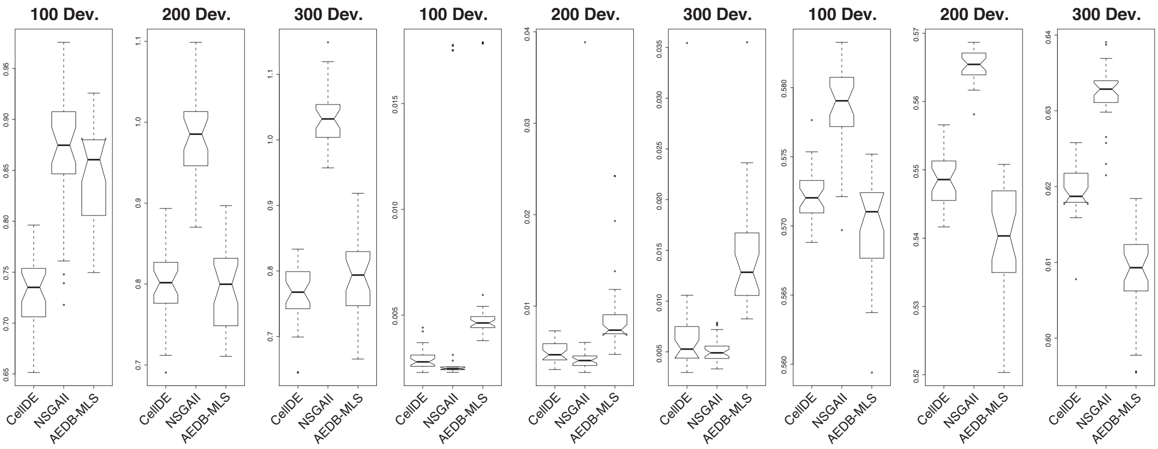
According to *spread*, CellDE outperformed NSGAII for the three instances (with statistical confidence), while it can only outperform AEDB-MLS for the sparsest density. AEDB-MLS is significantly better than NSGAII for the two biggest problem instances (no statistical confidence was found for the sparse networks). Regarding both *inverted generational distance* and *hypervolume*, AEDB-MLS is outperformed by the two MOEAs for the three instances.

We graphically show in Figure 9.5 the boxplots of the values obtained by the three algorithms in the 30 runs for the three considered metrics. We can see that, as mentioned before, AEDB-MLS is highly competitive with the MOEAs according to the *spread* metric for the three networks. The good *spread* values provided by the proposed local search algorithm, better than NSGAII, show its capability to effectively explore the search space to provide a diversified set of solutions. However, regarding the accuracy of the algorithms, it is visible that AEDB-MLS is not so competitive with the MOEAs.

We also checked how many solutions from the *Reference* Pareto front (built from the solutions of the two MOEAs) are dominated by at least one solution of the AEDB-MLS Pareto approximation and vice versa. We found out that AEDB-MLS dominates 13 solutions of the *Reference* Pareto front for the 100 devices density, while its solutions are dominated by 54 solutions of the *Reference* Pareto front. For the 200 devices network, AEDB-MLS dominates 11 solutions and is dominated 40 times. Finally, for the densest configuration, AEDB-MLS dominates 15 solutions and is dominated 17 times. The results obtained demonstrate that the quality of the solutions found using the local search is competitive compared to the MOEAs. Please, notice that we are comparing here our local search versus the best results of the two MOEAs. Therefore, we foresee that enriching the MOEAs with the proposed local search algorithm could significantly improve the quality of the obtained results.

	NSGAII	AEDB-MLS
Spread		
CellDE	▲▲▲	▲ – –
NSGAII		– ▽▽
Inverted generational distance		
CellDE	▽▽ –	▲▲▲
NSGAII		▲▲▲
Hypervolume		
CellDE	▽▽▽	▲▲▲
NSGAII		▲▲▲

Table 9.2: Comparison of the algorithms according to Wilcoxon test



(a) *Spread* (b) *Inverse generational distance* (c) *Hypervolume*

Figure 9.5: Boxplot comparison of the quality of the Pareto fronts obtained for the studied densities

Regarding the execution time, AEDB-MLS requires, in average, 48, 188, and 417 minutes to find the Pareto front approximations for the three network densities: 100, 200, and 300 devices/km^2, respectively. The evolutionary algorithms take 32, 123, and 264 hours on the same server (Intel Xeon L5640 under Debian 6.0). It means that the multi-objective local research presented here is over 38 times faster than any of the evolutionary algorithms, and it performs 2.4 times more evaluations than the EAs.

9.5 Selecting solutions from the Pareto front

The main goal of the multi-objective optimisation process is to have a complete view of the protocol behaviour in terms of the different objectives. So that depending on the situation or the needs the designer can tune the protocol for promoting one objective or another. Additionally, this process allows him/her to fine tune the protocol for different scenarios or even doing it adaptive to the circumstances.

All the solutions in the Pareto front approximations are non dominated, that is, none is better than another. However, from the point of view of the protocol designer, a decision must be taken to choose the most appropriate one according to the expected performance.

As we did for previous chapter, we want to have a comprehensive idea of the protocol for all the different objectives (that is why no more constraint were added), and then focus on one small area for better understanding the behaviour of the protocol. For that, we selected some solutions from the Pareto front and applied some restriction considering we are using a broadcasting algorithm in an ad hoc network. Therefore, we consider that the coverage must be at least 80% of the total number of devices, and the number of forwarding should be less than 30%. From the remaining solutions fulfilling those constraints we compute the energy saved per forwarding (in miliWatt), and consider the ten ones with higher values. The solutions obtained and the values for their parameters are shown in Table 9.3. Solutions

	Parameters					*Objectives*		
	minD	*maxD*	*bordersT*	*marginT*	*neighboursT*	*%Es*	*%C*	*%F*
AEDB	0.0	1.0	−90.0	0.5	8	58.6	75.2	24.8
100dSol1	0.1796	0.6389	−90.9565	0.4686	24.4182	11.0	85.2	24.0
100dSol2	0.0928	0.8193	−90.5793	0.3923	24.6660	40.8	90.0	28.4
100dSol3	0.2882	0.8802	−90.5645	1.2530	28.5137	17.5	89.2	27.2
100dSol4	0.3799	0.6743	−90.6264	0.1654	25.4845	27.6	84.4	23.2
100dSol5	0.1307	0.7251	−91.4043	0.5598	16.0962	22.6	84.0	22.8
100dSol6	0.3579	0.6742	−90.5978	0.1355	34.3880	29.4	88.4	26.8
100dSol7	0.3684	0.6310	−90.6968	0.0775	22.1308	36.1	87.2	26.0
100dSol8	0.1698	0.6378	−90.9297	0.2342	26.4092	33.8	86.4	25.6
100dSol9	0.0928	0.9170	−90.5793	0.2031	21.8288	37.3	86.4	25.2
100dSol10	0.2214	0.5405	−91.2530	0.1878	47.2978	42.47	85.6	24.4
AEDB	0.0	1.0	−90.0	0.5	8	33.1	92.4	22.8
200dSol1	0.2679	0.5308	−93.5228	0.2751	32.7293	31.9	92.8	12.6
200dSol2	0.4894	0.5929	−91.9719	0.9398	10.6796	23.7	100.0	18.6
200dSol3	0.0133	0.5883	−92.2266	0.2404	12.0021	29.1	99.6	18.4
200dSol4	0.3107	0.7424	−93.6534	0.2234	29.1790	37.6	88.0	10.4
200dSol5	0.2328	0.5508	−93.0524	0.3565	19.5563	23.6	98.6	10.8
200dSol6	0.2426	0.5618	−93.3382	0.2899	44.1326	26.6	96.2	14.8
200dSol7	0.2568	0.5394	−93.7108	0.1750	31.8875	30.2	88.8	10.8
200dSol8	0.0000	0.6734	−92.9580	0.8386	40.3794	31.7	95.8	14.2
200dSol9	0.0928	0.4158	−92.9908	0.1787	45.6975	34.9	95.6	14.2
200dSol10	0.2368	0.3712	−93.6204	0.0962	10.9326	23.8	89.4	11.2
AEDB	0.0	1.0	−90.0	0.5	12	20.6	100	18.66
300dSol1	0.3304	0.9755	−93.8238	1.0391	15.5752	31.2	93.1	8.3
300dSol2	0.0043	0.6865	−92.5747	2.4238	12.1417	18.2	99.9	12.5
300dSol3	0.0844	0.3344	−92.0849	0.0176	28.0880	25.9	100.0	13.7
300dSol4	0.1242	0.6474	−94.0484	0.6994	10.2404	38.1	86.5	6.8
300dSol5	$2.81E-4$	0.7205	−93.9438	0.9199	5.5229	31.2	88.5	7.1
300dSol6	0.2361	0.7497	−93.7313	1.1220	15.3702	17.1	93.6	8.8
300dSol7	0.4102	0.5029	−94.0152	0.6055	1.9229	25.1	82.1	6.1
300dSol8	0.2188	0.3125	−93.0334	1.8721	10.9504	15.2	97.9	10.1
300dSol9	0.0212	0.9739	−93.0191	1.7999	13.2432	25.6	99.5	10.9
300dSol10	0.1300	0.9242	−92.7375	2.1095	5.1715	17.2	99.7	11.3

Table 9.3: Domain of the variables of the chosen solutions.

in **bold font** dominate the original configuration of AEDB. Additionally, the quality of the solutions appears in the last 3 columns. The first one, represents the percentage of the energy saved per forwarding. The second, the percentage of the coverage reached, and the last one, the percentage of nodes that forward the message.

The energy saved per forwarding is calculated as:

$$EgSaved = DefTx(mW) - EgPerForwarding(mW) \ ; \quad EgPerForwarding = \frac{EgUsed}{\#forwarding + 1} \ . \tag{9.5}$$

That is, the difference between the energy used in case all the nodes sending the message are using the default transmission power and the actual energy used by the protocol (in miliWatts), divided by the number of forwarding. Analysing the values of the solutions obtained we can extract several conclusions:

- As the network density grows, the value of the *borders_Threshold* decreases. This behaviour was expected as for dense networks, the dissemination of the message is easier and less percentage of nodes resending the message is needed. Thus, the smaller the area of forwarding, the lower the number of potential forwarders, what also means that the energy saved will not vary much with other densities as the new transmission power is set to reach the furthest node that is in the limit of the forwarding area.

- The value of the *neighbours_Threshold* is very high for the two sparsest densities (100 and 200 devices/km^2). In fact, this high value is disabling the mechanism for excluding neighbours from the one-hop neighbourhood. Only, in the densest network, there are 4 solutions were this adaptive behaviour is allowed. They are 300dSol5, 300dSol7, 300dSol8, and 300dSol10.

- The solutions obtaining the highest value of coverage are not necessarily the ones consuming the most (i.e., the percentage of energy saved for 300dSol3 is higher than in the case of 300dSol8), but are the ones with higher number of forwarding.

- The configurations with high values of forwarding are not always saving less energy (i.e., 100dSol2 has the highest value of forwarding as well as the highest value for the energy saved per forwarding). We must clarify that this is happening because we are showing the energy saved per forwarding. Even when the energy used is high, if the number of forwarding is also high, it could mean that many nodes forward with low transmission power. Therefore, obtaining high value of the energy saved per forwarding.

- although there are solutions for the densest network that use the neighbours_Threshold, we observe from the solutions presented that similar behaviour can be obtained using a very specific configuration of the other parameters.

Additionally, we would like to emphasise that in the table we are including the percentage of energy saved per forwarding. Meaning that, high energy saving will appear in solutions where the total energy used is not necessarily low, but the number of forwarding is high. Additionally, solutions that tend to use high transmission power but low number of forwarding will present low energy savings. That is the reason why 100devSol1 (in **bold font**) dominates AEDB but the values presented for the energy savings is lower (58,8% and 11% for AEDB and 100devSol1, respectively).

From the values obtained, we can conclude that, finding an optimal configuration of the parameters is a complex task. Even analysing the obtained values, what we can conclude there is not a general rule we can apply for obtaining a specific behaviour. Therefore, we confirm that multi-objective optimisation is needed for optimally configuring the parameters of any protocol.

9.5.1 Performance of the selected solutions

The values of the solutions showed in Table 9.3 were obtained after optimising the dissemination algorithm in ten different networks. The optimisation algorithm looks for a combination of parameters that gives the best possible performance of the protocol on those networks. Therefore, very sensitive solutions are obtained. Small changes in the value of the parameters highly influence the behaviour of the algorithm.

	10 runs			*100 runs*		
	%Es	*%C*	*%F*	*%Es*	*%C*	*%F*
AEDB	58.6	75.2	24.8	38.4	81.1	29.0
100dSol2	40.8	90.0	28.4	29.3	79.0	26.8
100dSol4	27.6	84.4	23.2	32.1	80.2	26.4
100dSol5	22.6	84.0	22.8	27.2	75.8	22.7
AEDB	33.1	92.4	22.8	32.9	93.0	15.4
200dSol2	23.7	100.0	18.6	30.6	89.4	16.2
200dSol3	29.1	99.6	18.4	28.1	93.5	16.6
200dSol5	23.6	98.6	10.8	19.1	79.0	11.4
AEDB	20.6	100.0	18.7	26.7	98.1	18.3
300dSol3	25.9	100.0	13.7	27.0	98.8	13.6
300dSol4	38.1	86.5	6.8	38.0	72.4	5.6
300dSol8	15.2	97.9	10.1	9.4	91.9	10.4

Table 9.4: Quality of selected solutions in 100 networks.

We select now some solutions from this small area of the Pareto front we are studying in depth, and test them in 100 different networks in order to see the robustness of the solutions. For the selection process, at least one of the solutions that dominate the AEDB original configuration was chosen, as well as, the solution with the highest value for the coverage and forwarding (it is always the case). Analysing the behaviour of solutions that obtained a high value of coverage using a low number of forwarding is also desired (a very sensitive solution).

The average broadcast time for 100 executions of all the above solutions is lower than 2 seconds. In Table 9.4, the quality of the selected solutions is presented. We can see the values obtained for the different objectives for the optimised 10 networks, and also when the solutions are evaluated in 100 different networks. A drop in the quality of solutions is expected as the parameters were optimised for those 10 networks.

Solutions with high coverage but low value of number of forwarding are very sensitive as they were really fine tune for those 10 networks. Therefore, their coverage values for 100 networks decrease. This is the case for 100dSol5, 200dSol5 and 300dSol8.

The biggest changes are noticed in the 100 devices configuration as it is the network where the dissemination process is harder. Therefore, good configurations require exhaustive fine tuning, and small changes really influences on the behaviour of the protocol.

In order to provide statistical confidence to our results, we employ the Wilcoxon matched-pairs signed-rank test. This method is used to check whether two data samplings belong to different populations or not. Therefore, it can be used to compute if there are statistically significant differences between the data reported by two different protocol configurations on the 100 studied networks. The null hypothesis for this test is that the median difference between pairs of observations in the underlying populations represented by samples of results provided by the protocol is zero. In Table 9.5 the comparison between the original configuration and the 3 selected solutions of each density is shown. Each solution is compared to AEDB in terms of the three different objectives. The symbol ▲ means that there is statistical difference and the solution found by the EAs is better. Contrary, ▽ represents statistical differences but the original configuration of AEDB performs better. Finally, – stands for no differences found.

The original configuration of AEDB was experimentally chosen. Therefore, different configurations were manually tested and the best one was chosen. From Table 9.5, we can see

	EnergyUsed	*Coverage*	*forwarding*
100dSol2	–	–	–
100dSol4	▲	–	▲
100dSol5	▲	▽	▲
200dSol2	▲	▽	▲
200dSol3	▲	–	▲
200dSol5	▲	▽	▲
300dSol3	▲	–	▲
300dSol4	▲	▽	▲
300dSol8	▲	▽	▲

Table 9.5: Comparison between the AEDB original configurations and selected solutions in 100 different networks in terms of the three objectives.

that this initial setting highly promotes the number of devices reached. Indeed, none of the solutions found by the evolutionary algorithms perform better than AEDB in terms of the coverage achieved (from the three selected ones) in any density. Other solutions from the Pareto front should be chosen in case achieving high coverage is the main priority for the protocol, over the energy and the number of forwarding. On the contrary, 8 out of 9 solutions behave better than AEDB with statistical significance in terms of the other two objectives (energy and number of forwarding).

For a better understanding of the behaviour of the different solutions, we show in Figure 9.6, the relative values to the original configuration of AEDB for all the selected solutions. We plot the relative performance for the three optimised objectives as well as for the broadcast time. We should remark that all the solutions presented have an average value of the broadcasting time lower than 2 seconds. A red line marks the behaviour of AEDB using the original settings. On one hand, in case of maximisation (coverage) values above the red line means an improvement. On the other hand, if minimising (energy used, number of forwarding and broadcast time) values below the red line represent a better performance than the original configuration of AEDB.

As we mentioned before, we can see that, in average, none of the solutions outperform AEDB with the original settings in terms of the coverage achieved. Some solutions are very close to the red line, but none is over it. Oppositely, all the solutions outperform it for the two other optimised objectives (energy used and number of forwarding). Regarding the broadcast time, a couple of solutions take longer to disseminate the message but we should remark that the broadcast time was a constraint, not an objective in the optimisation process.

9.6 Large scale networks simulation

For better analysing the results obtained with the optimisation algorithms, we select all the solutions that outperform AEDB (marked in **bold font** in Table 9.3) for any density and execute them over 6 different densities and 3 different scenarios. In order to get confident results, for each density and scenario, we perform 100 executions. The results shown are the average value over those 100 independent runs.

The densities considered are: 50, 100, 200, 300, 400 and 500 devices/km^2. The three different scenarios are:

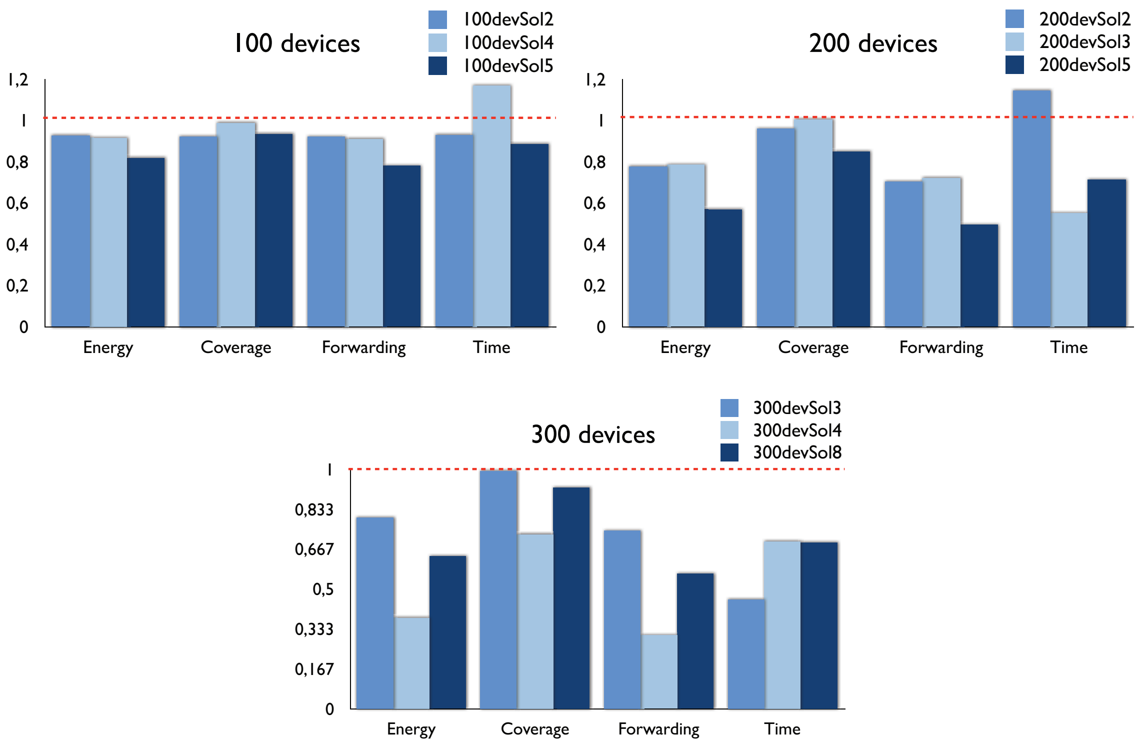


Figure 9.6: Relative value of the different solutions with respect to the original AEDB setting.

1. 500 m × 500 m → 0.25 km^2 (original scenario);
2. 500 m × 1000 m → 0.5 km^2 (medium scenario);
3. 1000 m × 1000 m → 1 km^2 (largest scenario).

Being the first one the original scenario used in the optimisation process, the second one double the size of the first, and the third one is four times bigger than the original scenario.

9.6.1 Performance over large scale networks

Below, we analyse the general behaviour of each solution in the different densities and scenarios. We first study the differences between the set of solutions of each density used by the optimisation algorithms, i.e. solutions obtained for the 100, 200 and 300 devices/km^2 configurations.

The Wilcoxon test was also applied for studying the selected solutions over different network densities and scenarios. However, in this case we are not comparing to a reference solution but we are comparing statistical differences between all the solutions. Therefore, solution with a light grey background colour means that for a specific density that solution is statistically worse than all the others, or to some of them and there are no statistical differences with the rest. Similarly, solutions with dark grey background colour means, that for the specific density is statistically better that the others, or better than some of them and there are no significant differences with the rest.

	50	100	200	300	400	500
500_500_100dSol1.txt	57.08	73.72	94.88	99.47	99.79	99.96
500_500_100dSol4.txt	56.08	80.16	97.22	99.47	99.82	99.98
500_500_100dSol5.txt	48.50	75.84	94.40	99.13	99.74	99.86
500_500_100dSol10.txt	56.83	78.00	95.20	99.24	99.51	99.93
500_500_200dSol2.txt	47.92	66.28	89.40	97.33	99.39	99.59
500_500_200dSol3.txt	47.42	63.56	93.50	96.28	96.18	99.30
500_500_200dSol5.txt	48.17	63.76	79.02	91.83	98.42	98.78
500_500_300dSol3.txt	41.00	61.36	88.82	98.77	99.16	99.84
1000_1000_100dSol1.txt	24.14	56.95	97.19	99.69	99.91	99.97
1000_1000_100dSol4.txt	24.36	62.95	95.86	99.81	99.95	99.99
1000_1000_100dSol5.txt	25.08	55.45	96.54	99.04	99.88	99.95
1000_1000_100dSol10.txt	21.92	64.45	98.38	99.71	99.92	99.94
1000_1000_200dSol2.txt	19.20	43.10	91.18	98.88	98.82	99.79
1000_1000_200dSol3.txt	18.52	48.10	88.00	97.19	99.41	99.80
1000_1000_200dSol5.txt	17.80	33.96	75.23	92.70	96.98	99.52
1000_1000_300dSol3.txt	16.54	41.00	90.64	97.96	99.73	99.94
500_1000_100dSol1.txt	33.20	65.06	97.51	99.59	99.79	99.95
500_1000_100dSol4.txt	43.00	64.66	98.19	99.65	99.95	99.94
500_1000_100dSol5.txt	32.28	59.18	93.74	98.37	99.72	99.82
500_1000_100dSol10.txt	35.88	64.82	91.68	99.01	99.82	99.92
500_1000_200dSol2.txt	30.00	53.96	90.97	95.97	99.34	99.66
500_1000_200dSol3.txt	34.08	51.86	83.13	95.96	99.22	99.46
500_1000_200dSol5.txt	31.32	44.42	73.50	85.14	97.50	99.10
500_1000_300dSol3.txt	28.40	50.30	82.57	98.55	99.53	99.86

Table 9.6: Percentage of the coverage achieved

The results obtained for the percentage of the coverage achieved are shown in Table 9.6. We can see that for the densest networks, i.e. from 300 devices/km^2 and up, the coverage achieved is very high independently of the scenario studied. For the three first densities, the best values are obviously obtained for the scenario that was optimised (500 $\times$ 500 m). Additionally, we can also see that the solutions obtained from the 100 devices optimisation are the ones behaving better. Indeed, statistical analysis reveals that for all the studies scenarios, those solutions (100dSol1, 100dSol4, 100dSol5 and 100dSol10) are usually statistically better than the others for all densities. From those 4 solutions, concerning the percentage of coverage achieved and the statistical results we can say that 100dSol4 is the one behaving better followed by 100dSol1.

The percentage of nodes forwarding the broadcast message is shown in Table 9.7. The solutions obtained from the optimisation process of 200 and 300 devices configuration are the ones that, in general, report less number of forwarding nodes. The solutions scale very well considering the number of forwarding nodes. As expected, the number of forwarding nodes decreases as the density increases.

Among the three different scenarios, the behaviour is similar. From Table 9.7, it is possible to see that in the sparsest densities the 500 $\times$ 500 m scenario uses a higher number of forwarding nodes. But this is due to the higher coverage obtained. Indeed, the number of forwarding nodes in terms of the number of nodes reached was also studied and the behaviour was similar.

The statistical results show what we already mentioned: for all the studied scenarios and densities 200dSol2, 200dSol3, 200dSol5 and 300dSol3 are usually statistically better than the others or there are no significant differences. However, from all of them, the 200dSol5 and 300dSol3 are the ones that outperforms the others with higher frequency.

	50	100	200	300	400	500
500_500_100dSol1.txt	24.00	22.64	19.50	15.39	12.21	10.24
500_500_100dSol4.txt	25.17	26.44	21.62	16.03	12.65	10.86
500_500_100dSol5.txt	17.67	22.68	18.32	14.31	11.56	10.36
500_500_100dSol10.txt	22.75	23.92	19.78	14.65	11.85	9.74
500_500_200dSol2.txt	15.92	16.88	16.20	13.43	11.54	10.04
500_500_200dSol3.txt	17.92	16.00	16.60	12.49	10.10	9.11
500_500_200dSol5.txt	13.92	14.12	11.40	10.60	9.28	7.48
500_500_300dSol3.txt	13.25	16.80	16.32	13.64	10.80	8.88
1000_1000_100dSol1.txt	9.82	18.19	20.21	14.87	11.80	9.93
1000_1000_100dSol4.txt	11.26	21.73	21.17	15.91	12.48	10.68
1000_1000_100dSol5.txt	9.46	16.75	18.79	14.15	11.68	10.55
1000_1000_100dSol10.txt	8.84	20.74	20.12	14.60	11.60	9.45
1000_1000_200dSol2.txt	6.40	11.65	16.50	13.97	11.87	10.45
1000_1000_200dSol3.txt	6.36	13.08	15.32	12.71	10.84	9.66
1000_1000_200dSol5.txt	4.84	7.28	11.61	10.64	8.86	7.55
1000_1000_300dSol3.txt	5.22	11.52	16.68	13.05	10.41	8.68
500_1000_100dSol1.txt	13.64	20.50	20.77	15.30	11.98	10.18
500_1000_100dSol4.txt	19.12	22.36	21.88	16.16	12.78	10.88
500_1000_100dSol5.txt	12.12	17.42	18.59	14.54	11.79	10.44
500_1000_100dSol10.txt	15.40	20.88	18.77	15.01	11.76	9.68
500_1000_200dSol2.txt	9.96	14.44	16.80	13.45	11.70	10.31
500_1000_200dSol3.txt	11.92	13.62	14.68	12.68	10.80	9.66
500_1000_200dSol5.txt	9.20	9.92	11.08	9.65	9.00	7.64
500_1000_300dSol3.txt	9.80	14.18	15.22	13.34	10.74	8.88

Table 9.7: Percentage of forwarding nodes

Concerning the energy consumption, in Table 9.8 we show the percentage of the energy saved per forwarding (in miliWatts).

We can see that although the highest values of energy savings are within the optimised scenario, 500 *times* 500 m, for the other two scenarios we also find high savings. The highest values are found in the sparsest density, however, this is normally due to the low coverage and low number of forwarding nodes.

Statistically, the solution that behaves better in general is 300dSol3. Additionally 200dSol2 also presents statistically better performance in some cases of the densest networks. The worst solutions with significant differences are 100dSol1 and then, 200dSol5.

We have also done studies about the broadcasting time. In Table 9.9, the time needed to broadcast a message in the network is shown. As we explained before, the broadcast time is not optimised in this work but it is considered as a constrain by the optimisation algorithms. We already explained, that during the optimisation process, the algorithms consider a solution is not feasible if its broadcast time is longer than 2 seconds. This value was obtained from preliminary work [283].

The results show that none of the solutions are over 2 s when studying the original scenario of 500m $\times$ 500m (except 200dSol2 in the 300m 400 and 500 devices configuration). Therefore, the solutions are scaling well regarding the broadcast time when dealing with lower or higher

	50	100	200	300	400	500
500_500_100dSol1.txt	45.37	30.00	13.39	14.51	20.46	8.74
500_500_100dSol4.txt	68.37	30.00	14.02	27.05	12.45	15.82
500_500_100dSol5.txt	72.97	27.16	28.33	11.37	17.57	15.81
500_500_100dSol10.txt	54.76	18.89	12.19	5.83	7.54	21.23
500_500_200dSol2.txt	48.14	42.36	30.56	29.00	23.34	27.82
500_500_200dSol3.txt	72.31	16.61	28.13	24.19	31.16	28.60
500_500_200dSol5.txt	60.94	40.54	19.08	4.80	22.58	20.34
500_500_300dSol3.txt	70.81	51.00	34.49	27.02	10.81	25.57
1000_1000_100dSol1.txt	32.74	22.64	8.55	4.93	7.39	9.93
1000_1000_100dSol4.txt	43.46	17.94	11.94	5.71	4.15	14.01
1000_1000_100dSol5.txt	32.27	13.24	6.82	8.05	14.59	25.83
1000_1000_100dSol10.txt	50.51	16.43	12.48	4.78	6.46	6.74
1000_1000_200dSol2.txt	57.49	13.69	3.33	11.96	23.50	29.40
1000_1000_200dSol3.txt	58.53	28.89	9.18	14.38	19.99	28.32
1000_1000_200dSol5.txt	58.13	31.78	13.80	2.58	6.08	3.33
1000_1000_300dSol3.txt	54.14	26.17	14.04	12.13	6.79	7.46
500_1000_100dSol1.txt	60.51	30.72	8.38	3.16	2.49	11.93
500_1000_100dSol4.txt	45.61	33.35	10.32	14.29	9.02	16.84
500_1000_100dSol5.txt	67.05	19.20	10.99	6.26	14.25	27.39
500_1000_100dSol10.txt	44.50	27.59	10.78	10.90	9.32	12.45
500_1000_200dSol2.txt	56.84	33.56	7.42	20.02	23.03	25.07
500_1000_200dSol3.txt	38.84	20.63	13.40	21.05	17.90	28.58
500_1000_200dSol5.txt	66.71	15.64	25.67	13.25	1.42	15.51
500_1000_300dSol3.txt	66.74	42.30	23.58	20.85	12.10	14.52

Table 9.8: Percentage of the energy saved per forwarding

densities. As expected, for the other two larger scenarios (500m × 1000m and 1000m × 1000m), this restriction generally fails for all solutions. In general the larger the scenario, the longer the broadcasting time. Only 300dSol3 is the one that keeps the broadcast time below the 2 seconds for the 2 scenarios in all densities.

9.6.2 Best scalable solutions

As we previously mentioned, the result of any multi-objective technique is not an specific value but a set of them that makes the dissemination algorithm promote one objective or another. As the designer of the broadcasting protocol, we are looking for a value of each threshold that generally makes the algorithm behaving better than other values in any kind of scenario for the three objectives. For example, obtaining high savings in energy but poor coverage is not a good solution from the designer point of view. Therefore, in this work, we try to find a tradeoff between the different objectives.

The study made showed the expected behaviour of the original scenario (500 *times* 500 m): the solutions found using the 100 devices network, i.e. 100dSol1, 100dSol4, 100dSol5 and 100dSol10 promote the coverage sacrificing the number of nodes forwarding and the energy saved. That is the desired behaviour as this is a quite sparse network and the dissemination process is more difficult. However, the solutions found in denser networks, i.e. 200dSol2, 200dSol3, 200dSol5 and 300dSol3 promote reducing the number of nodes forwarding. This is clearly seen in tables 9.6 and 9.7. This behaviour is observed for any of the scenarios studied.

After the analysis, we can see that if we want to promote the coverage achieved the four first solutions are the best one. On the contrary, for promoting the number of forwardings and the energy saved, we should select one of the four last solutions.

We could foresee an adaptive algorithm that uses a smart mechanism for selecting different optimal configurations depending on the circumstances and/or requirements.

	50	100	200	300	400	500
500_500_100dSol1.txt	6.2802E008 ±5.5523E008	1.0600E009 ±5.6299E008	1.2083E009 ±3.4851E008	1.0873E009 ±2.7931E008	1.0370E009 ±2.7279E008	9.9507E008 ±2.2238E008
500_500_100dSol4.txt	7.8577E008 ±7.1649E008	1.6595E009 ±8.1547E008	1.6875E009 ±5.2423E008	1.6705E009 ±4.4611E008	1.6445E009 ±3.9314E008	1.5539E009 ±3.3688E008
500_500_100dSol5.txt	4.7143E008 ±5.1031E008	1.2503E009 ±6.3196E008	1.3605E009 ±5.7923E008	1.2271E009 ±3.7271E008	1.1851E009 ±3.6312E008	1.1217E009 ±2.5350E008
500_500_100dSol10.txt	5.4429E008 ±4.5984E008	1.0291E009 ±5.2180E008	1.2588E009 ±4.7901E008	1.1579E009 ±2.9422E008	1.0984E009 ±2.7771E008	1.0194E009 ±2.3924E008
500_500_200dSol2.txt	6.0855E008 ±6.6160E008	1.2870E009 ±7.6254E008	1.9024E009 ±6.0208E008	2.0569E009 ±5.5163E008	2.0297E009 ±4.5807E008	2.0599E009 ±4.7540E008
500_500_200dSol3.txt	3.3620E008 ±3.7138E008	6.3796E008 ±4.6233E008	9.1548E008 ±4.4445E008	7.8755E008 ±3.3667E008	7.7167E008 ±3.0967E008	7.1182E008 ±2.2063E008
500_500_200dSol5.txt	3.9379E008 ±3.6382E008	8.7262E008 ±5.8194E008	1.1800E009 ±5.4009E008	1.3756E009 ±5.0803E008	1.3502E009 ±4.0152E008	1.1930E009 ±3.2879E008
500_500_300dSol3.txt	1.5746E008 ±2.0568E008	4.5995E008 ±3.3589E008	6.8652E008 ±2.8660E008	6.2671E008 ±1.8885E008	6.0969E008 ±1.5205E008	5.6516E008 ±1.3858E008
1000_1000_100dSol1.txt	1.0276E009 ±8.7444E008	2.6721E009 ±1.7380E009	2.9992E009 ±7.4697E008	2.3945E009 ±4.7044E008	2.2456E009 ±3.8067E008	2.1920E009 ±3.3605E008
1000_1000_100dSol4.txt	1.3887E009 ±1.2339E009	3.9501E009 ±1.9894E009	4.0137E009 ±9.8040E008	3.5744E009 ±5.0548E008	3.4488E009 ±5.3582E008	3.5205E009 ±5.1168E008
1000_1000_100dSol5.txt	1.0199E009 ±8.4806E008	2.6238E009 ±1.5185E009	3.0532E009 ±7.3613E008	2.5602E009 ±4.9938E008	2.4579E009 ±4.3202E008	2.3204E009 ±3.5041E008
1000_1000_100dSol10.txt	8.6453E008 ±9.5478E008	2.8506E009 ±1.5637E009	2.8817E009 ±6.8224E008	2.5613E009 ±4.3662E008	2.2964E009 ±3.8654E008	2.2621E009 ±3.2175E008
1000_1000_200dSol2.txt	9.6618E008 ±8.8038E008	2.8136E009 ±1.9550E009	5.0243E009 ±1.4114E009	4.7225E009 ±8.8304E008	4.5851E009 ±7.4599E008	4.5636E009 ±5.8274E008
1000_1000_200dSol3.txt	5.1838E008 ±5.2973E008	1.5611E009 ±1.0387E009	2.1271E009 ±6.7200E008	1.7812E009 ±4.9904E008	1.5795E009 ±4.1436E008	1.5047E009 ±3.6946E008
1000_1000_200dSol5.txt	5.9771E008 ±4.6408E008	1.6197E009 ±1.1134E009	2.9388E009 ±1.2515E009	3.1838E009 ±9.8294E008	2.7496E009 ±5.2301E008	2.6568E009 ±1.9292E008
1000_1000_300dSol3.txt	3.1163E008 ±3.0345E008	9.9234E008 ±7.5120E008	1.6022E009 ±5.3237E008	1.4123E009 ±3.1443E008	1.2302E009 ±2.3067E008	1.0883E009 ±1.6939E008
500_1000_100dSol1.txt	7.7551E008 ±7.1732E008	1.7116E009 ±8.5237E008	2.3352E009 ±6.0109E008	2.0180E009 ±4.6179E008	1.7038E009 ±3.4287E008	1.7143E009 ±3.4983E008
500_1000_100dSol4.txt	1.2317E009 ±9.5234E008	2.4814E009 ±1.4911E009	3.2619E009 ±7.9193E008	2.9041E009 ±6.5887E008	2.6950E009 ±5.4730E008	2.7485E009 ±5.4653E008
500_1000_100dSol5.txt	7.0181E008 ±7.2967E008	1.6303E009 ±1.0515E009	2.2780E009 ±6.6804E008	1.9377E009 ±5.5033E008	1.9370E009 ±4.1161E008	1.8450E009 ±4.3853E008
500_1000_100dSol10.txt	8.1295E008 ±7.0898E008	1.6994E009 ±8.9772E008	2.2036E009 ±7.5509E008	1.9222E009 ±4.3763E008	1.8257E009 ±3.4859E008	1.7702E009 ±3.7212E008
500_1000_200dSol2.txt	7.8010E008 ±7.2280E008	2.1078E009 ±1.1555E009	3.5095E009 ±9.9124E008	3.4466E009 ±8.8993E008	3.7474E009 ±7.9074E008	3.5252E009 ±7.3662E008
500_1000_200dSol3.txt	5.0355E008 ±4.6608E008	1.0711E009 ±6.6918E008	1.4913E009 ±6.6223E008	1.3452E009 ±4.2782E008	1.2464E009 ±3.8913E008	3.4029E009 ±6.9422E008
500_1000_200dSol5.txt	5.5568E008 ±5.0471E008	1.1658E009 ±7.5317E008	1.9573E009 ±8.2046E008	2.0409E009 ±7.4336E008	2.1575E009 ±5.4316E008	2.1218E009 ±5.1904E008
500_1000_300dSol3.txt	2.9376E008 ±2.7158E008	6.9014E008 ±5.1023E008	1.1363E009 ±5.0506E008	1.1066E009 ±2.9291E008	9.6382E008 ±1.9634E008	9.0663E008 ±1.8044E008

Table 9.9: Broadcast Time

Part V

Conclusions and Perspectives

Chapter 10

Conclusions and Perspectives

Contents

10.1 Summary

In this thesis, we study the network wide broadcast problem over mobile ad hoc networks. It consists of efficiently disseminating a message in the network, overcoming the drawbacks related to MANETs. We do not only tackle the problem from two different perspectives, using fixed and variable transmission range, but also present a framework for evaluating, validating and optimising the proposed algorithms. Concerning the setup of the algorithm, we also proposed an iterative local search algorithm for finding the best possible configuration, and validated it using two well known state of the art metaheuristics. Near optimal configurations are computed by the use of these highly accurate multi-objective optimisation algorithms, accounting for coverage, energy used, bandthwidth, and broadcasting time. A selection of such solutions is carefully done, and they are evaluated on a number of networks with different densities to analyse their overall performance on different situations.

The major contributions contained in this PhD thesis are:

1. An extensive state of the art in broadcasting protocols and energy aware algorithms is presented, providing also a new taxonomy. Additionally, a literature review in different optimisation techniques for problems dealing with ad hoc network is included.

2. Creation of an experimental framework that gives clarity and modularity to the development process. It efficiently evaluates, validates and optimises any algorithm under different circumstances or using any tool.

3. The expenses of the creation and maintenance of a decentralised tree based topology in different types of ad hoc networks (mobile and vehicular) is studied.

4. Two different broadcasting algorithms are proposed, and compared to state of the art broadcasting techniques. The first approach relies on a tree topology, and the second one uses a cross layer design for reducing the energy consumption.

5. Optimisation of the proposed broadcasting algorithm using a massively parallel iterated local search algorithm specifically designed for finding the best possible tradeoff configurations of AEDB.

6. Comparison the proposed local search and two state of the art evolutionary algorithms to validate the quality of its results in terms of some well known quality metrics: hypervolume, spread and epsilon.

10.1.1 The experimental framework

We present a experimental framework for mitigating the difficulties of designing algorithms in mobile ad hoc network. There exist different steps in the development of an algorithm. First, the protocol must be implemented and evaluated over different use cases in order to test it. Secondly, it must be validated by comparing to other state of the art protocols. Finally, if the algorithm contains different threshold values that have been experimentally chosen, there is another required step: the optimisation of the algorithm.

In order to give clarity and modularity to the development process, we are proposing an experimental framework for efficiently evaluate, validate and optimise any algorithm designed for MANETs. It is composed of several modules that can be easily plugged in or unplugged during the development step. All the modules composing the framework must be tightly integrated, as in most cases they must collaborate to solve the problems at hands.

10.1.2 Expenses of tree topology creation and maintenance in MANETs/-VANETs

The use of spanning trees has been widely studied in the literature for fixed and mobile ad hoc networks, but establishing a tree topology in a VANET may lead us to a contradiction. On the one hand, the restricted mobility of devices, all moving one after the other with similar speeds (thus, a low relative speed) and higher coverage range than MANETs make us thinking that it is an appropriate scenario for the use of a tree topology. On the other hand, the high speed of devices, overtaking each other and meeting devices circulating in different directions provoke that some researchers do not think it is a proper environment for dealing with trees. As having a tree topology working over MANETs is strongly accepted in the research community, we checked by making realistic experiments its suitability for vehicular networks. For that, we compare the resources needed for maintaining a tree in MANETs and VANETs. In our experiments, we consider 3 different scenarios specifically design for Luxembourg city, in order to see how the maintenance of the trees evolve in each situation. In order to show the validity of using tree topologies over VANETs, we study the stability of the tree in a network by studying the behaviour of its links. For that, we are considering three different aspects of the network. The first aspect considered is a measure of the number of tree links created and broken during the simulation. The second aspect is the total number of messages exchanged for creating the tree. And finally, we the volatility of the tree links is also considered. It refers to the number of connections and disconnections a concrete link suffers during its life time. Results show that the expenses for creating and maintaining tree topologies in VANETs and MANETs are similar.

10.1.3 Different broadcasting approaches

According to the taxonomy presented in this thesis, the are mainly two different approaches: using a fixed transmission power, or allowing devices to adjust it. We are designing a broadcasting algorithm for each of those proposals. We first propose BODYF, an efficient broadcasting protocol specifically designed for highly dynamic ad hoc communication networks, that relies on a spanning forest. The algorithm is able to successfully disseminate the message over different network partitions, that is, over different trees in the network.

The second approach proposed, AEDB uses a cross-layer design for efficiently manage the transmission power of the device. In such constrained networks, we are promoting energy savings by discarding 1-hop neighbours in dense networks. Results show the effectiveness in reducing the energy consumption while still guaranteeing certain reachability.

10.1.4 The optimisation of AEDB by means of a local search and evolutionary algorithms

The adaptability of the broadcasting algorithms is usually obtained by relying on different thresholds, so that the behaviour of the protocol adapts to the current circumstances. The value of those thresholds directly influences on the performance of the algorithms. Fine tuning these values is a complex and critical task, as protocols are usually highly sensitive to small changes. Slightly changing the value of one threshold can provoke a considerable change in its behaviour. Thus, much effort must be put in the process of selecting these values.

We are using three multi-objective approximation algorithms in order to find the best possible configuration of AEDB. The first one, AEDB-MLS, is a parallel multi-objective iterated local search specifically designed for optimising AEDB using knowledge obtained by the sensitivity analysis applied to study the influence of the values of the different variables on the performance of AEDB. Then, we are using two state of the art evolutionary algorithms, namely NSGAII and CellDE, in order to validate the results obtained by the local search.

All the optimisation algorithms found solutions that outperform the original configuration for all considered objectives. Among the set of the best solutions found, different ones were chosen any analysed in larger networks and densities to study the behaviour of the algorithm.

10.1.5 Local search versus evolutionary algorithms

The results obtained with the iterated local search were validated with the two EAs. The solutions obtained were compared in terms of the *inverted generational distance*, the *spread* and the *hypervolume* metrics, which provide a quality value of the Pareto fronts according to accuracy, diversity, and both of them together, respectively. Additionally, the algorithms were also compared in terms of the execution time.

AEDB-MLS only outperforms NSGAII in terms of the *spread* metric for the two denser networks. And both EAs outperform AEDB-MLS according to *inverted generational distance* and *hypervolume*. However, the EAs take much longer to get the results. In fact, the multi-objective iterated local research presented here is over 38 times faster than any of the evolutionary algorithms, and it performs 2.4 times more evaluations than the EAs. In the case that highly accurate solutions are required, it is possible to easily combine the EAs with AEDB-MLS by using it as a local search step to refine solutions after applying the genetic operators.

10.2 Future research lines

There are still some open issues we would like to address in the future. We have in mind, looking for robust and reliable solutions that are able to ensure the performance of the algorithm at a given predefined level is desirable. In order to find these robust solutions, apart from a different optimisation process, solutions must be evaluated over a large number of networks, what considerably increases the execution time. Therefore, advanced parallel multi-objective algorithms are needed.

Additionally, the design of an adaptive algorithm that changes the complete set of values of the thresholds is also key. The idea is to create a smart mechanism that is able to choose the right configuration from a set of optimal ones, in order to adapt to circumstances and/or requirements.

However, once the optimisation problem has been studied and we have the complete view of the problem, we could also use a massively parallel approximation algorithm that just focuses on a targeted area that we specify. In this case, applying the constraints before hands in a weighted function.

Concerning the tree based topology algorithm, we would like to compare its creation and maintenance cost to other underlying topologies as clusters or connected dominating sets.

References

[1] Infographic: Mobile statistics, stats & facts 2011. http://www.digitalbuzzblog.com/2011-mobile-statistics-stats-facts-marketing-infographic. 1

[2] The ns3 project website, http://www.nsnam.org/. 75

[3] Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- common specifications part 3: Media access control (mac) bridges. *ANSI/IEEE Std 802.1D, 1998 Edition*, pages i–355, 1998. doi: 10.1109/IEEESTD.1998.95619. 41

[4] IEEE standard 802.1d-2004: IEEE standard for local and metropolitan area networks: Media access control (mac) brideges, June 2004. 81

[5] Geonet handbook. GeoNet Deliverable D8.3, 2010. 16

[6] Car 2 car communication consortium. Available at http://www.car-to-car.org/, Last accessed September 2012. 16

[7] coopers project (2006-2011). Available at http://www.coopers-ip.eu/index.php?id=2, Accessed July 2012. 16

[8] Hart communication foundation. Available at http://www.hartcomm.org/hcf/aboutorg/aboutorg.htm Last accessed September 2012. 22

[9] Wi-fi alliance. Available at http://www.wi-fi.org/, Last accessed September 2012. 12

[10] Zigbee alliance. Available at www.zigbee.org, Last accessed September 2012. 20

[11] National institute of standards and technology (nist). Available at http://www.nist.gov/, Last accessed September 2012. 19

[12] Paramics microsimulation, Last accessed April 2013. URL `Availableathttp://www.paramics.com`. 77

[13] Cvis project. Available at http://www.cvisproject.org, Accessed July 2012. 16

[14] ISA 100.11a 2011 Standard. Wireless systems for industrial automation: Process control and related applications, isa-100, 2011. 22

[15] Facebook Statistics (March 2012). Available at http://newsroom.fb.com/content/default.aspx?newsareaid=22, Accessed July 2012. 11

[16] IEEE 802.11p published standard. Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pages 1–51, 2010. 15

[17] R. F. Abdel-Kader. Hybrid discrete pso with ga operators for efficient qos-multicast routing. *Ain Shams Engineering Journal*, 2(1):21 – 31, 2011. ISSN 2090-4479. 61

[18] W. Abdou, A. Henriet, C. Bloch, D. Dhoutaut, D. Charlet, and F. Spies. Using an evolutionary algorithm to optimize the broadcasting methods in mobile ad hoc networks. *Journal of Network and Computer Applications*, 34:1794–1804, 2011. 57

[19] N. Abramson. The aloha system: another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285, Houston, Texas, November 1970. ACM. 7

[20] N. Abramson. Development of the alohanet. *Information Theory, IEEE Transactions on*, 31(2):119–123, 1985. 7

[21] A.S. Ahluwalia and E.H. Modiano. On the complexity and distributed construction of energy-efficient broadcast trees in wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 4(5):2136–2147, 2005. 49

[22] J.-H. Ahn and T.-J. Lee. A multipoint relay selection method for reliable broadcast in ad hoc networks. In *ICT Convergence (ICTC), 2011 International Conference on*, pages 616–617, 2011. 38, 39

[23] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002. 17

[24] S.O. Al-Humoud, L.M. Mackenzie, and J. Abdulai. Neighbourhood-aware counter-based broadcast scheme for wireless ad hoc networks. In *GLOBECOM Workshops, 2008 IEEE*, pages 1–6, 2008. 31, 34

[25] L. Al-Kanj and Z. Dawy. Offloading wireless cellular networks via energy-constrained local ad hoc networks. In *IEEE Global Telecommunications Conference, GLOBECOM.*, pages 1–6, 2011. 12

[26] E. Alba and B. Dorronsoro. *Cellular Genetic Algorithms*. Operations Research/Compuer Science Interfaces. Springer-Verlag Heidelberg, 2008. 50, 96, 98

[27] E. Alba, P. Bouvry, B. Dorronsoro, F. Luna, and A.J. Nebro. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs. In

Nature Inspired Distributed Computing (NIDISC) sessions of the (IPDPS) Workshop, page 192b, 2005. 131

[28] E. Alba, B. Dorronsoro, F. Luna, A. J. Nebro, P. Bouvry, and L. Hogie. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan manets. *Computer Communications*, 30(4):685–697, 2007. 98

[29] E. Alba, B. Dorronsoro, F. Luna, A.J. Nebro, P. Bouvry, and L. Hogie. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs. *Computer Communications*, 30(4):685 – 697, 2007. ISSN 0140-3664. 62

[30] ZigBee Alliance. Specifications. Available at http://www.zigbee.org/Specifications.aspx, Last accessed September 2012. 21

[31] G. Anastasi, M. Conti, E. Gregori, and A. Passarella. 802.11 power-saving mode for mobile computing in wi-fi hotspots: limitations, enhancements and open issues. *Wireless Networks*, 14(6):745–768, December 2008. ISSN 1022-0038. 46

[32] A. Andronache and S. Rothkugel. Nlwca node and link weighted clustering algorithm for backbone-assisted mobile ad hoc networks. In *Seventh International Conference on Networking*, pages 460–467, 2008. 43, 44, 90

[33] A. Andronache, P. Ruiz, and S. Rothkugel. Improving inter-cluster broadcasting in ad hoc networks by delayed flooding. In *MCO*, pages 478–487, 2008. 43

[34] T. Anwar, C. Jones, D. Schreib, and G. Strauch. Mobile pr-network in a densely populated urban environment. In *Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM '90., IEEE*, pages 1574–1578, 1990. 13

[35] J. Arango, M. Degermark, A. Efrat, and S. Pink. An efficient flooding algorithm for ad-hoc networks. In *Proceedings of the Second Workshop on Modeling and Optimizations in Mobile Ad Hoc and Wireless Networks (WiOpt)*, 2004. 32, 34

[36] S. Asadinia, M.k. Rafsanjani, and A.B. Saeid. A novel routing algorithm based-on ant colony in mobile ad hoc networks. In *3rd IEEE International Conference on Ubi-media Computing (U-Media)*, pages 77–82, 2010. 59

[37] K. Ashton. That 'internet of things' thing. *RFID Journal*, 2009. 21

[38] E.K. Asl, M. Damanafshan, M. Abbaspour, M. Noorhosseini, and K. Shekoufandeh. EMP-DSR: An enhanced multi-path dynamic source routing algorithm for MANETs based on ant colony optimization. In *Third Asia International Conference on Modelling Simulation AMS*, pages 692–697, 2009. 60

[39] N. M. Asraf, R. N. Ainon, and P. K. Keong. Qos parameter optimization using multi-objective genetic algorithm in manets. In *Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference on*, pages 138 –143, may 2010. 61

[40] ASTM. Astm e2213 - 03(2010) standard specification for telecommunications and information exchange between roadside and vehicle systems — 5 ghz band dedicated short range communications (dsrc) medium access control (mac) and physical layer (phy) specifications (2003). Available at http://www.astm.org/Standards/E2213.htm, Accessed July 2012. 15

[41] N.A. Aziz, A.W. Mohemmed, and M. Zhang. Particle swarm optimization for coverage maximization and energy conservation in wireless sensor networks. In *Applications of Evolutionary Computation*, volume 6025 of *Lecture Notes in Computer Science*, pages 51–60. Springer Berlin Heidelberg, 2010. 56

[42] T. Bäck, D.B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, 1997. 145

[43] S. Bai, Z. Huang, D. Kwak, Sa. Lee, H. Oh, and J. Jung. Vehicular multi-hop broadcasting protocol for safety message dissemination in vanets. In *Vehicular Technology Conference Fall (VTC 2009-Fall), IEEE 70th*, pages 1 –5, Anchorage, AK, Sept 2009. 38, 39

[44] A. Banerjee, C.H. Foh, C.K. Yeo, and B.S. Lee. Performance improvements for network-wide broadcast with instantaneous network information. *Journal of Network and Computer Applications*, 35(3):1162 – 1174, 2012. Special Issue on Trusted Computing and Communications¡/ce:title¿. 34

[45] J. S. Baras and H. Mehta. A probabilistic emergent routing algorithm for mobile ad hoc networks, 2003. 60

[46] A. Barolli, E. Spaho, F. Xhafa, L. Barolli, and M. Takizawa. Application of ga and multi-objective optimization for qos routing in ad-hoc networks. In *14th International Conference on Network-Based Information Systems (NBiS)*, pages 50–59, 2011. 59

[47] S. Basagni, M. Conti, S. Giordano, and I. Stojmenović. *Mobile Ad Hoc Networking*. Wiley, San Francisco, CA, USA, 2004. 19

[48] S. Basagni, M. Conti, S. Giordano, and I. Stojmenović. *Mobile Ad Hoc Networking*. Wiley-IEEE Press, 2004. ISBN 978-0-471-37313-1. 25

[49] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - simulation of urban mobility: An overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 2011. 77

[50] Abderrahim Benslimane. Optimized dissemination of alarm messages in vehicular ad-hoc networks (vanet). In *HSNMC*, pages 655–666, 2004. 111

[51] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: a media access protocol for wireless lan's. *SIGCOMM Comput. Commun. Rev.*, 24(4):212–225, October 1994. ISSN 0146-4833. 9

[52] R. C. Biradar and S. S. Manvi. Review of multicast routing mechanisms in mobile ad hoc networks. *Journal of Network and Computer Applications*, 35(1):221 – 239, 2012. ISSN 1084-8045. 60

[53] S. Bitam and A. Mellouk. Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks. *Journal of Network and Computer Applications*, 2012. doi: 10.1016/j.jnca.2012.01.023. in press. 60

[54] D.M. Blough, M. Leoncini, G. Resta, and P. Santi. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(9):1267–1282, 2006. 44

[55] P.P. Bonissone, R. Subbu, N. Eklund, and T.R. Kiehl. Evolutionary algorithms + domain knowledge = real-world evolutionary computation. *Evolutionary Computation, IEEE Transactions on*, 10(3):256 – 280, june 2006. 25

[56] A. Boukerche and V. Sheetal. A performance evaluation of a dynamic source routing discovery optimization protocol using gps system. *Telecommunication Systems*, 22(1–4): 337–354, 2003. 70

[57] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011. ISSN 1389-1286. 58

[58] K. Bulbul, O. Ercetin, and T. Unluyurt. Hop constrained energy-efficient broadcasting: Insights from massively dense ad hoc networks. *IEEE Transactions on Mobile Computing*, 8(2):188–202, 2009. 47, 49

[59] Z-Y CAO, Z-Z JI, and M-Z HU. An energy-aware broadcast scheme for directed diffusion in wireless sensor network. *Journal of Communication and Computer*, 4,(5):28–35, 2007. ISSN 1548-7709. 33, 34, 108

[60] J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *Proceedings of the 36th Hawaii International Conference on System Sciences*, pages 189–204, 2003. 32, 34

[61] J. Cartigny, D. Simplot, and I. Stojmenović. Localized minimum-energy broadcasting in ad-hoc networks, 2003. 45, 47

[62] J. Cartigny, D. Simplot-Ryl, and I. Stojmenović. An adaptive localized scheme for energy-efficient broadcasting in ad hoc networks with directional antennas. In Ignas Niemegeers and SoniaHeemstra Groot, editors, *Personal Wireless Communications*, volume 3260 of *Lecture Notes in Computer Science*, pages 399–413. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-23162-2. doi: 10.1007/978-3-540-30199-8_33. 48, 49

[63] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Localized lmst and rng based minimum-energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, 3 (1):1–16, 2005. 45, 47

[64] A. Casteigts. Model driven capabilities of the DA-GRS model. In *Intl. Conf. on Autonomic and Autonomous Systems (ICAS'06). San Francisco. USA*, page 24, 2006. 81, 82

[65] A. Casteigts and S. Chaumette. Dynamicity aware graph relabeling system - a local computation model to describe manet algorithms. In *Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and System*, pages 198–204, 2005. 42

[66] A. Casteigts and S. Chaumette. Dynamicity aware graph relabeling systems and the constraint based synchronization: a unifying approach to deal with dynamic networks. In *Intl. Conf. on Wireless Algorithms, Systems and Applications (WASA'06). LNCS - Xi'An, China*, pages 688–697, 2006. 82

[67] A. Casteigts, S. Chaumette, F. Guinand, and Y. Pigné. Distributed maintenance of anytime available spanning trees in dynamic networks. *CoRR*, abs/0904.3087, 2009. 42

[68] C. Chen, C-K Hsu, and H-K Wang. A distance-aware counter-based broadcast scheme for wireless ad hoc networks. In *IEEE Military Communications Conference (MILCOM)*, pages 1052–1058, 2005. 31, 34

[69] X. Chen, M. Faloutsos, and S. Krishnamurthy. Distance adaptive (dad) broadcasting for ad hoc networks. In *Proceedings MILCOM*, volume 2, pages 879–883, 2002. 32, 34

[70] X. Chen, M. Faloutsos, and S.V. Krishnamurthy. Power adaptive broadcasting with local information in ad hoc networks. In *Conference on Network Protocols*, page 168. IEEE Computer Society, 2003. 45, 47

[71] H. Cheng and S. Yang. Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. *Engineering Applications of Artificial Intelligence*, 23(5):806 – 819, 2010. ISSN 0952-1976. 60

[72] H. Cheng, S. Yang, and J. Cao. Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks. *Expert Systems with Applications*, 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.08.050. in press. 61

[73] T. Chiang, C. Liu, and Y. Huang. A near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm. *Expert Systems with Applications*, 33(3): 734–742, 2007. 60

[74] A. Chiganmi, M. Baysan, K. Sarac, and R. Prakash. Variable power broadcast using local information in ad hoc networks. *Ad Hoc Networks*, 6(5):675–695, July 2008. ISSN 1570-8705. 46, 47

[75] I. Chlamtac, M. Conti, and J.J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1:13–64, 2003. 9

[76] C-Y Chong and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003. 17, 18

[77] T.H. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Technical report, IETF, October 2003. 36, 38, 39

[78] A.E.F. Clementi, P. Penna, and R. Silvestri. The power range assignment problem in radio networks on the plane. In *The 17th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '00, pages 651–660, London, UK, UK, 2000. Springer-Verlag. 44, 45

[79] Federal Communications Commission. Fcc news. Available at http://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html (1999), Accessed July 2012. 14

[80] CAR 2 CAR Communication Consortium. Car 2 car communication consortium manifesto, version 1.1. overview of the c2c-cc system, 2007. 16

[81] S. L. O. B. Correia, J. Celestino, and O. Cherkaoui. Mobility-aware ant colony optimization routing for vehicular ad hoc networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1125–1130, 2011. 59

[82] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. IETF RFC 2501, 1999. URL `http://www.ietf.org/rfc/rfc2501.txt`. 63

[83] X. Cui, C. Lin, and Y. Wei. A multiobjective model for qos multicast routing based on genetic algorithm. In *Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on*, pages 49 – 53, oct. 2003. doi: 10.1109/ICCNMC.2003.1243026. 61

[84] W. Dabbous and C. Huitema. Prometheus: Vehicle to vehicle communications. Technical report, Research report. INRIA-Renault collaboration, 1988. 13

[85] F. Dai and J. Wu. Distributed dominant pruning in ad hoc networks. In *IEEE International Conference on Communications*, volume 1, pages 353–357, 2003. 40

[86] G. Danoy, P. Bouvry, and L. Hogie. Coevolutionary genetic algorithms for ad hoc injection networks design optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 4273–4280, 2007. 55

[87] Jr. Davis, M.H. and M. Baldwin. An intercommunication system for the surgical opperating room. In *1956 IRE National Convention*, pages 24–28, 1956. 17

[88] F. Davoli, A. Giordano, and S. Zappatore. Architecture and protocol design for a car-to-infrastructure packet radio network. In *Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM '90., IEEE*, pages 1579–1585, 1990. 13

[89] M. F. de Castro, L. B. Ribeiro, and C. H. S. Oliveira. An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing. *Journal of Network and Computer Applications*, 2012. ISSN 1084-8045. doi: 10.1016/j.jnca.2012.07.023. in press. 58

[90] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. 146, 147

[91] C.J. Debono and E. Sammut. Location estimation of an intruder in wireless ad hoc networks. In *Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean*, pages 158 –162, may 2008. 64

[92] O. Dengiz, A. Konak, and A. E. Smith. Connectivity management in mobile ad hoc networks using particle swarm optimization. *Ad Hoc Networks*, 9(7):1312 – 1326, 2011. ISSN 1570-8705. doi: 10.1016/j.adhoc.2011.01.010. 57

[93] G. A. Di Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *Eur. Trans. on Telecom.*, 16 (5):443–455, 2005. 58

[94] Z. Dongmei, Z. Qu, Y. Yang, and X. Feng. A service negotiation mechanism in mobile ad hoc network. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 –4, sept. 2011. 63

[95] B. Dorronsoro, G. Danoy, P. Bouvry, and E. Alba. Evaluation of different optimization techniques in the design of ad hoc injection networks. In *The 2008 International Conference High Performance Computing & Simulation (HPCS) and in conjunction with The 22nd European Conference on Modeling and Simulation (ECMS).*, 2008. 55

[96] B. Dorronsoro, P. Ruiz, G. Danoy, P. Bouvry, and L. Tardon. Towards connectivity improvement in VANETs using bypass links. In *IEEE Congress on Evolutionary Computation CEC*, pages 2201–2208, 2009. 55

[97] F. Dressler and O. B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881 – 900, 2010. ISSN 1389-1286. 50

[98] J.J. Durillo and A. J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011. ISSN 0965-9978. 146, 148

[99] J.J. Durillo, A.J. Nebro, F. Luna, and E. Alba. Solving three-objective optimization problems using a new hybrid cellular genetic algorithm. In *Parallel Problem Solving from Nature (PPSN X)*, volume 5199 of *LNCS*, pages 661–670. Springer, 2008. 146, 147

[100] A. Durresi, V.K. Paruchuri, S.S. Iyengar, and R. Kannan. Optimized broadcast protocol for sensor networks. *IEEE Transactions on Computers*, 54(8):1013–1024, 2005. 32, 34

[101] V. Dyo and C. Mascolo. Adaptive distributed indexing for spatial queries in sensor networks. In *16th International Workshop on Databases and Expert Systems Applications*, pages 1103–1107, 2005. 18

[102] A. Ephremides, J.E. Wieselthier, and .J Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73, 1987. 43, 44

[103] L.J. Eshelman and J.D. Schaffer. Real-coded genetic algorithms and interval-schemata. In *FOGA*, pages 187–202, 1992. 139

[104] M. Farooq. *Bee-inspired Protocol Engineering*. Natural Computing Series. Springer, 2009. 59

[105] K. P. Ferentinos and T. A. Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, 51(4):1031–1051, 2007. 53

[106] K. P. Ferentinos and T. A. Tsiligiridis. A memetic algorithm for optimal dynamic design of wireless sensor networks. *Computer Communications*, 33(2):250–258, 2010. 53

[107] O. Flauzac, B.S. Haggar, and F. Nolot. Tree and cluster management for manet. In *New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference on*, pages 73–80, 2010. 42

[108] HART Communication Foundation. Hart field communication protocol specification, revision 7.0, September 2007. 22

[109] M.R. Friedberg. Mobile antennas for vehicular communications. *Transactions of the IRE Professional Group on Vehicular Communications*, 1(1):100–104, 1952. 13

[110] T. Fujie. An exact algorithm for the maximum leaf spanning tree problem. *Computers and Operations Research*, 30(13):1931–1944, November 2003. ISSN 0305-0548. 41

[111] K. Fujita, A. Saito, and T. Matsui. An adaptive ant-based routing algorithm used routing history in dynamic networks. In *4th Asia-Pacific Conference on Simulated Evolution And Learning*, pages 20–40, 2002. 58

[112] S. Gai-ping, G. Hai-wen, W. Dezhi, and W. Jiang-hua. A dynamic ant colony optimization algorithm for the ad hoc network routing. In *Fourth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pages 358 –361, 2010. 58

[113] W. Gang, W. Shigang, L. Cai, and Z. Xiaorong. Research and realization on improved manet distance broadcast algorithm based on percolation theory. In *International Conference on Industrial Control and Electronics Engineering (ICICEE)*, pages 96–99, 2012. 33, 34

[114] B. Garbinato, A. Holzer, and F. Vessaz. Six-shot broadcast: A context-aware algorithm for efficient message diffusion in manets. In *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems:*, OTM '08, pages 625–638, 2008. ISBN 978-3-540-88870-3. 32, 34

[115] B. Garbinato, A. Holzer, and F. Vessaz. Context-aware broadcasting approaches in mobile ad hoc networks. *Comput. Netw.*, 54(7):1210–1228, May 2010. ISSN 1389-1286. 29

[116] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'05 special session on real parameter optimization. *Journal of Heuristics*, 15:617–644, 2009. 149

[117] J. García-Nieto, J. Toutouh, and E. Alba. Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics. *Engineering Applications of Artificial Intelligence*, 23(5):795 – 805, 2010. ISSN 0952-1976. 62

[118] F. Ge, Y. Wang, Q. Wang, and J. Kang. Energy efficient broadcasting based on ant colony system optimization algorithm in wireless sensor networks. In *Natural Computation. Third International Conference on*, volume 4, pages 129 –133, aug. 2007. 57

[119] M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*. Springer, 2010. 50, 71

[120] M. Gerla and J. T. Tsai. Multicluster, mobile, multimedia radio network. *Journal of Wireless Networks*, 1:255–265, 1995. 43, 44

[121] M. Gerla, T. Kwon, and G. Pei. On-demand routing in large ad hoc wireless networks with passive clustering. In *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*, volume 1, pages 100–105, 2000. 43, 44

[122] W.J Gillan. Prometheus and drive: their implications for traffic managers. In *Vehicle Navigation and Information Systems Conference, 1989. Conference Record*, pages 237–243, 1989. 13

[123] D. Gislason, editor. *Zigbee Wireless Networking Zigbee Wireless Networking Zigbee Wireless Networking*. Elsevier Inc., 2008. 21

[124] J. Gomez and A.T. Campbell. Variable-range transmission power control in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 6(1):87–99, 2007. ISSN 1536-1233. 44

[125] E.G. Goodaire and M.M. Parmenter. *Discrete mathematics with graph theory*. Printice-Hall, Inc., 2nd ed. edition, 2002. 85

[126] D.l Gorgen, H. Frey, and C. Hiedels. JANE–the Java Ad-hoc Network Environment. In *Proceedings of the 40th Annual Simulation Symposium*, pages 163–176, 2007. 90

[127] M.M. Goswami, R.V. Dharaskar, and V.M. Thakare. Fuzzy ant colony based routing protocol for mobile ad hoc network. In *International Conference on Computer Engineering and Technology ICCET*, volume 2, pages 438–444, 2009. 58

[128] R. B. Groenevelt, E. Altman, and P. Nain. Relaying in mobile ad hoc networks: The brownian motion mobility model. *Journal of Wireless Networks*, pages 561–571, 2006. 77, 114, 140

[129] IEEE 802.15 Working Group. 802.15.4-2011 - ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans), 2011. 20

[130] Special Interest Group. Bluetooth. Available at https://www.bluetooth.org/apps/content/, Last accessed September 2012. 21

[131] S.J. Gudakahriz, S. Jamali, and E. Zeinali. Nisr: A nature inspired scalable routing protocol for mobile ad hoc networks. *International Journal of Computer Science Engineering and Technology*, 1(4):180–184, 2011. 59

[132] M. Gunes, U. Sorges, and I. Bouazizi. Ara-the ant-colony based routing algorithm for manets. In *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 79 – 85, 2002. 60

[133] M. Günes, M. Kähmer, and I. Bouazizi. Ant-routing-algorithm (ARA) for mobile multi-hop ad-hoc networks – new features and results. In A. Belghith, S. Tabbane, N. B. Ali, and A. Gazdar, editors, *Proceedings of the 2nd Medite+anean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003)*, pages 9–20, 2003. 54

[134] S. Guo and O.W.W. Yang. Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. *Computer Communications*, 30(9):2129–2148, June 2007. ISSN 0140-3664. 44

[135] S. Guo, C. Dang, and X. Liao. Joint opportunistic power and rate allocation for wireless ad hoc networks: An adaptive particle swarm optimization approach. *Journal of Network and Computer Applications*, 34(4):1353 – 1365, 2011. ISSN 1084-8045. doi: 10.1016/j.jnca.2011.03.020. 54

[136] D. Gutiérrez-Reina, S.L. Toral Marín, P. Johnson, and F. Barrero. An evolutionary computation approach for designing mobile ad hoc networks. *Expert Systems with Applications*, 39(8):6838 – 6845, 2012. ISSN 0957-4174. 55

[137] J.C. Haartsen. The bluetooth radio system. *Personal Communications, IEEE*, 7(1): 28–36, 2000. 21

[138] R. Handorean, C.D. Gill, and G-C Roman. Accommodating transient connectivity in ad hoc and mobile settings. In *Pervasive*, pages 305–322, 2004. 81

[139] E.A. Hanysz, C.E. Quinn, J.E. Stevens, and W.G. Trabold. Dair—a new concept in highway communications for added safety and driving convenience. In *Vehicular Technology Conference, 1966. 17th IEEE*, pages 108–120, 1966. 13

[140] H. Hartenstein and K. Laberteaux, editors. *VANET Vehicular Applications and Inter-Networking Technologies*. Wiley, 2009. 13

[141] M. Heissenbüttel and T. Braun. Ants-based routing in large scale mobile ad-hoc networks. In *In Kommunikation in verteilten Systemen (KiVS03*, pages 91–99, Leipzig,Germany, 2003. 58

[142] R. Hekmat. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer, 2004. ISBN 978-1-4020-5166-1. 2

[143] H. Hernández and C. Blum. Ant colony optimization for multicasting in static wireless ad-hoc networks. *Swarm Intelligence*, 3:125–148, 2009. ISSN 1935-3812. 57

[144] H. Hernández and C. Blum. Minimum energy broadcasting in wireless sensor networks: An ant colony optimization approach for a realistic antenna model. *Applied Soft Computing*, 11(8):5684–5694, 2011. 57

[145] F. Herrera, M. Lozano, and A. M. Sanchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18(3):309–338, 2003. 139

[146] L. Hogie. The madhoc simulator. Technical report, Le Havre University, 2005. 62, 75

[147] L. Hogie. *Mobile Ad Hoc Networks, Modelling, Simulation and Broadcast-based Applications*. PhD thesis, University of Luxembourg, 2007. 98

[148] L. Hogie, F. Guinand, and P. Bouvry. A heuristic for efficient broadcasting in the metropolitan ad hoc networks. In *KES*, pages 529–536, 2004. 35, 39

[149] L. Hogie, P. Bouvry, M. Seredynski, and F. Guinand. A Bandwidth-Efficient Broadcasting Protocol for Mobile Multi-hop Ad hoc Networks. In *Demo proceeding of the 5th International Conference on Networking (ICN'06)*, page 71. IEEE, October 2006. 98

[150] P.K.G. Hoolimath, M. Kiran, and G.R.M. Reddy. Optimized termite: A bio-inspired routing algorithm for MANET's. In *International Conference on Signal Processing and Communications (SPCOM)*, pages 1 –5, 2012. 60

[151] P.-C. Hsiao, T.-C. Chiang, and L.-C. Fu. Particle swarm optimization for the minimum energy broadcast problem in wireless ad-hoc networks. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012. 57

[152] X.-M. Hu, J. Zhang, Y. Yu, H.S.-H. Chung, Y.-L. Li, Y.-H. Shi, and X.-N. Luo. Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *Evolutionary Computation, IEEE Transactions on*, 14(5):766 –781, oct. 2010. 54

[153] C. Huang, Y. Chuang, and K. Hu. Using particle swarm optimization for QoS in ad-hoc multicast. *Engineering Applications of Artificial Intelligence*, 22(8):1188–1193, 2009. 60

[154] C-M Huang, L Tu, and C-H Chou. Rewarn: An opportunistic relay scheme for cooperative collision warning in vanet. In *Personal, Indoor and Mobile Radio Communications, IEEE 20th International Symposium on*, pages 3030 –3034, Tokyo, Sept 2009. 33, 34

[155] J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, and M. Hamdi. The terminode project: towards mobile ad-hoc wans. In *IEEE International Workshop on Mobile Multimedia Communications.*, pages 124–128, 1999. 10

[156] P.A. Humblet. A distributed algorithm for minimum weight directed spanning trees. *IEEE Transactions on Communications*, 31(6):756–762, 1983. 49

[157] S. Hussain, A. W. Matin, and O. Islam. Genetic algorithm for hierarchical wireless sensor networks. *Journal of Networks*, 2(7):87–97, 2007. 61

[158] O. H. Hussein, T. N. Saadawi, and J. L. Myung. Probability routing algorithm for mobile ad hoc networks' resources management. *IEEE Journal on Selected Areas in Communications*, 23(12):2248 – 2259, dec. 2005. ISSN 0733-8716. doi: 10.1109/JSAC.2005.857205. 58

[159] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(6):536–547, 2006. 48, 49

[160] SAE Std. J2735. SAE International. Dedicated short-range communications (dsrc) message set dictionaty (nov. 2009). Available at http://standards.sae.org/j2735_200911, Accessed July 2012. 15

[161] ISA. International society of automation. Available at http://www.isa.org/, Last accessed September 2012. 22

[162] S. S. Iyengar, H.-C. Wu, N. Balakrishnan, and S. Y. Chang. Biologically inspired cooperative routing for wireless mobile sensor networks. *IEEE Systems Journal*, 1(1): 29–37, 2007. 58

[163] J.G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A.K. Saha, and D.B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 32–43, 2003. 10

[164] C. Ji, Y. Zhang, S. Gao, P. Yuan, and Z. Li. Particle swarm optimization for mobile ad hoc networks clustering. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 372 – 375 Vol.1, march 2004. 61

[165] J. Jia, J. Chen, G. Chang, J. Li, and Y. Jia. Coverage optimization based on improved nsga-ii in wireless sensor network. In *IEEE International Conference on Integration Technology*, pages 614–618, march 2007. 53

[166] E.E. Johnson, Z. Tang, M. Balakrishnan, H. Zhang, and S. Sreepuram. Routing in hf ad-hoc wans. In *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, volume 2, pages 1040–1046, 2004. 10

[167] D.B. Johnsort. Routing in ad hoc networks of mobile hosts. In *Mobile Computing Systems and Applications. First Workshop on*, pages 158–163, 1994. 11

[168] J. Jubin and J.D. Tornow. The darpa packet radio network protocols. *Proceedings of the IEEE*, 75(1), 1987. 11

[169] R. Jurdak. *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective (Signals and Communication Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387390227. 106

[170] A. Juttner and A. Magi. Tree based broadcast in ad hoc networks. *Mobile Networks and Applications*, 10:753–762, 2005. 42, 44

[171] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for ‘smart dust’. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom ’99, pages 271–278, New York, NY, USA, 1999. ACM. ISBN 1-58113-142-9. 17

[172] R. Kahn. The organization of computer resources into a packet radio network. *Communications, IEEE Transactions on*, 25(1), 1977. 10

[173] B. Kalaavathi, S. Madhavi, S. VijayaRagavan, and K. Duraiswamy. Review of ant based routing protocols for MANET. In *International Conference onComputing, Communication and Networking (ICCCN)*, pages 1–9, 2008. 58

[174] I. Kang and R. Poovendran. Iterated local optimization for minimum energy broadcast. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WIOPT)*, pages 332–341, 2005. 57

[175] K. Karenos, A. Khan, S. Krishnamurthy, M. Faloutsos, and C. Xiaohu. Local versus global power adaptive broadcasting in ad hoc networks. In *IEEE Wireless Communications and Networking Conference*, volume 4, pages 2069–2075, 2005. 46

[176] P. Karn. Maca a new channel access method for packet radio. In *Computer Networking Conference*, volume 9th, pages 134–140, 1990. 9

[177] D. Kasamatsu, N. Shinomiya, and T. Ohta. A broadcasting method considering battery lifetime and distance between nodes in manet. In *29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS)*, pages 406–411, 2009. 33, 34

[178] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks, P. Arabshahi, and A.A. Gray. Adaptive-sdr: adaptive swarm-based distributed routing. In *Neural Networks, 2002. IJCNN ’02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 351 –354, 2002. 58

[179] V. Kawadia and P.R. Kumar. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(1):76–88, 2005. 44

[180] V. Kawadia and P.R. Kumar. Principles and protocols for power control in wireless ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 23(1):76 – 88, 2005. ISSN 0733-8716. 110

[181] J.B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, July 2011. 15

[182] A.M. Kermarrec, L. Massoulié, and A.J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. Parallel Distrib. Syst.*, 14(3):248–258, March 2003. ISSN 1045-9219. 30, 34

[183] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Color-based broadcasting for ad hoc networks. In *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–10, 2006. 31, 34

[184] M. Khabbazian, I.F. Blake, and V.K. Bhargava. Local broadcast algorithms in wireless ad hoc networks: Reducing the number of transmissions. *Mobile Computing, IEEE Transactions on*, 11(3):402–413, 2012. 36, 39

[185] E. A. Khalil and B. A. Attea. Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks. *Swarm and Evolutionary Computation*, 1(4): 195–203, 2011. 61

[186] R. Khanna and L. Huaping andC. Hsiao-Hwa. Self-organization of sensor networks using genetic algorithms. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 8, pages 3377 –3382, june 2006. 61

[187] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, 2000. 138

[188] A. Kokuti and V. Simon. Adaptive multihop broadcast protocols for ad hoc networks. In *8th International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*, pages 1–6, 2012. 33, 34

[189] K. Komathy and P. Narayanasamy. Study of co-operation among selfish neighbors in manet under evolutionary game theoretic model. In *Proc. International Conference on Signal Processing, Communications and Networking (ICSCN)*, pages 133–138, 2007. 63

[190] A. Konstantinidis and K. Yang. Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific moea/d. *Applied Soft Computing*, 11(6):4117–4134, 2011. 55

[191] A. Konstantinidis and K. Yang. Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition. *Computer Communications*, 34(1):83–98, 2011. 55

[192] A. Konstantinidis, K. Yang, H.-H. Chen, and Q. Zhang. Energy-aware topology control for wireless sensor networks using memetic algorithms. *Computer Communications*, 30 (14/15):2753–2764, 2007. 55

[193] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti. A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. *Computer Networks*, 54(6):960–976, 2010. 55

[194] D. Koscielnik and J. Stepien. The methods of broadcasting of information in ad-hoc wireless networks with mobile stations. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, pages 2043–2048, 2011. 29

[195] R. Kravets and P. Krishnan. Power management techniques for mobile communication. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, pages 157–168, New York, NY, USA, 1998. ACM. ISBN 1-58113-035-X. 70, 105

[196] P. Kromer, J. Martinovic, M. Radecky, R. Tomis, and V. Snasel. Ant colony inspired algorithm for adaptive traffic routing. In *Third World Congress on Nature and Biologically Inspired Computing*, pages 329–334, 2011. 62

[197] R. Kulandaivel, S. Periyanayagi, and S. Susikalac. Performance comparison of wsn & wsan using genetic algorithm. In *International Conference on Communication Technology and System Design 2011*, volume 30, pages 107–112, 2012. 56

[198] R.V. Kulkarni and G.K. Venayagamoorthy. Particle swarm optimization in wireless-sensor networks: A brief survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(2):262 –267, march 2011. ISSN 1094-6977. 56

[199] J. Kusyk, C. S. Sahin, M. U. Uyar, E. Urrea, and S. Gundry. Self-organization of nodes in mobile ad hoc networks using evolutionary games and genetic algorithms. *Journal of Advanced Research*, 2:253–264, 2011. 56

[200] M. Lacage and T.R. Henderson. Yet another network simulator. In *WNS2 '06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, page 12, 2006. ISBN 1-59593-508-8. 75, 113

[201] S-H Lee and Y-B Ko. An efficient neighbor knowledge based broadcasting for mobile ad hoc networks. In *Proceedings of the 6th international conference on Computational Science - Volume Part II*, ICCS'06, pages 1097–1100, Berlin, Heidelberg, 2006. Springer-Verlag. 35, 39

[202] J. Leitao, J. Pereira, and L. Rodrigues. Epidemic broadcast trees. In *26th IEEE International Symposium on Reliable Distributed Systems*, pages 301–310, 2007. 42, 44

[203] J. Leitao, J. Pereira, and L. Rodrigues. Gossip-based broadcast. In Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon, editors, *Handbook of Peer-to-Peer Networking*, pages 831–860. Springer US, 2010. ISBN 978-0-387-09750-3. 30

[204] S. Leu and R-S Chang. Simple algorithm for solving broadcast storm in mobile ad hoc network. *Communications, IET*, 5(16):2356–2363, 2011. 40

[205] D. Li, H. Huang, X. Li, M. Li, and F. Tang. A distance-based directional broadcast protocol for urban vehicular ad hoc network. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom. International Conference on*, pages 1520 –1523, 21-25 2007. 33

[206] D. Li, D. Kim, Q. Zhu, L. Liu, and W. Wu. Minimum total communication power connected dominating set in wireless networks. In Xinbing Wang, Rong Zheng, Tao Jing, and Kai Xing, editors, *Wireless Algorithms, Systems, and Applications*, volume 7405 of *Lecture Notes in Computer Science*, pages 132–141. Springer Berlin Heidelberg, 2012. 48, 49

[207] F. Li and I. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *26th Annual IEEE Conference on Local Computer Networks*, pages 193–202, 2001. 47, 49

[208] F. Li, S. Z., X. Wang, X. Xue, and H. Shen. Vote-based clustering algorithm in mobile ad-hoc networks. In *International Conference on Networking Technologies for Broadband and Mobile Networks*, Lecture Note in Computer Science, pages 13–23, 2004. 43, 44

[209] J. Li, K. Li, and W. Zhu. Improving sensing coverage of wireless sensor networks by employing mobile robots. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 899 –903, 2007. doi: 10.1109/ROBIO.2007.4522282. 56

[210] N. Li and J.C. Hou. Blmst: a scalable, power-efficient broadcast algorithm for wireless networks. In *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, QSHINE*, pages 44–51, 2004. 48, 49

[211] N. Li, J.C. Hou, and S. Lui. Design and analysis of an mst-based topology control algorithm. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, volume 3, pages 1702–1712, 2003. 48, 49

[212] N. Li, J.C. Hou, and S. Lui. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005. 42, 48, 70

[213] X. Li, T. D. Nguyen, and R. P. Martin. Using adaptive range control to optimize 1-hop broadcast coverage in dense wireless networks. In *SenSys*, pages 314–315, 2003. 46

[214] O. Liang. *Multipoint Relay and Connected Dominating Set Based Broadcast Algorithms for Wireless Ad Hoc Networks*. PhD thesis, Melbourne, Australia, 2007. 36

[215] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 112–122, 2002. 47, 49

[216] W.-H. Liao, Y. Kao, and Y.-S. Li. A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. *Expert Systems with Applications*, 38(10):12180 – 12188, 2011. ISSN 0957-4174. 56

[217] D. Liarokapis and A. Shahrabi. A probability-based adaptive scheme for broadcasting in manets. In *Proceedings of the 6th International Conference on Mobile Technology, Application &Systems*, Mobility '09, pages 46:1–46:8, 2009. 31, 34

[218] D. Liarokapis and A. Shahrabi. Fuzzy-based probabilistic broadcasting in mobile ad hoc networks. In *Wireless Days (WD), 2011 IFIP*, pages 1–6, 2011. 31, 34

[219] D. Liarokapis, A. Shahrabi, and A. Komninos. Diba: An adaptive broadcasting scheme in mobile ad hoc networks. In *Communication Networks and Services Research*, pages 224–231. IEEE Computer Society, 2009. ISBN 978-0-7695-3649-1. 34

[220] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, MSWIM '00, pages 61–68, New York, NY, USA, 2000. ACM. 35, 36, 39, 98

[221] S. Lim, C. Yu, and C.R. Das. Randomcast: An energy-efficient communication scheme for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 8(8):1039–1051, 2009. 46, 47

[222] J. Lipman, P. Boustead, and J. Judge. Utility-based multipoint relay flooding in heterogeneous mobile ad hoc networks. In *Proceedings of the Workshop on the Internet, Telecommunications and Signal Processing*, pages 27–33, 2002. 38, 39

[223] J. Lipman, P. Boustead, J. Chicharo, and J. Judge. Resource aware information dissemination in ad hoc networks. In *The 11th IEEE International Conference on Networks*, pages 591–596, 2003. 38, 39

[224] J. Lipman, H. Liu, and I. Stojmenović. *Guide to Wireless Ad Hoc Networks*, chapter Broadcast in Ad hoc Networks. Chapter 6, pages 121–150. Springer, 2009. 29

[225] L. Liu and G. Feng. Swarm intelligence based node-disjoint multi-path routing protocol for mobile ad hoc networks. In *Fifth International Conference on Information, Communications and Signal Processing*, pages 598 –602, 2005. 60

[226] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A swarm intelligence routing algorithm for manets. In *Communications, Internet, and Information Technology*, pages 484–489, 2004. 58

[227] Z. Liu, Z. Liu, and L. Wen. A modified leach protocol for wireless sensor networks. In *Advanced Computational Intelligence (IWACI), 2011 Fourth International Workshop on*, pages 766 –769, oct. 2011. 61

[228] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2):111–122, 2002. 36, 37, 39

[229] W. Lou and J. Wu. Double-covered broadcast (dcb): a simple reliable broadcast algorithm in manets. In *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 2084–2095, 2004. 38, 39

[230] H. Lundien, J. R.; Nikodem. A mathematical model for predicting microseismic signals in terrain materials. Technical Report M-73-4, U.S. Army Engineer Waterways Experiment Station, 1973. 17

[231] R. Stuart Mackay. Radio telemetering from within the human body. *Medical Electronics, IRE Transactions on*, ME-6(2):100–105, 1959. 17

[232] R. Stuart Mackay and Bertil Jacobson. Endoradiosonde. *Nature*, 179:1239–1940, 1957. 17

[233] B. Mans and N. Shrestha. Performance evaluation of approximation algorithms for multipoint relay selection. In *Med-Hoc-Net 2004, The Third An- nual Mediterranean Ad Hoc Networking Workshop*, pages 480–491, Bodrum, Turkey, 2004. 37, 39

[234] F. Marcelloni and M. Vecchio. Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization. *Information Sciences*, 180(10):1924 – 1941, 2010. ¡ce:title¿Special Issue on Intelligent Distributed Information Systems¡/ce:title¿. 64

[235] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. ACM/IEEE 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 255–265, 2000. 63

[236] S. Marwaha, C. K. Tham, and D. Srinivasan. Mobile agents based routing protocol for mobile ad hoc networks. In *IEEE Global Telecommunications Conference GLOBECOM*, volume 1, pages 163–167, 2002. 58

[237] J. M. McQuillan, I. Richer, and E. C. Rosen. The new routing algorithm for the arpanet. *IEEE Transactions on Communications*, COM-28(5):711–719, May 1980. 81

[238] S. Misra, S. K. Dhurandher, M. S. Obaidat, P. Gupta, K. Verma, and P. Narula. An ant swarm-inspired energy-aware routing protocol for wireless ad-hoc networks. *Journal of Systems and Software*, 83(11):2188–2199, 2010. 60

[239] K. Miyao, H. Nakayama, N. Ansari, and N. Kato. Ltrt: An efficient and reliable topology control algorithm for ad-hoc networks. *IEEE Transactions on Wireless Communications*, 8(12):6050–6058, 2009. 48, 49

[240] I.-H. M. Mkwawa and D.D. Kouvatsos. Broadcasting methods in manets: An overview. In DemetresD. Kouvatsos, editor, *Network Performance Engineering*, volume 5233 of *Lecture Notes in Computer Science*, pages 767–783. Springer Berlin Heidelberg, 2011. 29

[241] S.A. Mohammad, A. Rasheed, and A. Qayyum. Vanet architectures and protocol stacks: a survey. In *Proceedings of the Third international conference on Communication technologies for vehicles*, Nets4Cars/Nets4Trains'11, pages 95–105. Springer-Verlag, 2011. 16

[242] J.M. Mounce. Human factors requirements for real-time motorist information displays. Technical Report DOT-11-8505, Texas Transportation Institute. Texas A&M University., 1978. 13

[243] S. Mueller, R. Tsang, and D. Ghosal. Multipath routing in mobile ad hoc networks: issues and challenges. In *Performance Tools and Applications to Networked Systems*, volume 2965 of *Lecture Notes in Computer Science*, pages 209–234. Springer-Verlag, 2004. 59

[244] J. Muszynski, S. Varrette, P. Bouvry, F. Seredynski, and S.U. Khan. Convergence analysis of evolutionary algorithms in the presence of crash-faults and cheaters. *Computers & Mathematics with Applications*, 64(12):3805–3819, 2012. 53

[245] Y. Mylonas, M. Lestas, and A. Pitsillides. Speed adaptive probabilistic flooding in cooperative emergency warning. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, pages 1–7, 2008. 30, 34, 98, 99

[246] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. MOCell: A cellular genetic algorithm for multiobjective optimization. *Int. Journal of Intelligent Systems, Special Issue on Nature Inspired Cooperative Strategies for Optimization*, 24(7):726–746, 2009. 147

[247] Visual networking index. Cisco forecast. Available at http://www.cisco.com/en/us/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html, Accessed July 2012. 11

[248] M. Ni, Z. Zhong, K. Wu, and D. Zhao. An energy efficient clustering scheme for mobile ad hoc networks. In *IEEE 71st Vehicular Technology Conference (VTC 2010-Spring)*, pages 1–5, 2010. 49

[249] S-Y Ni, Y-C Tseng, Y-S Chen, and J-P Sheu. The broadcast storm problem in a mobile ad hoc network. In *5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, 1999. ISBN 1-58113-142-9. 3, 26, 29, 31, 34, 35, 99, 106, 107

[250] R. Oda, T. Ohta, and Y. Kakuda. A transmission power control adaptive to variation of node density in mobile ad hoc network clustering. In *Second International Conference on Communication Software and Networks*, pages 141–146, 2010. 49

[251] International Society of Automation. Isa100 wireless standard. Available at http://www.isa100wci.org/, Last accessed September 2012. 22

[252] National Institute of Standards and Technology (NIST). Ieee 1451 family of standards. Available at http://www.nist.gov/el/isd/ieee/ieee1451.cfm, Last accessed September 2012. 19

[253] F.J. Ovalle-Martínez, A. Nayak, I. Stojmenović, J. Carle, and D. Simplot-Ryl. Area based beaconless reliable broadcasting in sensor networks. In *Proceedings of the Second international conference on Algorithmic Aspects of Wireless Sensor Networks*, pages 140–151. Springer-Verlag, 2006. 32, 34

[254] S. Park and S-M Yoo. An efficient reliable one-hop broadcast in mobile ad hoc networks. *Ad Hoc Networks*, 11(1):19 – 28, 2013. 46, 47

[255] V. Paruchuri, A. Durresi, and R. Jain. Optimized flooding protocol for ad hoc networks. *CoRR*, cs.NI/0311013, 2003. 32, 34

[256] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pages 129–130, 2000. 35, 39, 98

[257] W. Peng and X. Lu. *Efficient Broadcast in Mobile ad hoc Networks Using Connected Dominating sets*, 12(4):1265–1275, 2001. 39

[258] W. Peng and X-C LU. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Computer Science and Technology*, 16:114–125, 2001. ISSN 1000-9000. 37, 39, 98

[259] S. Petersen and S. Carlsen. Wirelesshart versus isa100.11a: The format war hits the factory floor. *IEEE Industrial Electronics Magazine*, pages 23–34, 2011. 22

[260] Y. Pigné. *Modélisation et Traitement Décentralisé des Graphes Dynamiques - Application Aux Réseaux Mobiles Ad Hoc*. PhD thesis, L'Université du Harve, December 2008. 86, 94

[261] A. Piwonska and F. Seredynski. A genetic algorithm with a penalty function in the selective travelling salesman problem on a road network. In *IPDPS Workshops*, pages 381–387, 2011. 53

[262] A. Piyatumrong. *Efficient and Robust Spanning Forest for Topology Management in DT-MANETs: A Tree-Based Decentralized and Multi-Objective Approach*. PhD thesis, University of Luxembourg, Luxembourg, 2010. 42

[263] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda. Trusted spanning tree for delay tolerant manets. In *EUC (2)*, pages 293–299, 2008. 42

[264] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda. Trusted spanning trees for delay tolerant mobile ad hoc networks. In *IEEE Conference on Soft Computing in Industrial Applications*, pages 131–136, 2008. 81, 82

[265] A. Piyatumrong, P. Ruiz, P. Bouvry, F. Guinand, and K. Lavangnananda. Traversal strategies of a distributed spanning forest algorithm in mobile ad hoc - delay tolerant networks. In *In Proceedings of the 3rd International Conference on Advances in Information Technology*, pages 96–109, Bangkok, Thailand, 2009. 89

[266] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda. A multi-objective approach for high quality tree-based backbones in mobile ad hoc networks. *International Journal od Space-based and Situated Computing*, 2(2):83–103, 2012. 57

[267] P. M. Pradhan and G. Panda. Connectivity constrained wireless sensor deployment using multiobjective evolutionary algorithms and fuzzy decision making. *Ad Hoc Netw.*, 10(6):1134–1145, August 2012. ISSN 1570-8705. 56

[268] K.V. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution – A Practical Approach to Global Optimization*. NCS. Springer, 2005. 147

[269] Geonet Project. D 1.2 final geonet architecture design, 2010. 16

[270] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA, 2000. 36, 39, 98

[271] M. Qin and R. Zimmermann. Studying upper bounds on sensor network lifetime by genetic clustering. In ViktorK. Prasanna, SitharamaS. Iyengar, PaulG. Spirakis, and Matt Welsh, editors, *Distributed Computing in Sensor Systems*, volume 3560 of *Lecture Notes in Computer Science*, pages 408–408. Springer Berlin Heidelberg, 2005. 61

[272] S. Radhakrishnan, G. Racherla, C.N. Sekharan, N.S.V. Rao, and S.G. Batsell. Dst-a routing protocol for ad hoc networks using distributed spanning trees. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1543–1547 vol.3, 1999. 42, 44

[273] P. Ranganathan, A. Ranganathan, K. Berman, and A. Minai. Discovering adaptive heuristics for ad-hoc sensor networks by mining evolved optimal configurations,Ó evolutionary computation, 2006. cec 2006. In *A Self-Organizing Heuristic for Building Optimal Heterogeneous Ad-Hoc Sensor Networks,Ó Networking, Sensing and Control, 2006. ICNSC Õ06. Proceedings of the 2006 IEEE International Conference on*, pages 16–21, 2006. 54

[274] F. De Rango and A. Socievole. *Mobile ad hoc networks: Applications*, chapter Metaheuristics techniques and swarm intelligence in mobile ad hoc networks, pages 245–264. InTech, 2011. 58

[275] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001. ISBN 0130422320. 114, 136

[276] Research and Innovative Technology Administration (Rita). Intelligent Transportation Systems. Ieee 1609 - family of standards for wireless access in vehicular environments (wave) (sept. 2009). Available at http://www.standards.its.dot.gov/fact_sheet.asp?f=80, Accessed July 2012. 14

[277] H. Reumerman and M. Runi. Distributed power control for reliable broadcast in inter-vehicle communication systems. In *2nd International Workshop on Intelligent transportation*, 2005. 46, 47

[278] N. Ristanovic, J-Y Le Boudec, A. Chaintreau, and V. Erramilli. Energy efficient offloading of 3g networks. In *Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems. MASS.*, pages 202–211, 2011. 12

[279] L. Rosati, M. Berioli, and G. Reali. On ant routing algorithms in ad hoc networks with critical connectivity. *Ad Hoc Networks*, 6(6):827–859, August 2008. 58

[280] M. Roth and S. Wicker. Termite: ad-hoc networking with stigmergy. In *IEEE Global Telecommunications Conference GLOBECOM*, volume 5, pages 2937–2941, 2003. 60

[281] P. Ruiz and P. Bouvry. Enhanced distance based broadcasting protocol with reduced energy consumption. In *Workshop on Optimization Issues in Energy Efficient Distributed Systems (OPTIM), part of the 2010 International Conference on High Performance Computing and Simulation (HPCS)*, pages 249–258, 2010. 117

[282] P. Ruiz, B. Dorronsoro, D. Khadraoui, P. Bouvry, and L. Tardón. BODYF–A Parameterless Broadcasting Protocol Over Dynamic Forest. In *Proceedings of the High Performance Computing & Simulation Conference (HPCS 08)*, pages 297–303, Nicosia, Cyprus, 2008. 81, 83, 86, 96, 101

[283] P. Ruiz, B. Dorronsoro, and P. Bouvry. Optimization and performance analysis of the AEDB broadcasting algorithm. In *International Workshop on Wireless Mesh and Ad Hoc Networks, in conjunction with International Conference on Computer Communication Networks (ICCCN)*, pages 1–6, Maui, Hawaii, 2011. 133, 162

[284] P. Ruiz, B. Dorronsoro, G. Valentini, F. Pinel, and P.l Bouvry. Optimisation of the enhanced distance based broadcasting protocol for manets. *The Journal of Supercomputing*, 62(3):1213–1240, 2012. 132

[285] P.M. Ruiz and A.F. Gomez-Skarmeta. Using genetic algorithms to optimize the behaviour of adaptive multimedia applications in wireless and mobile scenarios. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 2064 –2068, march 2003. 64

[286] DSRC Committee. SAE Draft Std. J2945.1 Revision 2.2, SAE International. Draft dsrc minimum performance requirements–basic safety messages for vehicle safety applications, April 2011. 15

[287] M. Saleem, G. A. Di Caro, and M. Farooq. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Information Sciences*, 181 (20):4597 – 4624, 2011. 58

[288] M. Saleem, I. Ullah, and M. Farooq. Beesensor: An energy-efficient and scalable routing protocol for wireless sensor networks. *Information Sciences*, 200(0):38 – 56, 2012. 59

[289] A. Saltelli, S. Tarantola, and S. Chan. A quantitative, model independent method for global sensitivity analysis of model output. *Technometrics*, 41(39-56), 1999. 72

[290] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, 2004. 72

[291] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, 2005. 53

[292] P. Sateesh Kumar and S. Ramachandram. The performance evaluation of genetic zone routing protocol for manets. In *2008 IEEE Region 10 Conference*, pages 1 –6, nov. 2008. 60

[293] J. Schleich. *Robust Dominating Set based Virtual Backbones for Wireless Ad hoc Networks*. PhD thesis, University of Luxembourg, Luxembourg, 2010. 40

[294] J. Schleich, G. Danoy, P. Bouvry, and A. Le Thi Hoai. Blackbone2, an efficient deterministic algorithm for creating 2-connected m-dominating set-based backbones in ad hoc networks. In *Proceedings of the Seventh ACM International Workshop on Mobility Management & Wireless Access*, pages 91–98, Tenerife, 2009. 81

[295] J. Schleich, A. Le Thi Hoai, and P. Bouvry. Solving the minimum m-dominating set problem by a continuous optimization approach based on dc programming and dca. *Journal Combinatorial Optimization*, 24(4):397–412, 2012. 40

[296] M. Schwartz and N. Abramson. The alohanet - surfing for wireless data [history of communications]. *Communications Magazine, IEEE*, 47(12):21–25, December 2009. 7

[297] S. Sen and J.A. Clark. Evolutionary computation techniques for intrusion detection in mobile ad hoc networks. *Comput. Netw.*, 55(15):3441–3457, October 2011. ISSN 1389-1286. 63

[298] S. Sengupta, S. Das, M.D. Nasir, and B.K. Panigrahi. Multi-objective node deployment in wsns: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. *Engineering Applications of Artificial Intelligence*, 26(1):405 – 416, 2013. 56

[299] M. Seredynski and P. Bouvry. Direct reciprocity-based cooperation in mobile ad hoc networks. In *International Journal of Foundations of Computer Science*, volume 23, pages 501–523. Word Scientific, 2012. 63

[300] M. Seredynski, P. Bouvry, and M. A. Klopotek. Evolution of cooperation in ad hoc networks under game theoretic model. In *Proc. The 4-th ACM International Workshop on Mobility Management and Wireless Access (MobiWac 2006)*, pages 126–130. ACM press, 2006. 63

[301] M. Seredynski, P. Bouvry, and M.A. Klopotek. Evolution of strategy driven behavior in ad hoc networks using a genetic algorithm. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1 –8, march 2007. 63

[302] M. Seredynski, G. Danoy, M. Tabatabaei, P. Bouvry, and Y. Pigné. Generation of realistic mobility for vanets using genetic algorithms. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1 –8, june 2012. 62

[303] Zach Shelby and Carsten Bormann. *6LoWPAN: THE WIRELESS EMBEDDED INTERNET*. Wiley, November 2009. 21

[304] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC, 2003. 149

[305] H. Shokrani and S. Jabbehdari. A survey of ant-based routing algorithms for mobile ad-hoc networks. In *2009 International Conference on Signal Processing Systems*, pages 323 –329, may 2009. 58

[306] A. Singh and W. N. Bhukya. A hybrid genetic algorithm for the minimum energy broadcast problem in wireless ad hoc networks. *Applied Soft Computing*, 11(1):667–674, 2011. 57

[307] G. Singh, N. Kumar, and A. Kumar Verma. Ant colony algorithms in manets: A review. *Journal of Network and Computer Applications*, 35(6):1964 – 1972, 2012. 58

[308] M. Slavik and I. Mahgoub. Stochastic broadcast for vanet. In *Consumer Communications and Networking Conference (CCNC), 7th IEEE*, pages 1 –5, Las Vegas, february 2010. 30, 34

[309] M. Slavik, I. Mahgoub, and F.N. Sibai. The distance-to-mean broadcast method for vehicular wireless communication systems. In *International Conference on Innovations in Information Technology (IIT)*, pages 371–374, 2011. 33, 34

[310] E.Y. Song and Kang Lee. Understanding ieee 1451-networked smart transducer interface standard - what is a smart transducer? *Instrumentation Measurement Magazine, IEEE*, 11(2):11–17, 2008. 20

[311] E. Sopena and Y. Metivier. Graph relabeling systems: a general overview. *Computers and Artificial Intelligence*, 16(2):167–185, 1997. 82

[312] E.M. Sözer, M. Stojanovic, and J. G. Proakis. Initialization and routing optimization for ad-hoc underwater acoustic networks. In *In Proceedings of Opnetwork*, 2000. 59

[313] V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine*, 43(12):112–119, 2005. 106

[314] ITS Standars. Calm standards for its (nov. 2009). Available at http://its-standards.info/Feeds/21217Intro.html, Accessed July 2012. 15

[315] ITS Standars. Communications in cooperative intelligent transport systems - calm for c-its. Available at http://calm.its-standards.info/, Accessed July 2012. 16

[316] Twitter statistics. Available at http://blog.twitter.com/2012/03/twitter-turns-six.html, Accessed July 2012. 11

[317] I. Stojmenović. A general framework for broadcasting in static to highly mobile wireless ad hoc, sensor, robot and vehicular networks. In *IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 8–13, 2012. 40

[318] I. Stojmenović and J. Wu. *Broadcasting and activity scheduling in ad hoc networks.* Eds. S. Basagni and M. Conti and S. Giordano and I. Stojmenović, 2004. 27, 29, 40, 48, 49

[319] I. Stojmenovic, M. Russell, and B. Vukojevic. Depth first search and location based localized routing and qos routing in wireless networks. In *Intl. Conf. on Parallel Processing (ICPP'00)*, 2000. 86

[320] I. Stojmenović, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002. 40

[321] I. Stojmenović, A. Khan, and N. Zaguia. Broadcasting with seamless transition from static to highly mobile wireless ad hoc, sensor and vehicular networks. *International Journal of Parallel Emergent and Distributed Systems*, 27(3):225–234, June 2012. ISSN 1744-5760. 40

[322] J. Su, A. Goel, and E. de Lara. An empirical evaluation of the student-net delay tolerant network. *Mobile and Ubiquitous Systems, Annual International Conference on*, pages 1–10, 2006. 81

[323] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: a case study in routing in dynamic networks. In *Proceedings of the Fifteenth international joint conference on Artificial intelligence (IJCAI) - Volume 2*, pages 832–838, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-555860-480-4. 58

[324] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. Technical Report 248, Rutgers University, CAIP, 2000. 36, 39

[325] M.-T. Sun and T-H Lai. Location aided broadcast in wireless ad hoc network systems. In *IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 597–602, 2002. 32, 34

[326] P. Switalski and F. Seredynski. Solving multiprocessor scheduling problem with geo metaheuristic. In *IPDPS*, pages 1–8, 2009. 53

[327] M. Takahashi and H. Kita. A crossover operator using independent component analysis for real-coded genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 643 –649, 2001. 139

[328] E.G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009. ISBN 9780470496909. URL `http://books.google.lu/books?id=SIsa6zi5XV8C`. 50

[329] F.A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *Communications, IEEE Transactions on*, 23(12):1417–1433, 1975. 8

[330] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar. Broadcasting in vanet. In *Mobile Networking for Vehicular Environments*, pages 7 –12, 2007. 35

[331] O.K. Tonguz, N. Wisitpongphan, and Fan B. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. *Wireless Communications, IEEE*, 17(2): 47 –57, april 2010. 35, 39

[332] J. Toutouh, J. Garcia-Nieto, and E. Alba. Intelligent OLSR routing protocol optimization for VANETs. *IEEE Transactions on Vehicular Technology*, 61(4):1884 –1894, 2012. ISSN 0018-9545. 62

[333] J. Toutouh, S. Nesmachnow, and E. Alba. Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm. *Cluster Computing*, pages 1–16, 2012. ISSN 1386-7857. 62

[334] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, 52:545–557, 2002. 31, 33, 34, 35, 39

[335] D. Turgut, S.K. Das, R. Elmasri, and B. Turgut. Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 62–66, nov. 2002. 61

[336] E. Urrea, C.S. Şahin, I. Hökelek, M.I. Uyar, M. Conner, G. Bertoli, and C. Pizzo. Bio-inspired topology control for knowledge sharing mobile agents. *Ad Hoc Networks*, 7(4): 677–689, June 2009. ISSN 1570-8705. 56

[337] Bluetooth v4.0. Core specification. Available at http://developer.bluetooth.org/KnowledgeCenter/TechnologyOverview/Pages/v4.aspx, Last accessed September 2012. 21

[338] D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, 1999. 153

[339] M. Čagalj, J-P Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 172–182. ACM, 2002. 44, 45, 47, 49

[340] A. C. K. Vendramin, A. Munaretto, M. R. Delgado, and A. C. Viana. Grant: Inferring best forwarders from complex networksÕ dynamics through a greedy ant colony optimization. *Computer Networks*, 56(3):997 – 1015, 2012. 57

[341] K. Viswanath and K. Obraczka. An adaptive approach to group communications in multi hop ad hoc networks. In *Seventh International Symposium on Computers and Communications*, pages 559–566, 2002. 30, 34

[342] E. Vollset and P. Ezhilchelvan. A survey of reliable broadcast protocols for mobile ad-hoc networks. Technical Report CS-TR-792, University of Newcastle, 2003. URL `http://books.google.lu/books?id=yprkMgEACAAJ`. 29

[343] P-J Wan, G. Calinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1162–1171, 2001. 47, 49

[344] H-C Wang, J-B Lin, F-C Kuo, and K-C Ting. Combination of relative neighborhood graph and forbidden set in the design of distributed broadcast algorithms for wireless ad hoc networks. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 94–100, 2010. 45, 47

[345] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram. Hopnet: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*, 7 (4):690–705, 2009. 58

[346] X. Wang, S. Wang, and J-J Ma. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. *Sensors*, 7(3):354–370, 2007. ISSN 1424-8220. 56, 57

[347] C. Wewetzer and A. Caliskan, M.and Luebke. The feasibility of a search engine for metropolitan vehicular ad-hoc networks. In *IEEE Globecom Workshops*, pages 1–8, 2007. 10

[348] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *The nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 585–594, 2000. 47, 49, 70

[349] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. The energy efficiency of distributed algorithms for broadcasting in ad hoc networks. In *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, volume 2, pages 499–503, 2002. 48, 49

[350] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002. 26, 98

[351] N. Wisitpongphan, O.K. Tonguz, J.S. Parikh, P. Mudalige, F. Bai, and V. Sadekar. Broadcast storm mitigation techniques in vehicular ad hoc networks. *Wireless Communications, IEEE*, 14(6):84 –94, December 2007. 30, 34, 98, 100

[352] S. Wolf and P. Merz. Iterated local search for minimum power symmetric connectivity in wireless networks. In Carlos Cotta and Peter Cowling, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 5482 of *Lecture Notes in Computer Science*, pages 192–203. Springer Berlin-Heidelberg, 2009. 57

[353] St. Wolf and P. Merz. Evolutionary local search for the minimum energy broadcast problem. In Jano van Hemert and Carlos Cotta, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 4972 of *Lecture Notes in Computer Science*, pages 61–72. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-78603-0. 57

[354] W. Woon and K.L. Yeung. Variable power broadcasting based on local information for source-dependent broadcasting protocols. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 796–801, 2011. 46

[355] IEEE 802.11 working group. Ieee 802.11 timelines (1991). Available at http://www.ieee802.org/11/Reports/802.11_Timelines.htm, Accessed July 2012. 11, 12

[356] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):86–881, 2002. 39

[357] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, pages 7–14, 1999. ISBN 1-58113-174-7. 39

[358] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 3(2):155–173, 2003. 43, 44

[359] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc. *Wireless Communications and Mobile Computing*, 2(3):155–173, 2003. 27

[360] J. Wu and B. Wu. A transmission range reduction scheme for power-aware broadcasting in ad hoc networks using connected dominating sets. In *IEEE Vehicular Technology Conference (VCT2003*, 2003. 48, 49

[361] J. Wu, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 4:59–70, 2000. 39

[362] J. Wu, W. Lou, and F. Dai. Extended multipoint relays to determine connected dominating sets in manets. *IEEE Trans. Comput.*, 55(3):334–347, March 2006. ISSN 0018-9340. 40, 44

[363] X. Wu, X. Wang, and R. Liu. Solving minimum power broadcast problem in wireless ad-hoc networks using genetic algorithm. In *Communication Networks and Services Research Conference CNSR*, pages 203–207, 2008. 57

[364] H. Xu and J.J. Garcia-Luna-Aceves. Efficient broadcast for wireless ad hoc networks with a realistic physical layer. *Ad hoc Networks*, 8:165–180, 2010. 48, 49

[365] K. K. Yadu, A. Tiwari, and O. G. Kakde. Optimization based topology control for wireless ad hoc networks to meet QoS requirements. In *IEEE Symposium on Reliable Distributed Systems*, pages 30–36, 2010. 54

[366] S. Yang, H. Cheng, and F. Wang. Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(1):52 –63, jan. 2010. 59

[367] Y-S. Yen, Y-K. Chan, H-C. Chao, and J.H. Park. A genetic algorithm for energy-efficient based multicast routing on manets. *Comput. Commun.*, 31(4):858–869, March 2008. 60

[368] Y-S. Yen, H-C. Chao, R-S. Chang, and A. Vasilakos. Flooding-limited and multi-constrained qos multicast routing based on the genetic algorithm for manets. *Mathematical and Computer Modelling.*, 53(11-12):2238–2250, June 2011. 60

[369] H. Yetgin, K.T.K. Cheung, and L. Hanzo. Multi-objective routing optimization using evolutionary algorithms. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3030 –3034, april 2012. 59

[370] Y. Yi, M. Gerla, and T-J Kwon. Efficient flooding in ad hoc networks: a comparative performance study. In *IEEE International Conference on Communications (ICC'03)*, pages 1059–1063, 2003. 27

[371] C. Yibo, K-M Hou, H. Zhou, H-L Shi, X. Liu, X. Diao, H. Ding, J-J Li, and C. de Vaulx. 6lowpan stacks: A survey. In *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1–4, 2011. 21

[372] J. Yu, N. Wang, G. Wang, and D. Yu. Connected dominating sets in wireless ad hoc and sensor networks – a comprehensive survey. *Computer Communications*, 36(2):121–134, 2013. 39

[373] J.Y. Yu and P. H J Chong. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys Tutorials*, 7(1):32–48, 2005. 43

[374] P. Zanchetta, M. Sumner, F. Cupertino, M. Marinelli, and E. Mininno. On-line and off-line control design in power electronics and drives using genetic algorithms. In *Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE*, volume 2, pages 864 – 871 vol.2, oct. 2004. 25

[375] J. H. Zar. *Biostatistical Analysis*. Prentice Hall, 1999. 149

[376] H. Zeng, M. Li, H. Liu, and X. Jia. Efficient flooding in mobile ad hoc networks. In J.N. Turner and C.S. Boyer, editors, *Ad Hoc Networks: New Research*. Nova Science Publishers, 2009. 29

[377] W. Zhao, M. Ammar, and E. Zegura. Multicasting in delay tolerant networks: semantic models and routing algorithms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 268–275, New York, NY, USA, 2005. ACM. 81, 86

[378] J. Zhong and J. Zhang. Energy-efficient local wake-up scheduling in wireless sensor networks. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2280 –2284, june 2011. 54

[379] L. Zhou, G. Cui, H. Liu, Z.Wu, and D. Luo. Nppb: A broadcast scheme in dense vanets. *Information Technology Journal*, 9(2):247 –256, 2010. 30, 34

[380] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer. Parameter study for differential evolution using a power allocation problem including interference cancellation. In *IEEE Congress on Evolutionary Computation CEC*, pages 1857 –1864, 2006. 55

[381] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer. Optimization of power allocation for interference cancellation with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):128 –150, 2009. ISSN 1089-778X. 55

[382] H. Zimmermann. Osi reference model. *IEEE Transactions on Communications*, 28(4): 425, 1980. 16

[383] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. 153

Publications

[JSC12] Patricia Ruiz, Bernabé Dorronsoro, Giorgio Valentini, Frederic Pinel and Pascal Bouvry. *Optimisation of the enhanced distance based broadcasting protocol for MANETs.* Journal of Supercomputing. Special Issue on Green networks. Volume 62, Issue 3. Pages: 1213-1240. 2011 Springer

[IJCDS12] Patricia Ruiz and Pascal Bouvry. *On the Improvement of the Enhanced Distance Based Broadcasting Algorithm.* Int. J. of Communication Networks and Distributed Systems. InderScience. Vol. 9, Nos 3/4, Pages 225-246. 2011

[AdhocNet12] Patricia Ruiz, Bernabé Dorronsoro, Pascal Bouvry and Lorenzo J. Tardón. *Information Dissemination in VANETs based upon a Tree Topology.* Journal of Ad hoc Networks. Vol 10 issue 1. Pages 111-127. 2012. Elsevier.

[ClusCom13] Patricia Ruiz, Bernabé Dorronsoro and Pascal Bouvry. *Finding Scalable Configurations for AEDB Broadcasting Protocol using Multi-objective Evolutionary Algorithms.* Cluster Computing. Volume 16, Issue 3. Page 527-544. 2013.

[NIDISC13] Santiago Iturriaga, Patricia Ruiz, Sergio Nesmachnow, Bernabé Dorronsoro and Pascal Bouvry. *A Parallel Multi-objective Local Search for AEDB Protocol Tuning.* NIDISC The 16th International Workshop on Nature Inspired Distributed Computing held in conjunction with the 27th IEEE/ACM International Parallel and Distributed Processing (IPDPS). Pages 415-424, Boston, 2013

[ICCCN11] Patricia Ruiz, Bernabé Dorronsoro and Pascal Bouvry. *Optimization and Performance Analysis of the AEDB Broadcasting Algorithm.* International Workshop on Wireless Mesh and Ad Hoc Networks, in conjunction with International Conference on Computer Communication Networks, IEEE ICCCN. Pages 1-6. Maui, Hawaii. August, 2011

[GLOBCOM10] Patricia Ruiz and Pascal Bouvry. *Distributed Energy Self-Adaptation in ad hoc Networks.* In proc. of IEEE Int. workshop on Management of Emerging Networks and Services (MENS), in conjunction with IEEE Globecom. Pages 524-528. Miami, USA, 2010

[HPCS10] Patricia Ruiz and Pascal Bouvry. *Enhanced Distance Based Broadcasting Protocol with Reduced Energy Consumption.* Workshop on Optimization Issues in Energy Efficient Distributed Systems (OPTIM), held in conduction with the International Conference on High Performance Computing and Simulation (HPCS). Pages 249-258. Caen, France, 2010

[IAIT09] A. Piyatumrong, P. Ruiz, P. Bouvry, F. Guinand and K. Lavangnananda. *Token Traversal Strategies of a Distributed Spanning Forest Algorithm in Mobile Ad hoc - Delay Tolerant Networks*. In Proceedings of the 3rd International Conference on Advances in Information Technology (IAIT'09). Springer CCIS series. Pages 96-109. Bangkok, Thailand, 2009

[HPCS08] P. Ruiz, B. Dorronsoro, D. Khadraoui, P. Bouvry, and L. Tardón. *BODYF–A Parameterless Broadcasting Protocol over Dynamic Forest*. In Proceedings of the High Performance Computing & Simulation Conference (HPCS). Pages 297–303, Nicosia, Cyprus, 2008

Index